



**This electronic thesis or dissertation has been  
downloaded from Explore Bristol Research,  
<http://research-information.bristol.ac.uk>**

*Author:*  
**Sokol, Kacper**

*Title:*  
**Towards Intelligible and Robust Surrogate Explainers  
*A Decision Tree Perspective***

**General rights**

Access to the thesis is subject to the Creative Commons Attribution - NonCommercial-No Derivatives 4.0 International Public License. A copy of this may be found at <https://creativecommons.org/licenses/by-nc-nd/4.0/legalcode>. This license sets out your rights and the restrictions that apply to your access to the thesis so it is important you read this before proceeding.

**Take down policy**

Some pages of this thesis may have been removed for copyright restrictions prior to having it been deposited in Explore Bristol Research. However, if you have discovered material within the thesis that you consider to be unlawful e.g. breaches of copyright (either yours or that of a third party) or any other law, including but not limited to those relating to patent, trademark, confidentiality, data protection, obscenity, defamation, libel, then please contact [collections-metadata@bristol.ac.uk](mailto:collections-metadata@bristol.ac.uk) and include the following information in your message:

- Your contact details
- Bibliographic details for the item, including a URL
- An outline nature of the complaint

Your claim will be investigated and, where appropriate, the item in question will be removed from public view as soon as possible.

---

---

# Towards Intelligible and Robust Surrogate Explainers

*A Decision Tree Perspective*

---

---

By

KACPER SOKOL



Department of Computer Science  
UNIVERSITY OF BRISTOL

A dissertation submitted to the University of Bristol in accordance with the requirements of the degree of DOCTOR OF PHILOSOPHY in the Faculty of Engineering.

MARCH, 2021



## ABSTRACT

Artificial intelligence explainability and machine learning interpretability are relatively young and fast-growing research fields that may seem chaotic and difficult to navigate at times. Despite these immense endeavours, a universally agreed terminology and evaluation criteria are still elusive, with many methods introduced to solve a commonly acknowledged yet undefined problem, and their success judged based on ad hoc measures. To address this challenge and lay foundation for our research, we formalise **explainability** (our preferred term) as a technology providing insights that lead to understanding, which both defines such techniques and fixes their evaluation criterion. While the premise is clear, understanding largely depends upon the explanation recipients, who come with a diverse range of background knowledge, mental models and expectations. Therefore, in addition to technical requirements, explainability tools should also embody various social traits as their output is predominantly aimed at humans. To tackle this duality and organise a comprehensive collection of relevant properties, we introduce a unified explainable artificial intelligence **taxonomy**, which is a principled framework for reasoning about explainers. While most of our contributions are strictly technical, this formalisation allows us to develop them with a human component in mind, which leads us to consider explainability as a social, bi-directional process based on contrastive statements. Stemming from this research direction is **Glass-Box** – a conversational explainer that empowers its users to customise and personalise explanations in a natural language dialogue.

With strong foundations, clear requirements and fixed goals, we set out to design an appropriate explainer of predictive black boxes. In particular, we examine post-hoc and model-agnostic methods given that they are universally applicable to a wide variety of preexisting models, thereby increasing their potential reach and impact. While such explainers are appealing, their design can be an inherent cause of low-fidelity explanations, which lack truthfulness with respect to the underlying black box. Furthermore, their flexibility means that technically they can be applied to any predictive model, however they may not necessarily be equally well suited to the intricacies of each and every one of them. To address these challenges, we propose **bLIMEy** – a meta-algorithm for building tailor-made explainers composed of interchangeable building blocks spanning three dimensions: data augmentation, interpretable representation composition and explanation generation. Our method is a generic framework for developing surrogate explainers, which fits an explainable model in a desired decision subspace of a black box to mimic, thus simplify, its behaviour. We then investigate bLIMEy design principles to uncover that certain combinations of the aforementioned components may yield subpar explanations, pointing towards the benefits of using decision trees as the surrogate explanation generation model, which poses further challenges. While in some cases decision trees may be considered transparent, e.g., shallow and narrow trees, we argue that this does not imply their explainability. To achieve the latter, we propose **CtreeX** – an algorithm for extracting contrastive prediction explanations from decision trees, which are the gold standard of artificial intelligence explainability due to

---

their succinctness and human appeal. Finally, we merge our two findings in an approach called **LIMETree**, which uses surrogate multi-output (regression) trees to explain several classes at the same time, thus capturing their inter-dependencies within high-fidelity contrastive explanations.

*What I cannot create, I do not understand.*

(Richard Feynman)



## AUTHOR'S DECLARATION

I declare that the work in this dissertation was carried out in accordance with the requirements of the University's Regulations and Code of Practice for Research Degree Programmes and that it has not been submitted for any other academic award. Except where indicated by specific reference in the text, the work is the candidate's own work. Work done in collaboration with, or with the assistance of, others, is indicated as such. Any views expressed in the dissertation are those of the author.

SIGNATURE: ..... DATE: .....



## TABLE OF CONTENTS

	Page
<b>List of Tables</b>	<b>xi</b>
<b>List of Algorithms</b>	<b>xiii</b>
<b>List of Figures</b>	<b>xv</b>
<b>Glossary</b>	<b>xxiii</b>
 <b>1 What, Why and How of Explainable AI</b>	 <b>1</b>
1.1 Black-Box Artificial Intelligence . . . . .	1
1.1.1 Trading off Transparency for Predictive Power . . . . .	4
1.1.2 Explaining the Machine Learning Process . . . . .	5
1.1.3 Transparency Is Not the Preferred Nomenclature . . . . .	8
1.2 Humans and Explanations: Two Sides of the Same Coin . . . . .	11
1.2.1 Explanation Audience . . . . .	12
1.2.2 Explanatory Process . . . . .	13
1.2.3 Contrastive Explanations . . . . .	14
1.3 Benefits of Explainability . . . . .	15
1.4 Explainability Research . . . . .	18
1.5 Designing Intelligible and Robust Explainers . . . . .	21
1.5.1 Research Motivation . . . . .	22
1.5.2 Research Aims and Objectives . . . . .	23
1.5.3 Research Contributions . . . . .	24
1.6 Outline of the Thesis . . . . .	27
 <b>2 Explainable AI Taxonomy</b>	 <b>33</b>
2.1 Organising Explainability . . . . .	33
2.2 Taxonomic Ranks: Dimensions of Explainability . . . . .	35
2.2.1 Functional Requirements . . . . .	37
2.2.2 Operational Requirements . . . . .	40
2.2.3 Usability Requirements . . . . .	44

## TABLE OF CONTENTS

---

2.2.4	Safety Requirements . . . . .	48
2.2.5	Validation Requirements . . . . .	50
2.3	Taxonomy Trade-offs . . . . .	52
2.4	Applying the XAI Taxonomy . . . . .	54
2.4.1	Target Audience . . . . .	54
2.4.2	Delivery Format and Medium . . . . .	55
2.4.3	Operationalisation . . . . .	56
2.4.4	Selecting Dimensions and Requirements . . . . .	57
2.5	Systematic Evaluation Approaches in AI . . . . .	58
2.6	In Search of the Explainer: Surrogates Desiderata . . . . .	59
<b>3</b>	<b>bLIMEy: Modular Surrogate Explainers</b>	<b>61</b>
3.1	The Family of Surrogate Explainers . . . . .	62
3.2	From LIME to bLIMEy: Beyond Linear Surrogates . . . . .	66
3.2.1	Benefits of Interpretable Representations . . . . .	66
3.2.2	LIME: A Surrogate Explainer of Black-box Predictions . . . . .	71
3.2.3	bLIMEy: A Meta-algorithm for Building Modular Surrogate Explainers . . . . .	75
3.3	bLIMEy Modules . . . . .	78
3.3.1	The Challenge of Building Surrogate Explainers . . . . .	78
3.3.2	Interpretable Data Representation . . . . .	80
3.3.3	Data Sampling . . . . .	90
3.3.4	Explanation Generation . . . . .	92
3.4	Tailor-made Surrogate Explainers . . . . .	94
3.4.1	Compatibility of bLIMEy Modules . . . . .	95
3.4.2	Tree-based Surrogates . . . . .	97
3.4.3	Evaluating Surrogate Explainers . . . . .	98
3.5	Perspectives on Surrogate Explainers . . . . .	100
3.6	If You Were to Choose One . . . . .	101
<b>4</b>	<b>CtreeX: Contrastive Explanations for Decision Trees</b>	<b>105</b>
4.1	Decision Trees: Transparent but Not Explainable . . . . .	105
4.2	Inherent Transparency of Decision Trees . . . . .	109
4.3	Tree-based Class-contrastive and Supportive Explanations . . . . .	113
4.3.1	Meta-feature Tree Representation . . . . .	114
4.3.2	Distance Metrics . . . . .	116
4.3.3	Explanation Generation . . . . .	118
4.4	Making Trees Explainable . . . . .	119
4.5	Tree Explainability in the Literature . . . . .	121
4.6	Explainable Tree-based Surrogates . . . . .	123

<b>5</b>	<b>LIMEtree: Tree-based Surrogates</b>	<b>125</b>
5.1	Surrogates for Humans . . . . .	126
5.2	Surrogate Explainers, Revisited . . . . .	129
5.2.1	Local Surrogates of Images . . . . .	129
5.2.2	LIME Trade-offs . . . . .	132
5.3	Surrogate Multi-output Regression Trees . . . . .	136
5.3.1	Advantages of Multi-output Regression Surrogates . . . . .	136
5.3.2	LIMEtree . . . . .	138
5.3.3	Improved Surrogate Fidelity . . . . .	140
5.4	Examples of LIMEtree Explanations . . . . .	142
5.5	Advantages of LIMEtree . . . . .	145
5.5.1	Personalised and Interactive Explainability . . . . .	146
5.5.2	Generalisability and Applicability . . . . .	149
5.6	Experimental Results . . . . .	153
5.6.1	Synthetic Validation . . . . .	153
5.6.2	User Study . . . . .	155
5.7	Interactive and Surrogate Explainability Research . . . . .	157
5.8	Forging a Human–Machine Link . . . . .	159
<b>6</b>	<b>Glass-Box: One Explanation Does Not Fit All</b>	<b>161</b>
6.1	Interactive Explainability . . . . .	162
6.2	Conversational Explanations . . . . .	166
6.2.1	Glass-Box Design . . . . .	167
6.2.2	Explanation Desiderata . . . . .	169
6.2.3	Glass-Box Properties . . . . .	173
6.2.4	Glass-Box Reception and Feedback . . . . .	175
6.3	Real-life Deployment . . . . .	176
6.3.1	Lessons Learnt . . . . .	176
6.3.2	Improving Glass-Box . . . . .	178
6.4	Interactive Explainability in the Literature . . . . .	181
6.4.1	Technical Points of View . . . . .	181
6.4.2	Interdisciplinary Perspective . . . . .	183
6.5	It Is All about the Explainee . . . . .	185
<b>7</b>	<b>Conclusions and Future Directions</b>	<b>187</b>
7.1	Explainability, What Is It Good For? . . . . .	188
7.2	Towards Human-like Explanations . . . . .	193
<b>A</b>	<b>Explainability Fact Sheet Examples</b>	<b>195</b>

## TABLE OF CONTENTS

---

A.1	LIME: Local Interpretable Model-agnostic Explanations . . . . .	196
A.2	CtreeX: Contrastive tree eXplainer . . . . .	206
A.3	LIMEtree: Tree-based Surrogate Explainer . . . . .	215
<b>B</b>	<b>FAT Forensics and Reproducibility</b>	<b>227</b>
B.1	Origin of FAT Forensics . . . . .	227
B.2	Toolbox Overview . . . . .	229
B.3	Related Software . . . . .	234
<b>C</b>	<b>Analysis of bLIMEy Components and Their Interactions</b>	<b>237</b>
C.1	Occlusion-based Interpretable Representations of Images . . . . .	238
C.2	Tree-based Interpretable Representations of Tabular Data . . . . .	241
C.3	Binary Interpretable Representations of Tabular Data . . . . .	244
	<b>Bibliography</b>	<b>253</b>
	<b>Index</b>	<b>277</b>

## LIST OF TABLES

TABLE	Page
2.1 Overview of the explainable artificial intelligence taxonomy. . . . .	36
4.1 Tree splits $\mathcal{S}$ (left), meta-feature representation $\mathcal{B}$ (middle) and pairwise leaf distance (right) for the decision tree given in Figure 4.6. The splits $s_i \in \mathcal{S}$ are annotated as T for True, F for False and U for Undecided (i.e., not used/applicable). The meta-feature space built upon these splits is determined by $\mathcal{B} = \{b_{11} : x_1 = \text{red}, b_{12} : x_1 = (\text{green} \vee \text{blue}), b_{21} : x_2 < 5, b_{22} : 5 \leq x_2 < 7, b_{23} : 7 \leq x_2\}$ . The right part of the table groups the leaves by their predicted class to highlight similarity of leaves with the same (diagonal quadrants) and opposite (off-diagonal) predictions. . . . .	115
4.2 Summary reference of a subset of the XAI taxonomy (cf. Chapter 2) applicable to CtreetX. . . . .	120
5.1 <i>Per-class</i> fidelity computed with the LIME loss (Equation 5.2) for different surrogate approaches (cf. Section 5.6.1). The results are based on explanations of 100 images for their top three predicted classes. (Lower is better.) . . . . .	154
5.2 Fidelity of the <i>top n</i> classes computed with the LIMETree loss (Equation 5.4) for different surrogate approaches (cf. Section 5.6.1). The results are based on explanations of 100 images for their top three predicted classes. When computing the LIMETree loss for one class, the factor of $\frac{1}{2}$ is removed. (Lower is better.) . . . . .	154
6.1 Summary of a subset of the XAI taxonomy (cf. Chapter 2) applicable to interactive explainers that support personalisation. (See Section 6.2.2 for a comprehensive discussion of these properties.) . . . . .	170
6.2 Summary of a subset of the XAI taxonomy (cf. Chapter 2) specifically applicable to Glass-Box. (See Section 6.2.3 for a comprehensive discussion of these properties.) . . . . .	173
7.1 Summary of the explainable artificial intelligence taxonomy introduced in Chapter 2. . . . .	192
B.1 Fairness, accountability and transparency functionality implemented in the latest release (version 0.1.0) of FAT Forensics. . . . .	232



## LIST OF ALGORITHMS

ALGORITHM	Page
3.1 bLIMEy meta-algorithm. Sampling (Step 2) and weighting (Step 5) are done in the interpretable domain $\mathcal{X}'$ . The order of Steps 5 and 6 – which are <i>optional</i> – can be reversed. . . . .	77
5.1 LIMETree. . . . .	140



## LIST OF FIGURES

FIGURE	Page
1.1 Fictitious depiction of an anecdotal trade-off between transparency and predictive power of AI systems. . . . .	4
1.2 An explanation of a model’s prediction of the <i>versicolor</i> class when varying the <i>petal length</i> feature value for the Iris data set [41]. ICE of a selected instance is plotted in red; grey represents ICEs of all the training data; and orange is the PD of the model achieved by averaging all the individual ICEs. . . . .	7
1.3 Depiction of “The Blind Men and the Elephant” parable. It symbolises that individual pieces of evidence may be contradictory and can often be insufficient to understand the bigger picture without first being aggregated and grounded in a shared context. . . . .	17
3.1 Example of an influence-based explanation of text with a <i>bag-of-words</i> interpretable representation. Panel (a) illustrates a sentence whose (positive) <i>sentiment</i> is being decided by a black-box model. The colouring of each word in Panel (a) conveys its influence on the prediction, with Panel (b) depicting the corresponding magnitudes. . . . .	68
3.2 Example of an influence-based explanation of image data with the interpretable representation built upon <i>segmentation</i> . Panel (a) illustrates an image that is being classified by a black-box model. The colouring of each super-pixel in Panel (a) conveys its influence on <i>Eskimo dog</i> prediction, with Panel (b) depicting the corresponding magnitudes. . . . .	69
3.3 Image occlusion strategy influences the resulting explanations (see Appendix C.1). The picture shown in Figure 3.2a is classified by a black box as <i>Eskimo dog</i> with 83% probability. Mean-colour occlusion of all the segments but one (a) results in 77% and black occlusion (b) in 9% probability of the same class, showing that the former approach cannot effectively remove information from this particular image. . . . .	70

- 3.4 Example of an influence-based explanation of tabular data with the interpretable representation built upon *discretisation* and *binarisation*. Panel (a) illustrates an instance (red  $\star$ ) that is being predicted by a black-box model. The dashed blue lines mark feature partitions; grey and green denote two predicted classes; and  $x^\star$  is the binary IR created for the  $\star$  data point. Panel (b) depicts the magnitude of the influence that  $x_1^\star : 75 \leq x_1$  and  $x_2^\star : 40 \leq x_2 < 80$  have on predicting the *grey* class for the  $\star$  instance (as well as any other data point located within the same hyper-rectangle). 72
- 3.5 Validating a surrogate explainer as a whole may be insufficient (a) given its diverse building blocks and their parameterisation. Instead, each individual component – data sampling, interpretable representation and explanation generation – should be evaluated on its own (b). . . . . 78
- 3.6 Example of interpretable representation transformation in both directions for image data. Panel (a) depicts steps required to represent a picture as a binary on/off vector, and Panel (b) illustrates this procedure in the opposite direction. Both transformations are *deterministic* given a fixed image segmentation and occlusion colour. . . . . 81
- 3.7 Discretisation is the main building block of interpretable representations of tabular data. It can either be learnt based on data features alone – Panel (a) – or additionally consider their black-box predictions (background shading) – Panel (b). . . . . 83
- 3.8 Interpretable representations based on decision trees result in purer hyper-rectangles (y-axis, lower is better) and fewer encodings (x-axis) when compared to equivalent quartile-based IRs, i.e., they are more flexible and expressive. The number of unique encodings used by quartile-based IRs is constant for a data set and is displayed in the legend (presented as the number of encodings used, out of the theoretical limit supported by the representation); whereas for tree-based IRs, it is equivalent to the number of leaves, which is recorded on the x-axis. See Figure C.2 and Appendix C.2 for more details. . . . . 84
- 3.9 Interpretable representations learnt for the two-dimensional two moons data set. A global (a&c) or a local (b&d) data sample is used in combination with a quartile (a&b) or a decision tree-based (c&d) discretisation. Local approaches (b&d) are better at capturing the intricate behaviour of the black-box decision boundary in the neighbourhood of the explained instance (black dot). Additionally, tree-based interpretable representations (c&d) require less partitions and are more faithful since they account for the black-box predictions. . . . . 85

3.10	Mean squared error (y-axis) calculated between the top prediction of an image (probability estimate) and predictions of the same class when progressively occluding a higher number of segments (x-axis) with a given colouring strategy. The panels show that the mean occlusion strategy is not as effective at hiding information from the black box as using a single colour for all of the super-pixels (regardless of the colour choice). Similarly, randomising the occlusion colour for each individual segment does not seem to have the detrimental effect observed for the mean colouring. The plots also indicate that when an image is split into more segments, the ineffectiveness of the mean colouring approach gets magnified due to the increased colour uniformity of individual super-pixels – a “blurring” effect. See Appendix C.1 and Figure C.1 for more details. . . . .	86
3.11	Some hyper-rectangles $(x', y')$ – created with discretisation – become indistinguishable in the binary interpretable representation $(x^*, y^*)$ of tabular data. The $\star$ marker indicates the explained instance and the background shading illustrates unique binary (IR) encodings. . . . .	87
3.12	Example of interpretable representation transformation in both directions for tabular data. Panel (a) depicts the discretisation and binarisation steps required to represent a data point as a binary on/off vector, and Panel (b) illustrates this procedure in the opposite direction. The forward transformation is <i>deterministic</i> given a fixed discretisation (binning of numerical features), however moving from the IR to the original domain is <i>non-deterministic</i> and requires random sampling. . . . .	88
3.13	Effect of different sampling algorithms on the <i>locality</i> and <i>diversity</i> of the sample when applied to the original domain of the Iris data set. The panels are plotted along <i>sepal length (cm)</i> on the x-axis and <i>sepal width (cm)</i> on the y-axis, and the black dot represents the explained instance for which the sample is generated. Red, blue and green markers – the three classes of the Iris data set – capture black-box predictions. These experiments show the advantage of sampling algorithms that are aware of the class distribution – (c) <i>Mixup</i> and (d) <i>normal class discovery</i> [160] – which allows them to generate a diverse sample that discovers the local decision boundary. This information helps the surrogate to better approximate the behaviour of the black box in the explained neighbourhood, thus improving explanation faithfulness. . . . .	91
3.14	Comparison of linear and tree-based local surrogates for a toy tabular data set. The background shading represents the probability of the blue class predicted by the local (a) ridge regression and (b) regression tree surrogate models built to explain the instance marked with the black dot. The encoding of values predicted by these surrogates is given by the adjacent colour-bar. Since the output of a linear model (a) is unbounded, the predicted values may be outside of the expected $[0, 1]$ range. . . . .	98

3.15	Various approaches to quantitative evaluation of surrogate models based on their faithfulness with respect to the underlying black box are possible. These metrics determine the ability of the surrogate (the red lines in the panels above) to mimic the predictions of the explained model (the green contours) by measuring its <i>fidelity</i> within a certain region. We can either compute <i>global</i> (a&c) or <i>local</i> (b&d) faithfulness with respect to the location of the <i>explained instance</i> (a&b) or the ( <i>closest</i> ) <i>decision boundary</i> (c&d). . . . .	99
4.1	Visualisation of a classification tree trained on the Iris data set. It is an example of a decision tree-specific transparency approach that helps the explainee to comprehend the (global) behaviour of such models. Its complexity grows with the size of the tree and it may not be suitable for a lay audience who lacks relevant machine learning knowledge. . . . .	110
4.2	Bar plot depicting feature importance extracted from a classification tree trained on the Iris data set. It is an example of a decision tree-specific transparency approach that helps the explainee to comprehend the overall (global) importance of data attributes. In many cases, interpreting the plot only requires being familiar with the meaning of the underlying features. . . . .	111
4.3	Visualisation of logical rules – presented as a conjunction of logical conditions – extracted from root-to-leaf paths of a classification tree trained on the Iris data set. It is an example of a decision tree-specific transparency approach that helps the explainee to understand conditions imposed on data attributes that lead to a particular prediction (a local or cohort explanation). In many cases, interpreting the figure only requires being familiar with the meaning of the underlying features. The vertical number to the left of each logical condition reports the importance of the corresponding feature. . . . .	111
4.4	Visualisation of two setosa class exemplars extracted from the training data assigned to a selected leaf of a classification tree fitted to the Iris data set. It is an example of a decision tree-specific transparency approach that helps the explainee to understand similarities between instances grouped together (within a single leaf) by the underlying tree (a local or cohort explanation). In many cases, interpreting the figure only requires being familiar with the meaning of the data features, however it is up to the explainee to reason about the connections between the output instances. Since some of the features may be redundant – i.e., not conditioned on the root-to-leaf path leading to the selected leaf – they should be marked as such (indicated with <i>N/A</i> in this figure) to avoid misleading the explainee. . . . .	112

- 4.5 Visualisation of a what-if explanation extracted from a classification tree trained on the Iris data set. The left part of the plot depicts an instance selected to be explained, which is classified as *setosa*. The data point to the right has two of its attribute values modified by an explaine (red and blue/green shading), thus posing a what-if question leading to a *versicolor* prediction. This (local) explanation is an example of a decision tree-specific transparency approach that helps the explaine to understand influence of selected feature values on a tree prediction. In many cases, being familiar with the meaning of the data features is sufficient to interpret the figure. Often, only the differentiating factors are highlighted, i.e., the foil, to make the explanation sparse, however this example lists all of the features given their low number. Additionally, the visualisation indicates which feature value changes proposed by the user are meaningful – green shading – given that some attributes may not appear on the root-to-leaf path responsible for predicting the what-if instance or be used by the tree altogether (captured by blue shading and *N/A* markers to the left of each box). The change of prediction is shown by orange tint. . . . . 113
- 4.6 Balanced decision tree of depth  $d = 2$  with three splits  $s_i$  applied to two features  $x_i$  resulting in four leaves  $l_i$ . Feature  $x_1 \in \{\text{red, green, blue}\}$  is categorical, and feature  $x_2 \in \mathbb{Z}$  is numerical. Branching left corresponds to satisfying the split's logical condition ( $s_i = \text{True}$ ), and branching right denotes failing it ( $s_i = \text{False}$ ). . . . . 115
- 5.1 An example of a multi-output regression tree used to explain an image (taken from Figure 5.10) labelled as *tennis ball* by a black-box deep neural network image classifier. The super-pixels, i.e., segments, shaded in *blue* are not important to the explanation at any given tree node. A super-pixel which value is 0 in the interpretable representation is “removed” by occluding it with a solid black colour. A super-pixel assigned 1 in the interpretable representation is preserved. The probabilities estimated by the surrogate tree usually do not sum up to 1 in each tree node as these values may only represent a subset of modelled classes and are a result of a regression, thereby should not be treated as probabilities. . . . . 128
- 5.2 Visual decomposition of a surrogate explanatory process for image data based on the LIME algorithm. The steps include generating an interpretable representation (b) and presenting an explanation in two different formats: a bar plot (c) and an image mask (d). . . . . 129
- 5.3 Black-box predictions for a single segment (#3) using different occlusion techniques. The chosen super-pixel is the most important part of the image according to its LIME explanation shown in Panel 5.2c. (This figure is an altered reproduction of Figure 3.3.) 133

## LIST OF FIGURES

---

5.4	LIME explanation for the Husky image (Panel 5.2a) when using black occlusions. It was generated based on the same interpretable representation and (binary) data sample as the explanation for the mean-colour occlusion presented in Panel 5.2c, making them directly comparable. (Segment #5 is the one below #1.) . . . . .	135
5.5	LIME explanations for the top three classes predicted by a black-box model for the image shown in Panel (a): <i>tennis ball</i> with 99.56% (b), <i>golden retriever</i> with 0.42% (c) and <i>Labrador retriever</i> with 0.02% (d). . . . .	143
5.6	Three types of LIMETree explanations: (a) feature importance, (b) what-if explanation and (c–d) exemplar explanation. . . . .	144
5.7	The shortest LIMETree explanations of <i>tennis ball</i> . . . . .	145
5.8	Visual representation of a LIMETree rule explanation that maximises the <i>Labrador retriever</i> prediction (99%). . . . .	145
5.9	Customised (personalised) counterfactual explanations generated with LIMETree. . . . .	146
5.10	Default (a) and custom (b) interpretable representations of an image. The top two classes predicted by a black box are 99.6% <i>tennis ball</i> and 0.4% <i>golden retriever</i> . . . . .	147
5.11	Fidelity of the surrogate (y-axis) plotted against the depth-based complexity of the tree (x-axis), i.e., the ratio between the tree depth and the number of interpretable features. The results are computed for the top three classes predicted by the black box. Panels (a), (b) & (c) depict the LIME loss (Equation 5.2); and Panels (d), (e) & (f) depict the LIMETree loss (Equation 5.4). Note the different scales on the y-axes. . . . .	156
6.1	Surrogate explainers of image classifiers require an interpretable representation, such as super-pixel segmentation, to effectively communicate the explanation to the user. These explainers try to identify portions of an image that influence its classification the most, i.e., segments of high positive or negative importance. Since the default outcome of image segmentation (a) may be unintuitive, we encourage the explainee to personalise the segmentation (b), e.g., by merging its elements, such that it represents (semantically) meaningful concepts. . . . .	163
6.2	Glass-Box design and information flow. . . . .	168
6.3	Example explanatory conversation between Glass-Box and an explainee who personalises the explanations by asking counterfactual questions. . . . .	169
7.1	Depiction of the bLIMEy meta-algorithm and framework (Chapter 3) for developing bespoke surrogate explainers. It consists of: interpretable data representation, data sampling and explanation generation steps. . . . .	190
7.2	Possible use cases of the explainable artificial intelligence taxonomy (Chapter 2) include: fact sheets, work sheets, check lists and a reference for certification or standardisation procedures. . . . .	193

B.1	Typical architecture exhibited by academic software landscape – standalone code-bases, distributed with unnecessary dependencies, offering incompatible and non-standard APIs. . . . .	229
B.2	Modular architecture of FAT Forensics. The input requirements for data sets and predictive models are kept to a minimum and are very flexible: 2-dimensional NumPy arrays and Python objects with <code>fit</code> , <code>predict</code> and, optionally, <code>predict_proba</code> methods respectively. The FAT functionality is composed with atomic building blocks, making the process of constructing new tools, or creating variants of existing algorithms, as easy as connecting the right components. . . . .	230
C.1	Mean squared error calculated between the top prediction of an image (probability estimate) and predictions of the same class when progressively occluding a higher number of segments with a given colouring strategy. We use eight different approaches, the RGB (Red, Green, Blue) colour encodings of which are: white (255, 255, 255); black (0, 0, 0); red (255, 0, 0); green (0, 255, 0); blue (0, 0, 255); pink (255, 192, 203); random – drawn from a uniformly distributed colour space separately for each super-pixel of an individual image; and mean – each segment is occluded with its mean RGB colour. The panels show that the mean occlusion strategy is not as effective at hiding information from the black box as using a single colour for all of the super-pixels (regardless of the colour choice). Similarly, randomising the occlusion colour for each individual segment does not seem to have the detrimental effects observed for the mean colouring. The plots also reveal that when an image is split into more segments, the ineffectiveness of the mean-colouring approach gets magnified due to the increased colour uniformity of individual super-pixels. (See Appendix C.1 for the description of our experimental setup.) . . . . .	240
C.2	Interpretable representations based on decision trees achieve higher purity of hyper-rectangles (y-axes, lower is better) with fewer encodings (x-axes), i.e., they are more flexible and expressive. The number of unique encodings used by quartile-based IRs is constant for a data set and it is displayed in the legend (presented as the number of encodings used, out of the theoretical limit supported by the representation); whereas for tree-based IRs, it is equivalent to the number of leaves, which is recorded on the x-axes. Panels (c) and (d) do not capture the tree width at which this IR <i>globally</i> outperforms the quartile-based IR, which is 80 (compared to 441) and 224 (compared to 428) respectively for the <i>housing</i> and <i>diabetes</i> data sets. For more details, see the <i>Interpreting the Results</i> paragraph in Appendix C.2. . . . .	243
C.3	Example of discrete ( $x'_1, x'_2$ ) and binary ( $x_1^*, x_2^*$ ) interpretable representations of tabular data. $\star$ represents the explained instance. . . . .	250



## GLOSSARY

<b>AI</b>	Artificial Intelligence
<b>anchors</b>	local model-agnostic prediction explainer based upon high-precision rules (“sufficient” conditions) [130]
<b>API</b>	Application Programming Interface
<b>BAM</b>	Benchmarking Attribution Method [181]
<b>BETA</b>	Black-box Explanations through Transparent Approximations [85, 86]
<b>bLIMEy</b>	build LIME yourself [Chapter 3]
<b>BSD 3-Clause</b>	permissive code licence, imposing minimal restrictions on the use and distribution of covered software
<b>CART</b>	Classification And Regression Trees [23]
<b>CRISP-DM</b>	Cross-Industry Standard Process for Data Mining [26, 103]
<b>CtreeX</b>	Contrastive tree eXplainer [Chapter 4]
<b>DARPA</b>	Defense Advanced Research Projects Agency
<b>DT</b>	Decision Tree
<b>FACE</b>	Feasible and Actionable Counterfactual Explanations [123]
<b>FAT</b>	Fairness, Accountability and Transparency
<b>FAT Forensics</b>	Python toolkit for evaluating fairness, accountability and transparency of AI systems ( <a href="https://fat-forensics.org/">https://fat-forensics.org/</a> ) [Appendix B]
<b>GDPR</b>	General Data Protection Regulation
<b>Glass-Box</b>	interactive conversational voice-driven counterfactual explainer [Chapter 6]
<b>HCI</b>	Human–Computer Interaction

<b>ICE</b>	Individual Conditional Expectation [48]
<b>IEEE</b>	Institute of Electrical and Electronics Engineers – a professional association for electronic and electrical engineering
<b>ImageNet</b>	large database of images used in visual object recognition research [33]
<b>IML</b>	Interpretable Machine Learning
<b>Inception v3</b>	convolutional neural network for object detection and image classification
<b>IR</b>	Interpretable Representation
<b>K-LASSO</b>	selecting K features with LASSO using the regularisation path [36, 129]
<b>KDD</b>	Knowledge Discovery in Database [38]
<b>LASSO</b>	Least Absolute Shrinkage and Selection Operator [165]
<b>LIME</b>	Local Interpretable Model-agnostic Explanations [129]
<b>LIMETree</b>	tree-base surrogate explainer built upon bLIMEy [Chapter 5]
<b>Mixup</b>	class-aware sampling algorithm for tabular data [182]
<b>ML</b>	Machine Learning
<b>MSE</b>	Mean Squared Error
<b><i>n</i>-gram</b>	contiguous sequence of <i>n</i> items, e.g., words in a sentence
<b>OLS</b>	Ordinary Least Squares
<b>PD</b>	Partial Dependence [44]
<b>PI</b>	Permutation Importance [22]
<b>PyTorch</b>	Python machine learning library used for computer vision and natural language processing applications [116]
<b>RGB</b>	colour space in which individual colours are encoded as a number triplet in the 0–255 range signifying the intensity of Red, Green and Blue component respectively, e.g., (0, 0, 255) is pure blue
<b>RuleFit</b>	rule-based explanations scored with sparse linear models [45]
<b>scikit-image</b>	collection of algorithms for image processing implemented in Python [170]

---

<b>scikit-learn</b>	Python machine learning library used mainly for clustering, classification and regression tasks on tabular data [120]
<b>SHAP</b>	SHapley Additive exPlanations [100]
<b>SLIC</b>	image segmentation algorithm based on k-means clustering in the RGB colour space [2]
<b>TCAV</b>	Testing with Concept Activation Vectors [71]
<b>TREPAN</b>	tree-based surrogate explainer mimicking global behaviour of neural networks [28]
<b>XAI</b>	eXplainable Artificial Intelligence



## WHAT, WHY AND HOW OF EXPLAINABLE AI

**T**ransparency, interpretability and explainability engender understanding and confidence. As a society, we strive for transparent governance and justified actions that can be scrutinised and contested. Such a strong foundation provides a principled mechanism for reasoning about fairness and accountability, which we have come to expect. While widely applicable to our society, artificial intelligence systems are not universally held up to the same standards. This becomes problematic when such systems permeate to applications that either implicitly or explicitly affect peoples' lives, for example in banking, parole hearings, job screenings or school admissions. In such cases, creating explainable predictive models or retrofitting transparency to preëxisting algorithms is usually expected by the affected individuals, or simply required by law. A number of techniques and algorithms are being proposed to this end, however as a relatively young research area, there is no consensus on a suite of technology addressing these challenges.

### 1.1 Black-Box Artificial Intelligence

The term *black box* can be used to describe a system whose internal workings are opaque to the observer – its operation may only be traced by analysing its inputs and outputs [13, 24]. Similarly, in computer science, Artificial Intelligence (AI) and Machine Learning (ML), black box is a (data-driven) algorithm that can be understood as an automated process that we cannot reason about beyond observing its behaviour. For AI in particular, Rudin [133] points out two main sources of opaqueness:

1. a *proprietary* system, which may be transparent to its creators, but operates as a black box; and

2. a system that is too *complex* to be comprehend by *any human*.

While the latter case concerns systems that are universally opaque for the *entire population*, essentially, this definition of black boxes establishes a *spectrum of understanding* in contrast to a binary quantification [32].

Perception and comprehension of a phenomenon depend upon the observer’s cognitive capabilities and mental model, which is an internal representation of this phenomenon built on real-world experiences [80]. For example, Kulesza et al. [80] outline a *fidelity*-based understanding spectrum spanning two dimensions:

**completeness** how truthful the understanding is overall (generality); and

**soundness** how accurate the understanding is for a particular phenomenon (specificity).

Therefore, a *complete* understanding of an event from a certain domain is equivalently applicable to other, possibly unrelated, events from the same domain. A *sound* understanding, on the other hand, accurately describes an event without (over-)simplifications, which may result in misconceptions. Striking the right balance between the two depends upon the observer and may be challenging: completeness without soundness is likely to be too broad, hence uninformative; and the opposite can be too specific to the same effect.

Within this space, Kulesza et al. [80] identify two particularly appealing types of a mental model:

**functional** which is enough to operationalise a concept but does not necessarily entail the understanding of its underlying mechanism (akin to The Chinese Room Argument [139]); and

**structural** which warrants a detailed understanding of how and why a concept operates.

For example, a functional understanding of a switch and a light bulb circuit can just be the dependency between flipping the switch and the bulb lighting up. Whereas, a structural understanding of the same phenomenon may focus on the underlying physical processes, e.g., closing an electrical circuit allows electrons to move, which heats up the bulb’s filament, thus emitting light. The former understanding is confined to operating a light switch, while the latter can be generalised to many other electrical circuits. Each one is suitable for a different audience and their complexity should be fine-tuned for the intended purpose as explanations misdirected towards an inappropriate audience may be incomprehensible, leaving the system in question opaque.

Making AI systems intelligible faces similar challenges, especially given their varied, and sometimes ambiguous, audience [72, 124], purpose [56] and application domain [18]. While intelligent systems are often deemed (unconditionally) opaque, it is not a definitive property and it largely depends on all of the aforementioned aspects, some of which fall beyond the

standard AI development lifecycle. Without clearly defined explainability desiderata addressing them can be challenging, in contrast to designing AI systems purely based on their predictive performance, which is often treated as a quality proxy and can be universally measured, reported and compared. In view of this disparity, many engineers (incorrectly) consider these two objectives as competing [133], thus choosing to pursue high predictive performance at the expense of opaqueness, which may be incentivised by business opportunities.

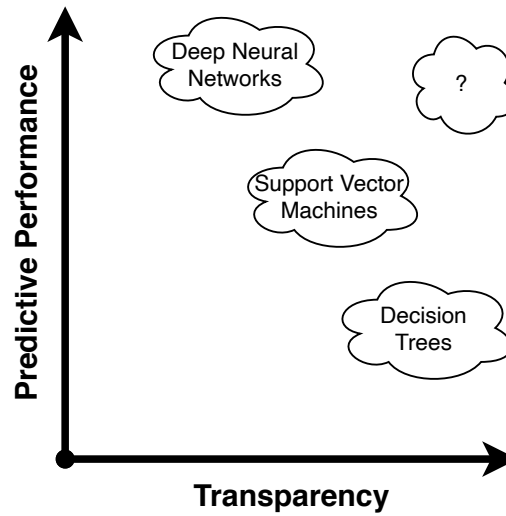
While high predictive power of an AI system makes it useful, its explainability is equally important. The pervasiveness of automated decision-making in our everyday life, some of which with social consequences, requires striking a balance between the two that is appropriate for what is at stake, e.g., approaching differently a car autopilot and an automated food recommendation. A different domain that could benefit from powerful and explainable AI is (scientific) discovery – intelligent systems may achieve super-human performance, e.g., AlphaGo [143], however a lack of transparency renders their mastery and ingenuity unattainable. Such observations have prompted the Defense Advanced Research Projects Agency (DARPA) to announce the eXplainable AI (XAI) programme [53, 54] that promotes building a suite of techniques to:

- create more explainable models, while preserving their high predictive performance; and
- enable humans to understand, trust and effectively manage intelligent systems.

We address this call by taking a closer look at explainability by design and removing opaqueness from preëxisting black boxes. To this end, we develop an *explainability taxonomy* (Chapter 2) to reason about such systems within a well-defined framework, which spans five distinct dimensions of social and technical requirements: functional, operational, usability, safety and validation. It covers human aspects of explanations, thus giving us a platform to consider the audience (explainees), explanation complexity and the interaction mode, among many others. This diversity of, sometimes competing, objectives prompts us to investigate an *explanatory process* akin to conversational explanations between humans. In such a scenario, an explainee can customise and contest various aspects of an opaque system within a congruent interaction that enables explanation personalisation (Chapter 6). Additionally, our taxonomy comprises of technical aspect of explanations such as compatibility with predictive models and information leakage, thereby allowing to judge their suitability for the problem at hand.

Furthermore, we show how transparency, and other terms often used to denote similar concepts, can be differentiated from explainability – both overcome opaqueness, but only the latter leads to understanding – which we exemplify with decision trees (Chapter 4). With this goal in mind, we develop two explainability techniques:

- contrastive and supportive explanations of decision trees (Chapter 4) that are inspired by explainability research in the social sciences [106]; and



**Figure 1.1:** Fictitious depiction of an anecdotal trade-off between transparency and predictive power of AI systems.

- a model-agnostic surrogate explainer based on multi-output regression trees that is capable of delivering the same explanation types for any predictive black box (Chapter 5).

The former is ante-hoc as the same model is used for predicting and explaining, thus it is specific to decision trees, whereas the latter is post-hoc since a simpler surrogate model, which mimics the behaviour of the black box, is used to generate explanations, making it model-agnostic. While the design of the tree explainer guarantees faithfulness, the fidelity of post-hoc methods is a well-known issue, which we analyse and address for tree-based surrogates. Creating a faithful surrogate requires choosing specific components of otherwise highly-modular explainability framework, which we develop in Chapter 3, where we also discuss its building blocks, parameterisation and trade-offs.

### 1.1.1 Trading off Transparency for Predictive Power

A common belief perpetuating the XAI community and motivating many methods published in the literature is the universal *dichotomy* between transparency and predictive power of AI systems. A popular example supporting this hypothesis is the unparalleled effectiveness of deep neural networks, whose ever increasing complexity, e.g., the number of layers and hidden units, improves their performance at the expense of transparency. This trade-off has been reiterated in the DARPA XAI program’s Broad Agency Announcement [53] and supported by an appealing graph reproduced in Figure 1.1. However, it is a *theory* based mostly on anecdotal evidence [133], with Rudin [133] criticising plots like Figure 1.1 given their lack of scale, transparency or performance metrics, and supporting data. Notably, Rudin [133] argues that investing more effort

into feature engineering can help to build inherently explainable AI systems that perform on a par with their black-box competitors [27].

This anecdotal trade-off and a tendency to focus on the predictive power alone mean that explainability is often only an afterthought. Such a mindset contributes to an AI landscape with an abundance of well-performing but inherently opaque algorithms that are in need of explainability, thus creating a demand for universal explainers that are post-hoc and model-agnostic, such as surrogates. This seemingly uncompromising development approach where state-of-the-art performance remains the main objective that is later complemented with a post-hoc explainer creates an attractive alternative (and rebuttal) to designing inherently explainable AI system, whose creation arguably requires more effort. While such explainers are compatible with any black-box model, they are not necessarily equally well suited for all of them – after all the computer science folklore of “no free lunch” applies here as well. Some post-hoc and model-agnostic explainers boast appealing properties and guarantees, however upon closer inspection caveats and assumptions required for them to hold, such as the underlying “black box” being a linear model [100], can often be found. Making an explainer model-agnostic introduces an extra layer of complexity that often entails a degree of randomness and lacklustre fidelity [158, 183], in which case using them becomes a stopgap to claim explainability of an inherently opaque AI system.

In Rudin’s [133] view, many high-stakes AI systems can be explainable by design with enough effort put towards data pre-processing and feature engineering (which otherwise, e.g., for neural networks, may go into architecture search and parameter tuning – a phenomenon humorously known as *graduate student descent*). Such ante-hoc explainers are usually domain-specific and after the initial engineering endeavour they are easy to manage and maintain. While such an approach should be championed for structured (tabular) data where it has been shown to perform on a par with state-of-the-art black boxes [27], the same may be unachievable for sensory data such as images and sounds, for which opaque models, e.g., deep neural networks, have the upper hand. In addition to black boxes modelling sensory data, preexisting, inaccessible or legacy AI systems may require interpretability, in which case they can be retrofitted with post-hoc explainers. However, falling back on off-the-shelf solutions may not guarantee advertised fidelity [158] (in particular, soundness and completeness), which is of paramount importance and may require tailor-made explainers and transparent communication of their limitations. We briefly discuss ante- and post-hoc explainability in Section 1.3 and cover it extensively, along other important properties, in the taxonomy introduced in Chapter 2.

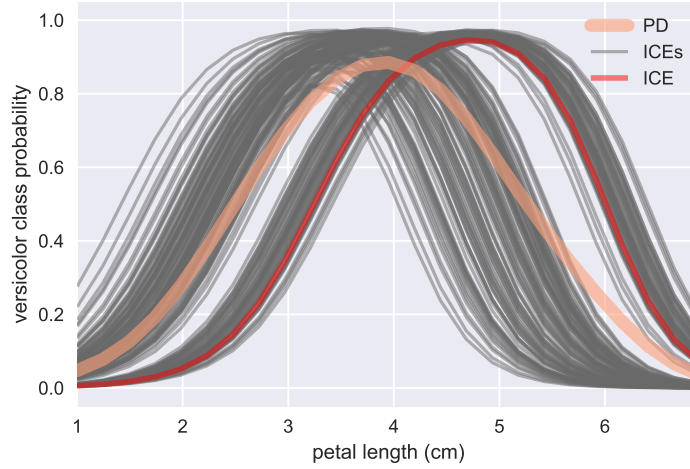
### 1.1.2 Explaining the Machine Learning Process

So far we have mostly focused on explaining predictions and actions of intelligent systems since they are observable and can be related to by a wide range of explainees regardless of their background. However, automated predictions are just artefacts of a more elaborate AI

or ML process that additionally consists of *data* and *models*, where any part of this pipeline may be opaque and in need of explaining. Predictive artificial intelligence and machine learning processes manipulate *data* to learn *models* that generalise well and are capable of predicting (previously unseen) instances [153, 154]. Explaining data may be challenging without any modelling assumptions, in which case it is mostly limited to summary statistics and descriptive properties. On the other hand, models and predictions are more apt to be explained with a wide range of diverse explanation types such as contrastive, exemplar and attribute importance. Since AI and ML processes are directional – from data, through models, to predictions – the latter components depend on the former, which also applies to their respective explanations. For example, if data attributes are incomprehensible, explanations of models and predictions expressed in terms of these features will also be opaque. To better understand the intricacies of such processes, we need methods and tools capable of peeking inside these three components.

**Data** can be considered the most difficult component to explain as doing so usually requires us to first develop a mental model of the underlying phenomenon itself. Therefore, there may not necessarily exist a pure data explanation method except simple *summary statistics*, such as class ratio or per-class feature distribution, and *descriptors*, e.g., “the classes are balanced”, “the data are bimodal” and “these features are highly correlated”. Note that the former simply state well defined properties and may not be considered explanations, whereas the latter can be contrastive and lead to understanding. Importantly, data are already a model – they express a (subjective and partial) view of a phenomenon and come with certain assumptions, measurement errors or even embedded cultural biases (e.g., “How much is a lot?”). “Data statements” [15], “data sheets” [47] and “nutrition labels” [60] attempt to address such concerns by capturing these (often implicit) assumptions. As a form of data explanations, they characterise important aspects of data and their collection process in a coherent way, e.g., experimental setup, collection methodology (by whom and for what purpose), applied pre-processing (cleaning and aggregation), privacy aspects and the data owners, among many others.

**Models** as a whole or their parts (e.g., specific subspaces or cohorts) can be explained to engender a general understanding of their functionality. While some models may be inherently transparent, e.g., shallow decision trees, their simulatability [96] – the explaineer’s ability to simulate their decisive process mentally *in vivo* – may not necessarily warrant understanding. (Recall the differentiation between the functional and operational mental models and The Chinese Room Argument [139].) Popular model explanations include feature importance [22, 40], feature influence on predictions [44], presenting the model in cognitively-digestible portions [76, 146] and model simplification [28] (e.g., mimicking its behaviour or a global surrogate). Since not all models operate directly on the input features, an *interpretable representation* may be necessary to convey an explanation, e.g., a super-pixel segmentation of an image [129]; alternatively, if the data are comprehensible, landmark exemplars can be used to explain the behaviour of a model



**Figure 1.2:** An explanation of a model’s prediction of the *versicolor* class when varying the *petal length* feature value for the Iris data set [41]. ICE of a selected instance is plotted in red; grey represents ICEs of all the training data; and orange is the PD of the model achieved by averaging all the individual ICEs.

or its parts [68, 70]. Regardless of the explanations’ type and scope, they should always lead to *understanding* in addition to presenting truthful and accurate behaviour of a model.

**Predictions** are explained to communicate a rationale behind a particular decision of a model. Depending on the explanation type, a range of diverse aspects concerning the model’s decisive process can be provided to the explainee. For example, the user may be interested in feature importance [22], feature influence [100, 129], relevant data examples [69] and training points [74], or contrastive statements [123, 173], to name a few. Note that while some of these explanation types are similar to model explanations, here they are explicitly generated with respect to a single data point and may not necessarily generalise beyond this particular case, whereas for model explanations they convey similar information for all data (i.e., the model). A good example of this duality is information communicated by Individual Conditional Expectation [48] (ICE) and Partial Dependence [44] (PD), both of which are feature influence explanations – the first with respect to a single data point and the latter concerning a whole data set. ICE measures a change in response of a predictive model when modifying a selected feature of a data point of interest; and PD averages such responses for a collection of data points, e.g., the training set of this model, to approximate its overall behaviour – see Figure 1.2. Similar to model explanations, the information can be conveyed in the raw feature space or, when the original domain is unintelligible, using an interpretable representation. Finally, multiple explanation types spanning data, models and predictions can be bundled together in a shared user interface to provide a multi-faceted view on behaviour of an AI or ML system [76, 77, 178].

With such a diverse range of explanations, their presentation media also differ [153, 154]. A simple approach to characterise an AI component is (statistical) *summarisation* – it is commonly used for describing properties of data with numerical tables and vectors, which can be difficult to digest for non-experts. *Visualisation* – a graphical representation of a phenomenon – is a more advanced, insightful and flexible analytical tool. Static figures communicate information in one direction, similar to summarisation; however, creating interactive plots can facilitate a “dialogue” with an explainee, thereby catering to a more diverse audience. Visualisations are often supported by a short narrative in the form of a caption, which increases their informativeness. *Textualisation* – a natural language description of a phenomenon – can express concepts of higher complexity and dimensionality than plots, which can help to overcome the curse of dimensionality and the inherent limitation of the human visual system to, arguably, three dimensions. Communicating with text enables a true dialogue and has been shown to be more insightful and effective than presenting raw, numerical and visual data [122], which can accompany the narrative to improve its expressiveness. A further refinement of textualisation is formal *argumentation* [35] – a structured and logically-coherent dialogue accounting for every disputable statement and giving the explainee an opportunity to contest the narrative, thus providing explanations leading to understanding rather than informative descriptions. The type of presentation medium and the communication protocol between the explainer and the explainee are just two properties of XAI systems, which we discuss in more detail in the taxonomy introduced in Chapter 2.

### 1.1.3 Transparency Is Not the Preferred Nomenclature

A monumental amount of research into explainable AI and Interpretable ML (IML) published in recent years may suggest that it is a freshly established field, however in reality it is more of a renaissance. While work in this area indeed picked up the pace in the past decade, interest in creating transparent and explainable, data-driven algorithms dates back at least to the 1990s [133], and further back to the 1970s if expert systems [92] are taken into account. With such a rich history and the increased publication velocity attributed to the more recent reestablishment of the field, one may think that this research area has clearly defined objectives and a widely shared and adopted terminology. However, with an abundance of keywords that are often used interchangeably in the literature – without precisely defining their meaning – this is not yet the case. The most common terms include, but are not limited to:

- explainability,
- explicability,
- interpretability,
- observability,
- intelligibility,
- simulatability,
- transparency,
- comprehensibility,
- justification,

- evidence,
- reason, and
- cause.

While early research might have missed out on an opportunity to clearly define its goals and nomenclature, recent work [19, 133] has attempted to tackle this problem. Biran and McKeown [19] were concerned with *explanations*, which they characterised as “giving a reason for a prediction” and answering “how a system arrives at its prediction”. They also defined *justifications* as “putting an explanation in a context” and conveying “why we should believe that the prediction is correct”, which, they note, do not necessarily have to correspond to how the predictive system actually works. Another important term is *cause*, which may not be used that often in the XAI and IML literature, however it should be reserved for insights extracted from causal models [118]. More recently, Rudin [133] defined *interpretability* as a domain-specific notion that imposes “a set of application-specific constraints on the model”, thus making this notion only applicable to predictive models that can provide their own explanations (i.e., ante-hoc interpretability). Therefore, in Rudin’s view a predictive model is interpretable if it “obeys structural knowledge of the domain, such as monotonicity, causality, structural (generative) constraints, additivity or physical constraints that come from domain knowledge”, which positions it on a *transparency* spectrum. Finally, Rudin [133] objects to using the term *explanation* when referring to “approximations to black box model predictions” (i.e., post-hoc explainability).

Each definition conveys a more or less precise meaning that can be used to label relevant techniques, however they do not necessarily clarify and help to navigate the complex landscape of IML and XAI research. In our work, we categorise this terminology based on three aspects:

- *properties* of systems,
- *functions* and *roles* which they serve, and
- *actions* required to process and assimilate them.

The core concept around which we build our nomenclature is **explainability**, which we define as **insights that lead to understanding** (the **role** of an explanation) – a popular rationale in the social sciences [12, 73, 97]. While it may seem abstract, understanding can be assessed with questioning dialogues [10, 101, 174–176] – e.g., a machine questioning the explainees to verify their understanding of the phenomenon being explained – which are the opposite of explanatory dialogues. Such a process reflects how understanding is tested in education, where the quality of tuition and knowledge of pupils is evaluated through standardised tests and exams (albeit not without criticism [104]). Furthermore, encouraging people to explain a phenomenon helps them to realise the extent of their ignorance and confront the complexity of the problem, which are important factors in uncovering The Illusion of Explanatory Depth [132] – a belief that one understands more than one actually does.

This notion of explainability and the three building blocks of XAI and IML terminology allow us to precisely define the other popular terms. Therefore,

- *observability*,
- *transparency*,
- *explicability*,
- *intelligibility*,
- *comprehensibility*, and
- *interpretability*

are **properties** of an AI system. They can convey information of varied complexity, *understanding* which depends upon the cognitive capabilities and (domain) expertise of the explainee. For example, observing an object falling from a table is a transparent phenomenon per se, but the level of its understanding, if any, is based upon the depth of observer’s physical knowledge, which underpins understanding. This transparency provides

- *evidence*,
- *reason*, and
- *justification*

(**roles**) that can be used to

- *reason* about,
- *interpret*, or
- *comprehend*

(note that here the three are used as verbs) behaviour of a black box, all three of which are **actions** that possibly lead to understanding. While *simulatability* (**action**) is also based upon observing a transparent behaviour and replicating it, such an action does not necessarily imply understanding of the underlying phenomenon – recall the difference between structural and functional mental models [80] and The Chinese Room Argument [139] discussed in the introduction. Lastly, a *cause* has a similar meaning to a *reason*, but the first one is derived from a proper causal model, whereas the latter is based purely on observation of the black-box model’s behaviour.

Such a setting paints an appealing dependency between the XAI and IML terminology where:

$$\text{Explainability} = \underbrace{\text{Reasoning}(\text{Transparency} | \text{Background Knowledge})}_{\text{understanding}},$$

which defines Explainability as the **process** of deriving *understanding* through Reasoning applied to Transparent insights from the black box adjusted to the explainee’s Background Knowledge. In this process, the Reasoning can either be done by the explainer or the explainee, and there is an implicit assumption that the explainee’s Background Knowledge aligns with the Transparent representation of the black box. If the latter is not the case, mitigation techniques such as employing an *interpretable representation* (see Chapter 3 for more details) can be used to communicate concepts that are otherwise incomprehensible. Reasoning also comes in many different shapes and sizes, depending on the underlying black box (Transparency) as well as the explainer and the explainee (Background Knowledge), for example:

- logical reasoning with facts,
- causal reasoning over a causal graph,
- case-based reasoning with a fixed similarity metric, and

- artificial neuron activation analysis for a *shallow* neural network.

Therefore, linear models are transparent (assuming a reasonable number of features), and with the right ML and domain background knowledge – requirement of normalised features, effect of feature correlation and the meaning of coefficients – the explainee can reason about their properties, leading to an explanation based on understanding. Similarly, visualisation of a *shallow* decision tree can be considered both transparent and explainable given that the explainee understands how to navigate its structure (ML background knowledge) and the features are meaningful (domain background knowledge); again, it is up to the explainee to reason about these insights. When the size of a tree increases, however, its visualisation loses the explanatory power because many explainees will be unable to process and reason about its structure. In this case, restoring the explainability of a deep tree requires delegating the reasoning process to an algorithm that can digest and output sought after insights in a concise representation. For example, when explaining a prediction, the tree structure can be traversed to identify a similar instance with a different prediction, e.g., as encoded by two neighbouring leaves with a shared parent, thus demystifying the automated decision – more on that in Chapter 4.

## 1.2 Humans and Explanations: Two Sides of the Same Coin

Defining explainability as “leading to understanding” and our categorisation into *properties*, *functions* and *actions* highlight an important aspect of this research topic: explanations are directed at some autonomous agent, either a human or machine, which is as important as the explainability algorithm itself. Notably, up until recently XAI and IML research has been undertaken predominantly within the computer science realm [107], thus bringing in numerous biases and implicit assumptions from this overwhelmingly technical field. While some explainability research has found its way into other scientific outlets, e.g., law [173], its considerable part gravitated purely around technical properties, which resulted in explaining AI for the sake of (possibly undefined) explainability. This research agenda was disrupted by Miller et al. [107], who observed that the function of an explanation and its recipients are largely neglected – a phenomenon which they dubbed “inmates running the asylum” – leading to a substantial paradigm shift. Miller’s [106] follow-on work grounded this observation in (human) explainability research in the social sciences, where this topic has been studied for decades, thus providing invaluable insights that can benefit XAI and IML.

Miller’s findings have arguably reshaped the field, with a substantial share of the ensuing research acknowledging the explainees – their goals, expectations, intentions and interactions. While explainability of autonomous systems has widespread benefits (see Section 1.3), it is usually requested when an AI agent operates inconsistently with the explainee’s expectations or mental model, e.g., an unexpected ML prediction resulting in a disagreement. In such a case, explainees’ preferences and goals should be considered to cater to their needs and maximise the effectiveness

of an explanation, for example by appropriately adjusting its complexity [106]. This step can be improved by treating explainability as a process instead of one-off information offloading [106]; by satisfying the explainees’ natural desire to interact and communicate with the explainer within a predictable protocol, they are provided with an opportunity to customise and personalise the explanation [151]. Perhaps the most influential of Miller’s [106] observations is the humans’ preference for contrastive explanations given their predominance in our everyday life. In the following subsections, we discuss these three aspects of human-centred explainability in more detail, with a more comprehensive account of this topic presented as *usability requirements* (Section 2.2.3) in our XAI taxonomy, which is introduced in Chapter 2.

### 1.2.1 Explanation Audience

Understanding is an elusive objective when it comes to explaining intelligent systems since different explainees may expect the explanations to convey different information. When taken into account, the purpose of explainability and the explainee’s goal also affect the explanation composition. For example, an explanation will look differently when its purpose is to help debug an ML model as opposed to justify a negative outcome of a loan application; note that the target audience also differs, with the former aimed at ML engineers and the latter at lay people. In certain cases, such as the aforementioned loan application, the *actionability* of insights provided to the explainees is important, e.g., saying that one would receive a loan had he or she been 10 years younger is futile. Multiplicity of apparently indistinguishable arguments can also decrease the perceived quality of an explanation when one is chosen at random without a user-centred heuristic in place, which, again, depends on the domain and audience. For example, research suggests [106] that if one of multiple, otherwise equivalent, time-ordered events has to be chosen as an explanation, the most recent one will best resonate with the explainee.

Another, related guiding principle is explanation brevity, which helps to avoid overwhelming the explainee with (redundant) insights or details. An explanation that aims to resolve a disagreement between an explainee and a black-box model should only present evidence that is novel, as reiterating facts that the explainees have already acknowledged may be detrimental to their attention. Filtering out mundane and optimising for surprising explanation content helps the audience to discover missing pieces of information that may resolve the underlying disagreement or signify an unexpected bug in the predictive model. Addressing this desideratum, however, is complicated as it requires access to the explainee’s background knowledge and mental model, which are vague and often undefined concepts that cannot be easily extracted and processed. Accounting for the explainee’s cognitive capabilities and skill level, on the other hand, is more practical and allows to adjust the complexity of an explanation towards the anticipated audience. For example, a medical diagnosis can be expressed in terms of test results as opposed to observable symptoms when it is targeted at medical staff and not patients, thus leading to a desired level of understanding. Here, we implicitly assume that the explanation recipient is a human,

but it can as well be another algorithm that further processes such insights, in which case other, more appropriate properties may be of interest.

While useful, brevity of an explanation can sometimes be at odds with its comprehensiveness – sacrificing the big picture for concise communication [81]. An explanation that accounts for all aspects contributing to a particular black-box prediction is *complete*, however it may be too convoluted to understand. For example, a collection of logical conditions may be *sufficient* for a particular prediction, thereby constituting a complete explanation, but only their small subset is *necessary* to guarantee this prediction. Nonetheless, explanatory minimalism, which is at the other end of the spectrum, bears a danger of oversimplification. When brevity is a strict requirement, explanation *soundness* can be favoured to focus on factors pertinent to the explained instance and filter out more general ones that are largely irrelevant. Such an approach can introduce inaccuracies with respect to the overall black-box system, but remain truthful for the individual instance, e.g., removing some of the necessary conditions in the aforementioned example. Finding the right balance between generality and specificity of an explanation often requires tuning its soundness and completeness according to the intended audience and application.

### 1.2.2 Explanatory Process

While difficult to achieve for an AI explainer, satisfying this wide range of diverse assumptions and expectations comes naturally for humans when they engage in an explanatory process among themselves. This is partly due to shared background knowledge, nonetheless it would amount to nothing without the interactive communication that allows to rapidly iterate through questions and refine answers to arrive at understanding. One explanation does not fit all – as we show in Chapter 6 – and treating explainability as a bi-directional process provides a platform to appreciate uniqueness of each explainee with personalised explanations. Dialogue is fundamental to human explainability, however it is largely absent in XAI techniques [151], which are often based on one-way communication, where the user receives a description of the black box without an opportunity to request more details or contest it. A similar interaction in a form of the aforementioned questioning dialogues can also be used to judge the explainee’s understanding of the explained concept, thus be a proxy for assessing effectiveness of the explainer.

Designing interactive explainers that operate in accordance with users’ expectations may require an interdisciplinary approach borrowing from the Human–Computer Interaction (HCI) research. However, an intelligible interface and a natural communication protocol are just one aspect influencing user satisfaction. The other is an alignment of the explainer’s behaviour with the explainee’s goals and intentions. While the latter topic has not received much attention in the literature, we can draw design insights and inspirations from research on *explanatory debugging* of predictive models [81]. For example, such a process should be *iterative*, enabling the explainee to learn, provide feedback and receive updated insights until reaching a satisfactory conclusion.

The explainer ought to always *honour user feedback* by incorporating it into the explanation generation process, or clearly communicate a precise reason if it is impossible. Furthermore, a *reversible* communication will allow the explainee to retract a requested change or annul a piece of feedback when it was provided by mistake or to explore an interesting part of the black box. To easily attribute each piece of feedback to an explanation change, the whole process should be *incremental*, thereby showing up-to-date results even after small tweaks.

Human dialogue tends to be verbal or written, both of which are based on natural language. While ubiquitous, this form of communication is not equally effective in conveying all types of information. To overcome this challenge, humans augment it with visual aids, which are especially helpful when the interaction serves explanatory purposes. The same strategy can be adopted in XAI, where the explainer would switch between various explanatory artefacts, such as natural language, images, plots, mathematical formulation, numbers and tables, that are best suited for the type of information being communicated, i.e., the context. Mixing and combining them is also possible and sometimes may be beneficial as the whole can be greater than the sum of its parts, e.g., a numerical table or a plot complemented with a caption. Using visualisation, textualisation and (statistical) summarisation, however, does not mean that there is a coherent relation, structure or story conveyed by these communication media, which can possibly be achieved by grounding them in *logical reasoning* or *formal argumentation* [35]. Finally, depending on the explanatory artefact, a compatible explanation type needs to be matched, such as contrastive statements, exemplars, feature importance or feature attribution, among many others.

### 1.2.3 Contrastive Explanations

Contrastive explanations dominate the human explanatory process and are considered a gold standard in XAI [106]. They juxtapose a hypothetical situation (foil) next to the factual account, highlighting their differences and the consequences or “would be” change in the outcome. They are appropriate for a lay audience and domain experts, can use concepts of varying difficulty and be expressed in different media such as natural language and images. Contrastive explanations are parsimonious as the foil tends to be based on a single factor, but, if desired, they can account for an arbitrary degree of feature covariance. They support interaction, customisation and personalisation, e.g., a foil built around a user-selected feature, which can be used to restrict their search space, possibly making them easier to retrieve. When deployed in a user-centred application, they can provide the explainees with appealing insights by only using actionable features in the foil. However, their effectiveness may be problematic when explaining a black box that is proprietary (e.g., protecting a trade secret) since contrastive explanations can leak sensitive information, thereby allowing the explainee to steal or game the underlying model. In an open world, they also suffer from vaguely defined or imprecise notions known as *non-concepts* [112], e.g., “What is not-a-dog?”

In XAI, a black box can be explained with various types of contrastive statements. *Class-contrastive* explanations, often called *counterfactuals*, are the most popular type. They provide a slight variation of feature values held by a factual data point under which its prediction changes. The contrast with respect to the predicted class can either be implicit, i.e., “Why class  $X$  (and not any other class)?”, or explicit, i.e., “Why class  $X$  and not  $Y$ ?” For binary classifiers these two cases are equivalent, however multi-class models would respectively produce a one-vs-rest and one-vs-other explanations. *Instance-contrastive* explanations, on the other hand, are used when the explaineer seeks to understand why two data points, which from the user’s perspective are similar, do not share the same prediction, i.e., the explaineer wants to learn what change would result in an identical classification outcome. Contrastive statements can also be characterised by their lineage: *model-driven* explanations are represented by an artificial data point (akin to a centroid), whereas *data-driven* explanations are instances recovered from a (training) data set (similar to a medoid).

All of the aforementioned properties make contrastive statements appealing, but some of them may be lost in practice, e.g., an imperfect implementation, resulting in subpar explanations. Notably, contrastive explanations resemble causal insights, but unless they are generated with a full causal model [119, Chapter 4], they should not be treated as such and instead be interpreted as insights about the black box’s decision boundary. If they are model-driven, as opposed to data-driven, they may not necessarily come from the data manifold, yielding explanations that are neither feasible nor actionable in the real life, e.g., “Had you been 200 years old, ...” Even if they are coherent with the data distribution, the foil may still come from a sparse region, thus prescribing possible but improbable feature values [123]. Contrastive explanations are often specific to a single data point, although humans are known to generalise such insights to unseen and possibly unrelated cases (“The Illusion of Explanatory Depth” effect [132]), which may result in overconfidence. Observing such discrepancies should encourage creators of contrastive explainers to report them for the benefit of the explainees and the engineers deploying these algorithms.

### 1.3 Benefits of Explainability

The predominant role of explainability is to engender understanding of selected aspects of an intelligent black-box system. Nonetheless, it can also become a tool to assess fairness and inspect accountability of such systems, which, along explainability, have recently become important research topics [147]. While some researchers claim that we should not expect machine learning algorithms, such as deep neural networks, to be explainable and instead regulate them purely based on their real-life performance [144], it is not a widely shared belief [65]. This insight comes from the alleged inability of humans to explain their actions since such justifications are post-factum stories that are concocted and retrofitted for the benefit of the audience. Certifying

autonomous agents based on their output, on the other hand, is consistent with human values as one can hypothesise about committing a crime, but one cannot be punished unless such a thought is acted upon. While the origin of human thought process may be shrouded in mystery, its formulation is expected to follow the reason of logic to be (socially) acceptable. In particular, Miller [105] refutes performance-based validation by arguing that explainability stemming from regulatory requirements is secondary to concerns arising from societal values such as ethics and trust.

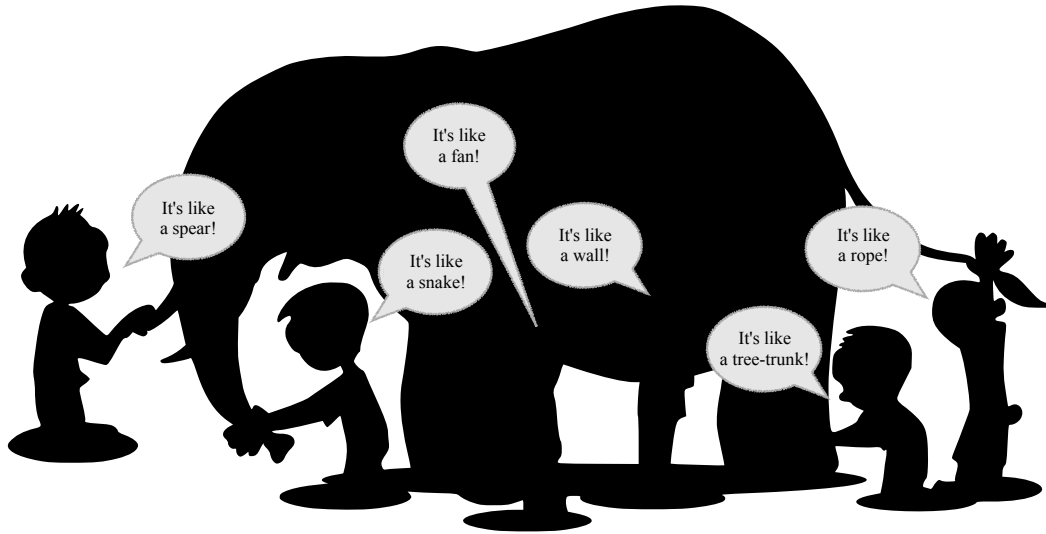
**Interpretability** The direct and intended benefit of explainability is the ability to interpret and understand a selected aspect of an intelligent system. If the system is a glass box, as opposed to a black box, it has been developed with (ante-hoc) explainability in mind, making it transparent by design and easy to examine. Black box systems that require post-hoc explainability, on the other hand, need to be retrofitted with an explainer that can provide the insights of interest. Choosing an appropriate explainer, however, can be challenging given the diversity of explanation types, their different communication media and varied difficulty of conveyed concepts, all of which need to be suitable for the intended audience and application as discussed earlier. As each method may just give a small, and quite possibly distorted, reflection of the true behaviour of a black box, achieving the desired level of transparency can require multiple, complementary explainers working together. This multiplicity of explanatory insights can be compared to unique probing and inspection techniques that without a shared context may yield competing or even contradictory evidence akin to the parable of “The Blind Men and the Elephant” [135], which is visualised in Figure 1.3.<sup>1</sup> While daunting, this versatility has its benefits: it allows us to tackle certain fairness and accountability issues of predictive systems.

**Fairness** Algorithmic decisions are mostly a reflection of the patterns elicited from training data, augmented by the inductive bias inherent to the chosen modelling technique. With the increase in predictive power of our models, more of such insights can be unearthed, some of which biased towards certain individuals or groups. Unfairness stemming from (historical) data [25, 114], however, may not necessarily be the only source of bias as predictive algorithms [136, 137], in particular their training procedure, can introduce or intensify these phenomena because of the underlying technical assumptions. Bias often manifests itself in unfair predictions, where a certain individual or a group is treated differently than comparable data points varying only in sensitive attributes such as gender or ethnicity. These notions are usually labelled as individual and group fairness or disparate treatment and disparate impact, corresponding to predictions or (ground-truth) labels of individual data points and their subgroups respectively.

In this setting, explainability can become an investigative toolkit for identifying various types of bias and its source. For example, feature importance and influence can reveal which sensitive

---

<sup>1</sup>[https://en.wikipedia.org/wiki/Blind\\_men\\_and\\_an\\_elephant](https://en.wikipedia.org/wiki/Blind_men_and_an_elephant)



**Figure 1.3:** Depiction of “The Blind Men and the Elephant” parable. It symbolises that individual pieces of evidence may be contradictory and can often be insufficient to understand the bigger picture without first being aggregated and grounded in a shared context.

attributes contribute to an (unfair) decision, whereas exemplars and influential training examples can be used to identify training data responsible for bias. Counterfactuals are particularly versatile since in addition to their very appealing explanatory properties (see Section 1.2.3), they are well suited to inspect individual counterfactual fairness [83]. A simple example of this practice is a contrastive statement conditioned on a protected attribute, e.g., “your loan application would have been accepted had you been a male”. This role of explainability is especially convenient for the end user when the explainer is provided as an interactive system that can be queried to receive customised and personalised counterfactual explanations, which, if feasible, are conditioned on selected (sensitive) attributes or their subsets.

**Accountability** Explainability can also help in discerning accountability of black-box models, including their robustness, safety and security. This is particularly important when such systems are used for high-stakes decisions, for example, parole hearings [9] and autonomous vehicle steering [52]. Ensuring that they are not susceptible to technological attacks or hacking can prevent life-threatening situations such as an autonomous vehicle performing a dangerous manoeuvre when a collection of small stickers is glued on the tarmac at a crossroads [3]. This is a real-life example of a phenomenon called adversarial attack [49], where human-imperceptible changes are introduced to a data point – an image captured by the car’s camera system in this case – causing it to be misclassified by the underlying predictive model. Notably, their similarity to counterfactuals [173] suggests that this type of explanations can help to identify and eradicate such undesirable model behaviour; the main difference between the two is the human ability to observe the modification in explanations but not in adversarial instances. In addition to

adversarial robustness, other aspects of intelligent systems accountability and their relation to explainability are worth investigating, e.g., susceptibility to recovering or stealing (parts of) predictive models and their training data, and validating monotonicity constraints.

## 1.4 Explainability Research

Thus far we have been mainly concerned with AI and ML explainability on a relatively abstract level, all of which constitute just a small portion of XAI and IML research. We touched upon the societal expectations of predictive black boxes and discussed popular terminology used in the field, including the differentiation between notions such as explainability and transparency. We introduced the illusory trade-off between explainability and predictive performance, which links to choosing between a post-hoc and an ante-hoc explainer. We also discussed the scope of explainability for the entire machine learning process: data, models and predictions, showing the objective at each stage and highlighting their interconnections. Notably, we looked at XAI and IML from the explainee’s perspective, outlining a range of crucial factors for adapting or designing human-friendly explanations. We rounded the discussion off with an overview of far-reaching benefits of explainability, which include improved interpretability of black-box systems as well as means of assessing their fairness and evaluating their accountability. This diverse theoretical overview of XAI and IML concepts sets the scene for a high-level survey of practical explainability approaches that follows. In particular, we examine explainers that are specific to a particular model, and universal algorithms applicable to any predictive black box.

**Explainability in Practice** With the theoretical and social aspects of explainers out of the way, we dive into practical explainability research. In an ideal world, XAI and IML publications would consider the aforementioned factors and build their mechanics around them, however it has only recently become a trend and numerous early pieces of work lack such a reflection. The technicalities of explainers, while often the main focus, can also be unintentionally misleading or misreported. For example, implementations that accompany papers may be inconsistent with the theoretical findings, pseudo-code or a restrictive toy example presented therein due to implicit assumptions embedded into utilised optimisation strategies or employed approximations. Therefore, treating an explainer at face value or judging it purely based on the theory outlined in the corresponding publication may have unintended consequences and an in-detail inspection should be considered before committing to it. The most prominent example of such a phenomenon are *model-agnostic* explainers, which by design can work with any type of a predictive black box, but are not equally well suited for each – a common misconception that we discuss in the following section.

A critical evaluation of explainers requires a collection of well-defined properties, which can also be used to categorise and describe explainability algorithms. We collect a comprehensive list of these characteristics in our XAI taxonomy presented in Chapter 2; for the benefit of the literature

overview that follows, we outline a relevant subset here. The biggest differentiating factor is black box compatibility – explainers can either be *model-agnostic*, i.e., work with any model; or *model-specific*, designed separately for each individual model type. Explainability approaches can further be categorised based on their relation to the black box: *ante-hoc* explanations are sourced from the black box itself (thus making it a glass box), and *post-hoc* explanations rely upon an additional model (i.e., a transparency layer) built on top of the black box. Furthermore, explanations can be *local*, i.e., pertaining to a data point or prediction; *cohort-specific* – applicable to a data or model subspace; or *global* to summarise a data set or simplify the behaviour of an entire black box. For some data types, explanations produced in the *original data domain* or the *internal representation* used by a black box may be unintelligible, in which case an *interpretable representation* is used to communicate the explanations. Finally, there are different *explanation types*: feature importance, influence of features on predictions, influence of training data on predictions, data exemplars, contrastive statements and supportive statements, among many others.

**Generic Explainers** The most popular explainers are *model-agnostic* and *post-hoc* since they can be retrofitted into any predictive black box. These include RuleFit [45], Local Interpretable Model-agnostic Explanations (LIME [129]), anchors [130], SHapley Additive exPlanations (SHAP [100]), Black-box Explanations through Transparent Approximations (BETA [85, 86]), PD [44], ICE [48] and Permutation Importance (PI [22]), among many others. Most of these methods operate directly on raw data, with the exception of LIME and anchors, which use interpretable representations to improve intelligibility of explanations composed for complex data domains such as text and images. While RuleFit is limited to tabular data, it represents instances with conjunctions of logical conditions, which are used to communicate the explanations to the user. RuleFit, LIME and anchors are predominantly local explainers, whereas permutation importance is global. SHAP is designed to provide local explanations, which can be generalised to cohort or global explanations; similarly, individual conditional expectation is a local explainer, which generalises to cohort or global explanations in the form of partial dependence. BETA is unique in this aspect as it produces a *scoped* global approximation of a black box, which naturally provides cohort and local explanations without additional processing. Both anchors and BETA use a form of contrastive and supportive statements; PI is based on feature importance; and RuleFit, LIME, SHAP, ICE and PD are expressed as (interpretable) feature influence.

Given that all of these methods are post-hoc, they create an additional modelling layer on top of a black box to generate the explanations. This independence from the underlying predictor comes at the expense of increased overall complexity, which can be detrimental to the explanation fidelity, as well as implicitly limit explanatory scope. RuleFit first generates logical rules that serve as binary characteristics of the explained data; then it quantifies the importance of these logical concepts by modelling them with a linear classifier. Similarly, LIME constructs an interpretable representation of the explained instance and fits a transparent

surrogate model – a sparse linear regressor – in its neighbourhood to approximate a nearby black-box decision boundary, thereby computing influence of the interpretable concepts. Anchors take a complementary approach and generate conditions on this instance-specific interpretable representation that are “sufficient” for a particular prediction to hold. BETA operates somewhat in-between, building a simple nested rule set that on the outer level globally mimics a black box; it optimises for explanation completeness, while the local “generalisation” of anchors and “specification” of LIME strive for soundness, which is a common trade-off for post-hoc techniques. These three approaches are examples of surrogate explainers, which explicitly create a simple and transparent predictive model for a selected part of a black box to mimic its operation in a human-intelligible fashion.

Individual conditional expectation (ICE) and partial dependence (PD) are both based on the same premise, except the former is intended for an individual instance and the latter for a selected data subset. They both record the response change of a black box with respect to user-specified features when varying them within a predefined range while keeping all the other attributes unchanged – an example of these two explainers has already been shown in Figure 1.2. Permutation importance works similarly, however its objective is to measure the importance of a selected feature for a black-box predictor. It shuffles values of this feature within a subset of data to quantify how sensitive the underlying model is with respect to it; this effect is measured by the magnitude of change in the predictions as compared to the original data. SHAP also measures feature influence, however it does so for *all* of the features for an individual prediction. It uses a concept from game theory called Shapley Values [140] to compute how much each feature contributes, positively or negatively, to a prediction.

**Deep Learning Explainers** Another attractive avenue of explainability research, which partly overlaps with post-hoc methods, is opening up (deep) neural networks by designing tools and techniques *specific to these models* or, more broadly, compatible with differentiable predictors. These models tend to be notorious black boxes, however their superior predictive performance for a wide spectrum of applications accelerates their popularity and widespread adoption [91]. Early work [28] dating back to 1996 concerns approximating the global behaviour of an entire neural network with a decision tree – a post-hoc, global model explanation. While still being an inspiration for explainability research, modern neural networks have grown tremendously in size, processing large amounts of diverse data types. These technical improvements and application diversification shifted the demand towards more compact and informative explainability approaches.

Saliency maps are an example of such an auditing technique, which is often used with image data to highlight pixels that are important for a particular prediction of a black-box neural network – a local feature importance metric [184]. A different approach is to identify training data that “bias” a model to predict an instance in a particular way, which can be achieved with influence functions [74], i.e., local exemplar-based explanations. Methods for generating

class-contrastive counterfactual explanations are another appealing technique used to inspect predictions of deep neural networks [173]; they can be computed by optimising an objective similar to the one used in adversarial learning [49]. Testing with Concept Activation Vectors (TCAV [71]) is a more recent approach that creates a collection of high-level, human-intelligible concepts used to reason about the behaviour of a neural network. It uses concept exemplars and class-contrastive statements – the influence of presence and absence of these concepts on a particular class – to engender global understanding of the underlying model.

**Specialised Explainers** An alternative XAI and IML research agenda concentrates on inherently explainable predictive models and ante-hoc explainers designed for popular black boxes. An example of the former are generalised additive models, which usually have good predictive power and are transparent – users can inspect their operation, although a technical understanding of their inner workings may be required [99]. A more explainee-friendly predictive model is a falling rule list – an ordered list of if-then rules helping the user to quickly grasp the reason behind a particular prediction [177]. A similar approach processes internals of a black box to compose its faithful explanation, i.e., ante-hoc explainability. A naïve Bayes classifier can be explained by presenting its user with a visualisation of the model’s weights for a particular class and feature contribution for an individual prediction [81], i.e., global and local explanations based on feature importance and contribution respectively. Another example is explaining clustering outcomes by showing exemplars and highlighting dominating features for each cluster [68].

This high-level literature review is far from exhaustive, instead focusing on landmark research contributions that have influenced our work. In particular, we omitted techniques that can be considered a part of the explainability research but do not address this issue directly. These include interactive exploratory user interfaces [63, 179], creative visualisations of explainability approaches [77] and systems combining multiple explainability techniques within a single tool [178]. Due to the diversity of concepts, spanning different disciplines and touching upon independent ideas, rather than compiling them into a single literature review, we introduce publications specific to each topic in relevant chapters of this thesis. Meanwhile, this overview of practical explainability approaches complements the theoretical account of this topic presented in the preceding sections and allows us to formulate our research agenda, which adheres to our findings, addresses identified shortcomings and improves upon current solutions.

## 1.5 Designing Intelligible and Robust Explainers

Theoretical expectations and desiderata do not always align with operationalisation and practicalities of XAI and IML algorithms, and the latter are what ends up affecting our lives. For example, explainability is an inherently social process that usually involves bi-directional communication, but most implementations – even the ones using contrastive statements [169, 173]

– output a single explanation that is optimised according to some predefined metric, thus not necessarily addressing concerns of every individual explainee [151]. Similarly, while inherently transparent predictive models and ante-hoc explainers may be preferred [133], such solutions are model-dependent, usually labour-intensive and tend to be application-specific, therefore limiting their scope and wider applicability. Instead, post-hoc and model-agnostic explainers dominate the field [100, 129, 130] since they are considered one-stop solutions – a unified explainability experience without a cross-domain adaptation overhead. This silver bullet attitude, however, comes at a cost: subpar fidelity that can result in misleading or incorrect explanations. While increasingly all such considerations find their way into publications, they are often limited to acknowledging the method’s shortcomings without offering a viable solution.

### 1.5.1 Research Motivation

Observing these discrepancies has prompted us to investigate XAI and IML approaches that respect explainees’ expectations in addition to being both post-hoc and model-agnostic. The latter choice is motivated by the widespread impact that such techniques could have if designed with high fidelity in mind. However, explainability is no different to many other concepts in computer science, in so far as there is no proverbial free lunch – a single, universal algorithm cannot outperform all the others across the board. In machine learning and data mining this often comes down to a series of investigative steps to guide algorithmic choices down the line, which can be operationalised within a standardised process for knowledge discovery such as KDD [38], CRISP-DM [26, 103] or BigData [6]. For example, by analysing feature correlation, data uniformity and class imbalance, we can account for these phenomena when engineering features and training models, thereby making the resulting AI systems more accountable and robust. Nonetheless, XAI and IML lack such a process or even a set of universal properties that could guide the development and assessment of explainers – their requirements and needs – which likely hinders adherence to best practice.

While developing a predictive pipeline, we have an abundance of pre-processing and modelling tools and techniques at our disposal, a selection of which will end up in the final system. The XAI and IML landscape, on the other hand, is quite different: explainers tend to be end-to-end tools with only a handful of parameters exposed to the user. In view of the “no free lunch” theorem, this is undesirable as despite being model-agnostic, i.e., compatible with any model type, these monolithic algorithms do not perform equally well for every one of them [158]. This variability in their behaviour can often be attributed to a misalignment between the assumptions baked into an explainer and the properties of the explained system, which manifests itself in low fidelity. Model-specific or ante-hoc explainers can be used to address this issue; however, as discussed earlier, such a solution may have limited applicability and cannot be retrofitted to preëxisting AI systems. This impasse points towards a need for *flexible model-agnostic and post-hoc explainers* that could be easily adapted for each individual predictive black box.

Another essential aspect of AI explainers is the human factor. The insights provided to the user should strive for *explainability* and not just transparency, which entails understanding the requirements and expectations of the intended audience and use case. This is particularly important when an explanation is provided to the users as a one-off “take it or leave it” statement – an approach that currently dominates the field [151] – in which case it needs to account for a wide range of social and technical aspects. However, when the audience is diverse, one predefined type of an explanation may be insufficient as it is unlikely to address all the possible questions and unique perspectives. In such cases, the solution comes from a *bi-directional communication* that gives the explainees an opportunity to interactively customise and *personalise* the explanation [101]; and if they disagree, it allows them to contest and rebut it. Finally, the explanation type and delivery medium should also be adjusted according to the circumstances, with the current literature [106, 173] suggesting *contrastive* explanations as the gold standard.

### 1.5.2 Research Aims and Objectives

Putting all of our preliminary findings together allows us to identify gaps and compose a self-contained research agenda to advance the field with human-centred, model-agnostic and post-hoc explainers that exhibit high fidelity. To this end, we first collect and organise technical and social properties of AI and ML explainers to create a reference list that can be used to systematically evaluate and compare preëxisting approaches, and express design desiderata and guidelines. Second, we develop a highly-customisable, model-agnostic and post-hoc explainer that can be easily adapted to the problem at hand, thus guaranteeing high fidelity of the resulting explanations. Third, we identify a composition and configuration of such an explainer enabling the explainees to interact with it to receive personalised experience that addresses their concerns and answers their questions. We ensure that regardless of the interaction mode, the explanations feel familiar, e.g., by using contrastive statements, and engender trust and understanding.

**Framework for Reasoning About Explainers** We collect and organise a *set of technical and social aspects* of AI and ML explainers, some of which are scattered throughout this chapter and summarised in the preceding section. Such a reference list is invaluable when designing and developing novel explainers; in particular, we use it to guide the creation of our flexible, model-agnostic and post-hoc explainer. Additionally, it serves as a principled evaluation tool for preëxisting explainers, thus helping to systematically compare and contrast their properties when choosing an approach suitable for the problem at hand. We make the framework compatible with abstract explainers and algorithmic implementations to allow investigating discrepancies between their theoretical and practical aspects. All of these constitute unified and multi-purpose explainability desiderata that clearly communicate capabilities of an arbitrary explainer, which is an improvement over current practices where reporting is non-existent or selective at best.

**Post-hoc and Model-agnostic Explainers** Our second objective is to design a flexible, highly-customisable, model-agnostic and post-hoc explainer that is faithful to the underlying black box. Instead of an end-to-end tool, we develop an explainability *meta-algorithm* to empower engineers to build tailor-made explainers that reflect their needs and respect the restrictions imposed by the explained system. We further ensure that our method works with any data type: tabular, image and text, and that the complexity of its explanations can be adapted to the expectations of the intended audience. To this end, we look into creating proxies for unintelligible data, e.g., raw pixels, to appropriately adjust the difficulty of their explanations. While we envisage many configurations of our meta-algorithm, navigating this landscape without an accompanying “user guide” may be challenging. Therefore, we study its selected configurations to understand their influence on the capabilities and limitations of the resulting explainer, thus recognising the consequences of certain choices.

**Human-centred Explainers** Having taken care of the explainer technicalities, we focus on its social setting and aligning its operation with the explaineer’s expectations. To this end, we consider using an *appealing and versatile explanation type*, such as contrastive statements, to serve a diverse range of audiences and applications. We also investigate *interaction protocols* between the explainer and the explaineer to facilitate customisation and personalisation of the explanatory process, thereby delivering explanations that answer each individual explaineer’s unique questions. Additionally, we evaluate our meta-algorithm against other social aspects of AI and ML explainability that have surfaced when composing our framework for reasoning about explainers, and adapt it appropriately.

### 1.5.3 Research Contributions

Based on our aims and objectives, we devised and executed a research agenda, the findings of which are summarised below. These include a *taxonomy* of XAI and a *modular surrogate* explainer meta-algorithm called **bLIMEy** (build **LIME** yourself). We further compose a *tree-based surrogate* explainer, named **LIMEtree**, for which we show how to achieve high fidelity and meaningful explanations. To this end, we analyse explanatory capabilities of decision trees and propose an algorithm for generating *counterfactual statements* based on the tree structure, which we call **CtreeX** (Contrastive **t**ree **e**xplainer). We then show how to deploy this algorithm – in a device called Glass-Box – to allow the user to *interactively control* the content of contrastive explanations via a voice-driven dialogue. A list of relevant publications resulting from our research and supporting each contribution is included at the end of each of the following paragraphs, which pertain to these individual contributions. A more detailed outline of the thesis structure and content is presented in Section 1.6.

### Explainable AI Taxonomy

We collate and organise a comprehensive list of social and technical properties of XAI and IML systems into a taxonomy, which we present in Chapter 2. We categorise them into five dimensions: *functional*, *operational*, *usability*, *safety* and *validation*, which are specific to certain audiences and applications, thus enabling easy navigation. This grouping is both role-driven – appealing to researchers, engineers and auditors – and application-driven – suitable for creation, reporting, evaluation and comparison of novel and preëxisting explainers. In Appendix A, we also show how to operationalise the taxonomy as *Explainability Fact Sheets*, which in this case are composed for LIME [129], CtreeX (Chapter 4) and LIMETree (Chapter 5). The following publications relate to the explainable AI taxonomy:

- [157] Kacper Sokol and Peter A Flach. Counterfactual explanations of machine learning predictions: Opportunities and challenges for AI safety. In *Proceedings of the SafeAI Workshop at the AAAI Conference on Artificial Intelligence*, 2019.
- [147] Kacper Sokol. Fairness, accountability and transparency in artificial intelligence: A case study of logical predictive models. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 541–542, 2019.
- [148] Kacper Sokol and Peter Flach. Desiderata for interpretability: Explaining decision tree predictions with counterfactuals. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 10035–10036, 2019.
- [149] Kacper Sokol and Peter Flach. Explainability fact sheets: A framework for systematic assessment of explainable approaches. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 56–67, 2020.

The publications authored by Kacper Sokol and Peter Flach are Kacper’s original ideas investigated under Peter’s supervision. This arrangement holds for all the publications, with the exceptions clearly marked by a footnote explaining individual contributions of relevant authors.

### Modular Surrogates

We design bLIMEy: a meta-algorithm for building modular surrogate explainers, which are both model-agnostic and post-hoc, allowing them to be configured and adapted for the problem at hand. This framework consists of three components: data sampling, interpretable representation transformation and explanation generation, each one analysed and discussed in Chapter 3. In particular, we show that the first two are mostly responsible for the explainer’s fidelity, and the latter two influence the complexity and appeal of the resulting explanations. Additionally, we introduce a selection of algorithms suitable for each building block of our meta-algorithm, investigate their pros and cons, and implement them in FAT Forensics – an open source Python package presented in Appendix B. An important finding of this in-depth analysis illustrates that, in certain circumstances, using a linear model as the explanation generation mechanism – a popular choice in the literature [45, 129] – may severely limit the explanation expressiveness as we demonstrate in Appendix C.3.

These discoveries lead us towards employing classification and regression trees to generate explanations as part of the surrogate meta-algorithm. While in certain cases decision trees are

transparent, in Chapter 4 we show that they are not inherently explainable and propose CtreeX as a remedy – an algorithm to generate contrastive and supportive statements for tree predictions. These two explanation types have wide-reaching social and technical benefits and are appealing to humans. In Chapter 5, we combine our findings and propose LIMETree: a surrogate explainer of probabilistic black boxes that is based on multi-output regression trees. We show how using this particular tree type for modelling the underlying predictor accounts for class dependencies and that our configuration of the surrogate achieves high fidelity. Our work on decision trees, surrogate explainers and their implementations is published in the following papers:

- [158]<sup>2</sup> Kacper Sokol, Alexander Hepburn, Raul Santos-Rodriguez, and Peter Flach. bLIMEy: Surrogate prediction explanations beyond LIME. *2019 Workshop on Human-Centric Machine Learning (HCML 2019) at the 33<sup>rd</sup> Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada, 2019*. URL <https://arxiv.org/abs/1910.13016>. arXiv preprint arXiv:1910.13016.
- [152] Kacper Sokol and Peter Flach. Towards faithful and meaningful interpretable representations. *arXiv preprint arXiv:2008.07007*, 2020. URL <http://arxiv.org/abs/2008.07007>.
- [150] Kacper Sokol and Peter Flach. LIMETree: Interactively customisable explanations based on local surrogate multi-output regression trees. *arXiv preprint arXiv:2005.01427*, 2020. URL <https://arxiv.org/abs/2005.01427>.
- [159]<sup>3</sup> Kacper Sokol, Raul Santos-Rodriguez, and Peter Flach. FAT Forensics: A Python toolbox for algorithmic fairness, accountability and transparency. *arXiv preprint arXiv:1909.05167*, 2019. URL <https://arxiv.org/abs/1909.05167>.
- [160]<sup>4</sup> Kacper Sokol, Alexander Hepburn, Rafael Poyiadzi, Matthew Clifford, Raul Santos-Rodriguez, and Peter Flach. FAT Forensics: A Python toolbox for implementing and deploying fairness, accountability and transparency algorithms in predictive systems. *Journal of Open Source Software*, 5(49):1904, 2020.
- [161]<sup>5</sup> Kacper Sokol, Alexander Hepburn, Raul Santos-Rodriguez, and Peter Flach. What and how of machine learning transparency: Building bespoke explainability tools with interoperable algorithmic components. *Hands-on Tutorial at The European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD), Ghent, Belgium, 2020*. URL [https://events.fat-forensics.org/2020\\_ecml-pkdd](https://events.fat-forensics.org/2020_ecml-pkdd).

## Interactive Explainers

We investigate how each component of our surrogate meta-algorithm can become interactive to support explanation customisation and personalisation for the benefit of the explainees. In Chapter 6 we demonstrate how data sampling can be configured to adjust the explanation scope to focus it on a single instance or a predefined neighbourhood, thus controlling the explanation generalisation. We further show how the interpretable representation can be shaped to encode

---

<sup>2</sup>bLIMEy is Kacper’s original idea researched under Raul’s and Peter’s supervision with help from Alexander, who contributed experiment design and code.

<sup>3</sup>Kacper was the lead designer and developer of the FAT Forensics package, which was developed under Peter’s and Raul’s supervision.

<sup>4</sup>Kacper was the lead designer and developer of the FAT Forensics package, which was developed under Peter’s and Raul’s supervision. Alexander, Rafael and Matthew contributed code.

<sup>5</sup>Kacper has planned and organised the tutorial in collaboration with Alexander, Raul and Peter, who contributed talks and hands-on resources.

personalised, human-intelligible concepts used to communicate the explanations – a significant improvement over machine-generated proxies. Finally, we illustrate how to customise the content of explanations by interacting with the explanation generation step, which appears to be the most beneficial. For example, when using contrastive explanations, the user can request certain attributes and prevent others from appearing in the conditional part of the statement; and if the black box models more than two classes, the explainee can also explicitly specify the contrast.

We take advantage of these findings (especially the last one) to design and deploy Glass-Box: an interactive, voice-driven system that explains its predictions to the user in a dialogue built atop class-contrastive, counterfactual statements. This interface allows the explainees to ask very specific, contrastive questions via which they can personalise the explanations, e.g., by requesting a particular subset of features to be included and/or excluded from the counterfactual condition (if at all possible). Glass-Box uses a decision tree as the underlying predictive model and generates contrastive explanations with our bespoke tree-specific algorithm. Nevertheless, interacting with and personalising contrastive explanations can be generalised to an arbitrary black box when using a tree-based surrogate explainer, such as LIMETree, which can also take advantage of interactively customising the data sampling and interpretable representation generation steps specific to surrogates. A discussion of various modes of interactive customisation and personalisation of XAI and IML explanations via a dialogue or otherwise is published in the following papers:

- [154] Kacper Sokol and Peter A Flach. The role of textualisation and argumentation in understanding the machine learning process: A position paper. In *Automated Reasoning Workshop*, pages 11–12, 2017.
- [153] Kacper Sokol and Peter A Flach. The role of textualisation and argumentation in understanding the machine learning process. In *IJCAI*, pages 5211–5212, 2017.
- [155] Kacper Sokol and Peter A Flach. Conversational explanations of machine learning predictions through class-contrastive counterfactual statements. In *IJCAI*, pages 5785–5786, 2018.
- [156] Kacper Sokol and Peter A Flach. Glass-Box: Explaining AI decisions with counterfactual statements through conversation with a voice-enabled virtual assistant. In *IJCAI*, pages 5868–5870, 2018.
- [151] Kacper Sokol and Peter Flach. One explanation does not fit all. *KI-Künstliche Intelligenz*, pages 1–16, 2020.

## 1.6 Outline of the Thesis

Our contributions are distributed across five chapters as follows.

**Chapter 2** outlines the explainable AI taxonomy, with

**Appendix A** showing examples of its operationalisation as *Fact Sheets* for (A.1) LIME, (A.2) Ctrees and (A.3) LIMETree;

**Chapter 3** introduces the bLIMEy meta-algorithm for building modular surrogates and discusses properties of each surrogate building block, with

**Appendix B** describing bLIMEy implementation within FAT Forensics and

**Appendix C** showing theoretical limitations of explaining tabular data with surrogates based on linear models (C.3), analysing benefits of tree-based interpretable representations for tabular data (C.2), and investigating the influence of occlusion colour and segmentation granularity on interpretable representations of images (C.1);

**Chapter 4** highlights the difference between transparency and explainability of decision trees and proposes CtreesX – an algorithm for generating contrastive and supportive explanations of their predictions;

**Chapter 5** presents LIMETree – a high-fidelity surrogate based on multi-output regression trees for explaining predictions of black-box probabilistic models; and

**Chapter 6** discusses interactive customisation and personalisation of various explanation aspects, and showcases Glass-Box – a voice-driven conversational user interface that allows explainees to influence and steer the explanation content.

We summarise these findings, discuss their implications and outline future research directions in Chapter 7. We also overview the content of chapters and review their respective contributions below to help the reader navigate this thesis.

**XAI Taxonomy** Our first contribution (Chapter 2) is an explainable AI taxonomy that can be used as a principled framework for reasoning about explainers. To this end, we collect a wide range of social and technical properties expected of explainability approaches and organise them into five distinct dimensions:

**functional** determines technical suitability of an explainer based on supported feature types, compatible black boxes and the explanation target, among many others;

**operational** describes how explainees interact with the explainer and what is expected of them, e.g., interaction mode, expected audience and the explanation medium;

**usability** spans explanation characteristics that are important from an explainee’s viewpoint, e.g., fidelity, coherence with the recipient’s mental model and actionability;

**safety** tackles the effect of explainability on robustness, security and privacy aspects of the underlying predictive system such as information leakage and explanation misuse; and

**validation** outlines a range of explainer evaluation techniques spanning synthetic validation and user studies.

This diverse taxonomy enables us to systematically evaluate available explainers, identify their shortcomings and recognise properties that are universally desired of them. We use these desiderata to plan research and guide development of an explaineer-centred, fidelity-oriented, post-hoc, model-agnostic and highly-customisable explainer. Such analysis can be formalised in *Explainability Fact Sheets*, examples of which characterising the LIME, CtreeX and LIMETree algorithms are shown in Appendix A – they discuss theoretical properties, implementation details and the discrepancies between the two for the aforementioned explainers.

**bLIMEy** Surrogate explainers (Chapter 3) best fit our desiderata, in so far as they are model-agnostic and post-hoc. Surrogates are simple and explainable models trained to mimic the underlying black box in the neighbourhood of a particular instance (local), for a specific data subspace (cohort) or spanning the entire instance space (global). Their apparent complexity is beneficial to our end since we can decompose them into functional building blocks, which allows us to reconfigure or replace each module to satisfy the diverse desiderata and further tune it to the needs of each individual use case. This flexible meta-algorithm, which we call bLIMEy, consists of three self-contained components:

**data sampling** generates (synthetic) data to cover the region of the decision space selected to be explained – these data are predicted by the black box to record its behaviour;

**interpretable representation** encodes the data in a format that is appropriate for the intended audience, e.g., numerical features of tabular data can be grouped into meaningful categories such as low, medium and high; and

**explanation generation** fits a simple and explainable model to the interpretable representation of the sampled data using their black-box predictions as the target.

This functional separation inspired us to implement these modular surrogate explainers in Python, which we publish in an open source package – called FAT Forensics and described in Appendix B – distributed under the BSD 3-Clause licence.

In order to satisfy the other two desiderata – high fidelity and explaineer’s satisfaction – we investigate properties and trade-offs of each individual surrogate building block. In particular, we show that the explainees can benefit from the interpretable representation complexity being tuned to their individual background knowledge and mental model; and that selecting a surrogate model capable of delivering familiar explanation types and, possibly, supporting user interactions is desirable as well. We also discover that class-aware sampling and determinism of the interpretable representation transformation are crucial for high fidelity of the resulting explainer. Moreover, the interpretable representation generation mechanism has to be functionally compatible with the employed surrogate model type – the mismatch may be detrimental to fidelity of the explainer. We investigate this pairing issue for tabular data to find that a discretisation-based

interpretable representation coupled with a linear model – a combination used by a popular surrogate explainer called LIME [129] – is sub-optimal and results in information loss. (We present analytical evidence of this phenomenon for Ordinary Least Squares regression in Appendix C.3.) Our findings further suggest that using *decision trees* as the surrogate model can substantially improve the quality of explanations, and their inherent transparency renders them a perfect candidate.

**CtreeX** While small decision trees are transparent, they are not necessarily explainable as we argued earlier in this chapter. This phenomenon is particularly striking when targeting a lay audience, who may find such insights uninformative or even unintelligible. The most common transparency techniques for decision trees include:

- tree structure visualisation;
- tree-based feature importance;
- root-to-leaf logical conditions;
- leaf-based exemplars; and
- model-based what-if statements.

We overcome this limitation by proposing CtreeX (Chapter 4), which is an efficient algorithm for extracting two types of human-centred explanations:

**contrastive** retrieved by applying logical reasoning to compare and contrast different root-to-leaf tree paths; and

**supportive** achieved by generalising logical conditions imposed on individual root-to-leaf tree paths.

In addition to being advocated in the literature [106, 173] for their appeal and versatility, such explanations visibly improve explainability of decision trees by decoupling the size of the explanations from the model size. For example, regardless of the tree depth, a class-contrastive counterfactual statement is only as large as the number of conditions in its foil. Our algorithm is ante-hoc, making the explanations fully faithful with respect to the underlying tree – they are computed based on a binary representation of the internal structure of the explained tree. While the approach is model-specific, it can be generalised to an arbitrary black box by using a tree as the explanation generation module in surrogate explainers. Such an approach allows us to reap the benefits of contrastive and supportive explanations for any type of a black-box model as long as the surrogate exhibits satisfactory fidelity.

**LIMEtree** We explore the intricacies of building tree-based surrogate explainers (Chapter 5) by focusing on image classification with deep neural networks, which deliver state-of-the-art performance but are notorious black boxes. While decision trees are designated for tabular

data and often cannot compete with neural networks on sensory data, thereby foregoing their explanatory powers, they can be used as surrogates to explain images and text. Given that the underlying black box can be a regressor, a probabilistic classifier or a crisp classifier, we investigate the suitability of surrogate regression and classification trees. We find that *multi-output regression trees* are best suited for approximating probabilistic black boxes since they can simultaneously model multiple classes and their inter-dependencies, thus improving upon current solutions. We further discuss how to design the data sampling algorithm, compose the interpretable representation and train the surrogate tree to achieve near-full or full fidelity of the explainer, which allows us to overcome the biggest criticism of post-hoc explainability. We back our findings with theoretical guarantees, synthetic evaluation, real data experiments and user studies.

**Glass-Box** Explainers that output a human-friendly explanation type may still meet with a mixed reception and have a limited success when the audience is diverse as one explanation does not fit all. Such systems acknowledge the human recipients but miss out on appreciating their uniqueness – a challenge that we address with interactive customisation and personalisation of the *explanatory process* (Chapter 6). In particular, we investigate how each component of a surrogate – data sampling, interpretable representation creation and explanation generation – can be tuned by the explaineer to improve the perceived explainability of the underlying black box and boost user satisfaction and confidence. For example, the explaineer can influence the sampling breadth to alter the explanation scope from local to cohort to global. Similarly, an interpretable representation can be personalised to reflect beliefs of an explaineer, e.g., the user may adjust the thresholds for dividing a numerical feature into small, medium and high categories. However, the biggest gains come from customising the explanation content; counterfactuals, for example, can answer specific questions by being conditioned on user-requested features. As a proof of concept, we operationalise these findings in Glass-Box – an interactive, voice-driven, conversational, counterfactual explainer of black-box predictions that is designed upon the operational and usability dimensions of our XAI taxonomy. We then present this explainability system to a lay audience and domain experts to seek diverse feedback and insights that may be useful for building and deploying such algorithms in the real life.

We conclude this thesis (Chapter 7) by summarising our key findings and discussing future research directions. In the light of our contributions, we revisit our definition of explainability, look back on the importance of explanation fidelity and reconsider the role of humans in the explanatory process. Moreover, we show how all of our findings – taxonomy, surrogates, counterfactual explanations of decision trees, tree-based surrogates and interactive explanation personalisation – fit together in the broader explainability context. This discussion distills our contributions and paves a way towards more intelligible and robust surrogate explainers. It also highlights areas for future work, which, among others, include searching for appealing

interpretable representations and investigating data sampling strategies that reduce the number of generated instances while maintaining a certain level of surrogate model fidelity. The interactive, human-like explanatory process also opens up a possible research avenue where both human-machine and machine-machine can communicate and exchange explanations (e.g., using the formal argumentation framework) to reduce their number and refine their content, offering a principled solution to explanation personalisation and multiplicity.

## EXPLAINABLE AI TAXONOMY

Explanations in machine learning come in many forms, but a consensus regarding their desired properties is yet to emerge. To organise this landscape, we introduce a taxonomy and a set of descriptors that can be used to characterise and systematically assess explainable systems along five key dimensions: functional, operational, usability, safety and validation. In order to design a comprehensive and representative taxonomy and associated descriptors we survey the explainable artificial intelligence literature, extracting the criteria and desiderata that other authors have proposed or implicitly used in their research. Our review includes papers introducing new explainability algorithms to see what criteria are used to guide their development and how these algorithms are evaluated, as well as papers proposing such criteria from both computer science and social science perspectives. This novel framework allows to systematically compare and contrast explainability approaches, not just to better understand their capabilities but also to identify discrepancies between their theoretical qualities and properties of their implementations. We develop an operationalisation of the framework in the form of an explainability *Fact Sheet*, which enables researchers and practitioners alike to quickly grasp capabilities and limitations of a particular explainability method. We present example fact sheets developed for (A.1) LIME, (A.2) CtreeX and (A.3) LIMETree in Appendix A. When used as a *Work Sheet*, on the other hand, our taxonomy can guide the development of new explainability approaches by aiding in their critical evaluation along the five proposed dimensions.

### 2.1 Organising Explainability

With the current surge in explainable AI research, it has become a challenge to keep track of, analyse and compare many of these approaches. A lack of clearly defined properties that explainable systems should be evaluated against hinders progress of this fast moving research

field. This can result in undiscovered (or undisclosed) limitations and properties of XAI approaches and their implementations, as well as discrepancies between the two. Regardless of whether the intention is to propose a novel explainer, implement an already published approach or deploy an existing tool, identifying the characteristics of interest is an elaborate and time consuming process prone to overlooking some of the implicit factors. The diversity of these requirements further complicates the task – while the research field is predominantly concerned with *technical* properties of the explainers, their *social* aspects should not be neglected since explanations are often targeted at human recipients.

To address this issue, we propose that every explainability method designed for predictive systems should be evaluated against a taxonomy that assesses its (1) *functional* and (2) *operational* requirements, which determine its technical properties. Moreover, the quality of explanations should be evaluated against a list of (3) *usability* criteria to better understand their usefulness from a user’s perspective, i.e., their social aspects. Their (4) *security*, *privacy* and any *vulnerabilities* that they may introduce into the underlying predictive system should also be part of the taxonomy, thereby disclosing their technical implications. Lastly, their (5) *validation*, either via user studies or synthetic experiments, should be communicated. Therefore, a standardised list of explainability properties – or, to put it otherwise, desiderata – spanning all of these five dimensions would facilitate a common ground for evaluation and comparison of explainability approaches, and help their designers consider a range of clearly defined aspects important for this type of techniques.

Despite theoretical guarantees for selected explainability approaches, some of these properties can be lost in implementation due to a particular optimiser, algorithmic proxy, application domain or data set used. As it stands, many implementations do not exploit the full potential of the selected explainability technique, for example, a method based on counterfactuals may not take advantage [173] of their social and interactive aspects [106]. Similarly, model-agnostic approaches can render some desiderata difficult to achieve since these explainers cannot benefit from model-specific insights such as direct access to the internals of the underlying predictive algorithm. A use of guidelines or a systematic evaluation of an approach with a standardised taxonomy could help discover these unexpected functionality losses and account for or simply report them for the benefit of the research community and end users.

Interpretability, explainability and accountability of AI and ML systems have recently become important research issues triggered by the pervasiveness of automated decision making in our everyday life. Despite broad interest in these topics, there is no agreement on their definitions and desiderata, with a large part of the XAI and IML literature revolving around three main trends:

1. discussions of what is generally desired of explanations from a user’s perspective [34, 70, 80, 81, 96];

2. investigations of theoretical properties of selected explainability approaches, e.g., Miller [106] does that for counterfactuals; and
3. reports on implementations and experimental results of selected explainers, e.g., Wachter et al. [173] propose and study generating counterfactuals for differentiable models.

What appears to be lacking is a connection between universal properties expected of explainability systems and their presence in specific methods and their implementations; at best, some studies [80, 81, 178] propose a narrow list of desiderata and use it to evaluate the method proposed therein. For example, Kulesza et al. evaluate interactive visualisations of: music recommendations with respect to explanation fidelity [80], and naïve Bayes spam email classification with respect to fidelity, interactiveness, parsimony and actionability [81].

To help address the need of consensus regarding a common set of properties that every explainability method should be evaluated against, we collect, review and organise a comprehensive set of characteristics that span both computer and social science insights on that matter. Our goal is to provide the community with an *XAI taxonomy* that users of explainable systems – researchers and practitioners alike – can utilise to systematically discuss, evaluate and report properties of their techniques. This will not only benefit research, but will be of particular importance as a guideline for designing, deploying and evaluating explainability methods. Such a framework can especially benefit applications where adhering to best practices, legal regulations or certification is of essence, e.g., compliance with the “right to explanation” foreshadowed by the European Union’s General Data Protection Regulation (GDPR<sup>1</sup>) [50, 172, 173].

In such circumstances, creators of explainability methods can consult the taxonomy and use it as a *Work Sheet* to understand and operationalise their specific requirements without overlooking some obscure properties. A similar approach can be taken retrospectively with regard to preëxisting explainers by using the taxonomy as a *Fact Sheet*, which enables a systematic evaluation and comparison of these methods. We demonstrate the usefulness and practical aspects of these explainability *Fact Sheets* in Appendix A, which holds their particular instantiation created for LIME [129], CtreetX (Chapter 4) and LIMETree (Chapter 5).

## 2.2 Taxonomic Ranks: Dimensions of Explainability

Our XAI taxonomy is intended to probe explainability systems along five dimensions, which are summarised in Table 2.1. First, the *functional* one, which considers algorithmic requirements such as the type of applicable predictive systems (classification, regression, etc.) and the component of a black box for which it is designed (data, models or predictions). This dimension also tackles the scope of the technique (e.g., local vs. global explanation) and its relation to the underlying model (post-hoc vs. ante-hoc), among others. Second, the *operational* dimension, which includes the type

---

<sup>1</sup><https://publications.europa.eu/s/inbX>

Functional	Operational	Usability	Safety	Validation
<b>F1</b> Problem Supervision Level	<b>O1</b> Explanation Family	<b>U1</b> Soundness	<b>S1</b> Information Leakage	<b>V1</b> User Studies
<b>F2</b> Problem Type	<b>O2</b> Explanatory Medium	<b>U2</b> Completeness	<b>S2</b> Explanation Misuse	<b>V2</b> Synthetic Experiments
<b>F3</b> Explanation Target	<b>O3</b> System Interaction	<b>U3</b> Contextfullness	<b>S3</b> Explanation Invariance	
<b>F4</b> Explanation Breadth/Scope	<b>O4</b> Explanation Domain	<b>U4</b> Interactiveness	<b>S4</b> Explanation Quality	
<b>F5</b> Computational Complexity	<b>O5</b> Data and Model Transparency	<b>U5</b> Actionability		
<b>F6</b> Applicable Model Class	<b>O6</b> Explanation Audience	<b>U6</b> Chronology		
<b>F7</b> Relation to Predictive System	<b>O7</b> Function of Explanation	<b>U7</b> Coherence		
<b>F8</b> Compatible Feature Types	<b>O8</b> Causality vs. Actionability	<b>U8</b> Novelty		
<b>F9</b> Caveats and Assumptions	<b>O9</b> Trust vs. Performance	<b>U9</b> Complexity		
	<b>O10</b> Provenance	<b>U10</b> Personalisation		
		<b>U11</b> Parsimony		

**Table 2.1:** Overview of the explainable artificial intelligence taxonomy.

of interaction with the end user, dichotomy between explainability and predictive performance, and the user’s background knowledge required to fully utilise the embodied explainability power, to name a few. Third, the *usability* dimension, which takes a user-centred perspective and deals with properties of the explanation that make it feel “natural” and easy to comprehend by the explainee. To fulfil the user’s expectations these include: fidelity of the explanation, its actionability from the explainee’s perspective and its brevity, to give just a few examples.

The fourth dimension is *safety*, which discusses robustness and security of an explainability approach as well as any hazards associated with it. For example, this includes an analysis of how much information about the predictive model or its training data an explanation leaks and whether an explanation for a fixed data point is consistent throughout different models given the same training data. Finally, the taxonomy deals with the *validation* process used to evaluate and prove the effectiveness of an explainability approach by auditing a user study or synthetic verification that was carried out. Since the taxonomy can evolve over time, we encourage the users to systematically *version* its implementations [15], such as *Fact Sheets* and *Work Sheets*, thereby making the recipients aware of any updates. Furthermore, given that the taxonomy is applicable to a (theoretical) algorithmic approach, an actual implementation or a mixture of the two, its operationalisation should clearly indicate how the properties of interest correspond to these. To put the dimensions of our proposed taxonomy into context, exemplar explainability *Fact Sheets* for (A.1) LIME, (A.2) CtreeX and (A.3) LIMETree are provided in Appendix A.

### 2.2.1 Functional Requirements

These can help to determine *whether a particular approach is suitable for a desired application*, therefore they resemble a classification of machine learning and explainability approaches. The list of nine functional requirements provided below (**F1–F9**) can be thought of as a check-list of an engineer who was tasked with identifying and deploying the most suitable explainability algorithm for a particular use case. All of these properties are well-defined and flexible enough to accommodate any explainability approach.

#### F1 Problem Supervision Level

An explainability approach can be applicable to any of the following learning tasks: *unsupervised*, *semi-supervised*, *supervised* and *reinforcement*. A large part of the literature focuses on supervised learning, where explanations serve as a justification of a prediction. Nevertheless, explainability can also benefit unsupervised learning, where the user may want to learn about the data insights elicited by a model; reinforcement learning, where the user is interested in autonomous agent’s decisions; and semi-supervised learning, where the user can help the system choose the most informative data points for learning by understanding the system’s behaviour.

**F2 Problem Type**

We can identify four main problem types in machine learning: *classification* (binary/multi-class/multi-label and probabilistic/non-probabilistic), *regression*, *reinforcement learning*, and *clustering*. (Additionally, we can consider types such as ranking and collaborative filtering.) By clearly defining the applicable type of a learning task for an explainability method, potential users can easily identify ones that are useful to them.

**F3 Explanation Target**

The machine learning process has three main components: *data* (both raw data and features), *models* and *predictions*. Explaining the first one may be difficult or even impossible without any modelling assumptions; these are usually summary statistics, class ratio, feature plots, feature correlation and dimensionality reduction techniques. Note that a data collection process requires adopting a view or model of the world and the underlying (natural) phenomenon as well as physical characteristics of the tools used to measure the attributes of interest, thereby expressing data within this implicit and possibly subjective framework. Explaining models is concerned with its general functionality and conveying its conceptual behaviour to the explainee. Explaining predictions provide a rationale behind the model's output for any particular data point.

**F4 Explanation Breadth/Scope**

This notion varies across data, models and predictions. It tells the user to what extent an explanation can be generalised. (See **U3** for a complementary view on this property from the *usability* perspective.) The main three explanation generalisability stages are: *local* – a single data point or a prediction; *cohort* – a subgroup in a data set or a subspace in the model's decision space; and *global* – a comprehensive model explanation.

**F5 Computational Complexity**

Given that some applications may have either *time*, *memory* or *computational power* constraints, each explainability approach should consider these. If, for example, a given method is capable of explaining both a single prediction and the whole model, both of these aspects should be discussed. *Algorithmic complexity* measures such as Big-O or Little-O notations can be used to assess these aspects of an explainable system. Alternatively to theoretical performance bounds, empirical evaluation can also be discussed, e.g., the average time over 1,000 iterations that it took to generate an explanation for a single data point with fixed parameters of the explainability algorithm on a single-core CPU with 16GB of RAM.

**F6 Applicable Model Class**

Every explainability method is designed to either work with a particular model class or it is model-independent. We can identify three main degrees of *portability* [131] for explainability algorithms: *model-agnostic* – working with any model family; *model class-specific* – designed for a particular model family, e.g., logical or linear models; and *model-specific* – only applicable to a particular model, e.g., decision trees.

**F7 Relation to the Predictive System**

We can characterise two main relations between a predictive model and an explainability technique. *Ante-hoc* approaches use the same model for predicting and explaining, e.g., explaining a linear regression with its feature weights. It is important to note that some of these techniques may come with caveats and assumptions about the training data or the training process (see **F9**), which need to be satisfied for the explanation to work as intended. With *post-hoc* approaches, on the other hand, predictions and explanations are made with different models, e.g., a local surrogate explainer. One can also name a third type – a special case of the post-hoc family – a (local, cohort or global) *mimic approach*, where to explain a complex (black-box) model a simpler (inherently transparent) model is built in an attempt to mimic the behaviour of the more complex one, e.g., a global surrogate explainer.

**F8 Compatible Feature Types**

Not all models are capable of handling all the feature types, for example, categorical features are at odds with predictive algorithms that use optimisation as their backbone. Furthermore, selected model implementations require categorical features to be pre-processed, one-hot encoded for example, rendering them incompatible with some explainability approaches. Therefore, every method should have a clear description of compatible feature types: *numerical*, *ordinal* (we differentiate numerical and ordinal features as the latter may have a bounded range) and *categorical*. In addition to these standard feature types, some tasks may come with a hierarchy of features and/or their values, in which case the explainability algorithm should clearly state whether these are beneficial for the quality of the resulting explanation and how to utilise this information.

**F9 Caveats and Assumptions**

Finally, any *functional* aspects of an explainability approach that do not fit into the previous categories should be included under this catch-all item. In particular, restrictions with respect to input and output of predictive models and explainability techniques [108]. These may include: support for black-and-white images only; validated behaviour on text corpora up to 100 tokens; numerical confidence of a prediction or an explanation; assumptions such as *feature independ-*

ence; the effect of correlated features on the quality of an explanation; or explainer-specific requirements such as *feature normalisation* when explaining a linear model with its weights.

### 2.2.2 Operational Requirements

The following ten properties (**O1–O10**) characterise *how users interact with an explainability system and what is expected of them*. These requirements can be thought of as considerations from a deployment point of view.

#### O1 Explanation Family

A very useful categorisation, which we believe is still up to date, of explainability approaches accounting for their presence in philosophy, psychology and cognitive science was introduced by Johnson and Johnson [64] for expert systems. The authors have identified three main types of explanations: *associations between antecedent and consequent*, e.g., model internals such as its parameters, feature(s)–prediction relations such as explanations based on feature attribution or importance and item(s)–prediction relations [76] such as influential training instances [74] (or neighbouring data points); *contrasts and differences* (using examples), e.g., prototypes and criticisms [68, 70] (similarities and dissimilarities) and class-contrastive counterfactual statements [106]; and *causal mechanisms*, e.g., a full causal model [118].

#### O2 Explanatory Medium

An explanation can be delivered as a: (statistical) *summarisation*, *visualisation*, *textualisation*, *formal argumentation* or mixture of the above. Examples of the first one are usually given as numbers, for example, coefficients of a linear model or summary statistics of a data set. The second one comprises all sort of plots that can be used to help the user comprehend behaviour of a predictive system, e.g., Individual Conditional Expectation [48] or Partial Dependence [44] plots. Textualisation is understood as any explanation in form of a natural language description, e.g., a dialogue system that can be queried by an explaine. Explainers based on a formal argumentation framework [35] encompass approaches that can output logical reasoning in support of an explanation, hence provide the explaine with an opportunity to argue against it, whether in a form of a natural language conversation or highlighting important regions in an image. Finally, an example of a mixture of these representations can be a plot accompanied by a caption that helps to convey the explanation to the end user. Such a mixture may be necessary at times as not all of the media are able to communicate the same amount or type of information [37]. For example, visualisations are confined to three dimensions (four when counting time, i.e., animations) due to the limitations of the human visual perception system and counterintuitiveness of higher dimensions – a phenomenon known as the curse of dimensionality.

The choice of an explanatory medium is also important as it may limit the *expressive power* of an explanation.

### **O3 System Interaction**

The communication protocol that an explainability method employs can either be *static* or *interactive* (with and without user feedback). The first one is a one-size-fits-most approach where the system outputs an explanation based on a predefined protocol specified by the system designer, hence may not always satisfy the user's expectations, for example, always outputting the most significant factors in favour and against a particular prediction when the user is interested in a feature not included therein. Alternatively, the system can be interactive, thereby allowing an explaineo to explore all aspects of a particular explanation. These include interactive user interfaces and dialogue systems, among others. Furthermore, in case of an interactive system, its creator should indicate whether the explainer can incorporate any feedback (and in what form) given by the explaineo and how, if at all, it influences the underlying predictive model (e.g., incremental-learning algorithms) [81].

### **O4 Explanation Domain**

Explanations are usually expressed in terms of the underlying model's parameters or data exemplars and their features – *original domain*. However, it may be the case that the explanation is presented in a different form – *transformed domain*. Consider, for example, a system explaining image classification results where the data domain is an image and the explanation is a natural language description as opposed to a saliency map superimposed onto the image. Another approach can be an *interpretable data representation*, e.g., super-pixels instead of raw pixel values, introduced by Ribeiro et al. [129] as part of the LIME algorithm (cf. **O5**).

### **O5 Data and Model Transparency**

An explainability approach should clearly indicate whether the underlying predictive algorithm and/or the (training) data are expected to be *transparent*, or they can be *opaque*. (This requirement is tightly related to **F7**, in particular when we are dealing with ante-hoc explainability approaches.) In case of model explanations, does an explaineo need to understand the inner workings of a predictive model? When data or predictions are being explained, do the data features need to be human-understandable in the first place? (This concept, in turn, is related to Lipton's [96] validation approaches discussed in the last paragraph of Section 2.2.5: what sort of understanding of the model and/or features is expected of the user.) For example, consider explaining a prediction in terms of a room temperature as opposed to using a squared sum of a room temperature and its height. In cases where the input domain is incomprehensible, the system designer may decide to give a list of meaningful data transformations as a remedy or choose

an exemplar-based explainer instead [138]. For example, applying a super-pixel segmentation to an image and using its output as higher-level features that are intelligible to humans can help to explain image classification tasks, as shown by LIME.

## **O6 Explanation Audience**

The intended audience of an explainability method may range from a *domain expert*, through a requirement of a *general knowledge about a problem*, all the way to a *lay audience*. Considering the type of the domain expertise is also important: ML and AI knowledge can be distinct from domain knowledge. Therefore, discussing the level and type of background knowledge required to comprehend an explanation is crucial [20, 124, 167]. Certain techniques may allow to adjust (**U4** and **U5**) the size (**U11**) or complexity (**U9**) of an explanation based on the intended audience (**U10**) via some of the *usability* requirements (see Section 2.2.3). Furthermore, the transparency of the features (and/or the model – refer to **O5**) should be judged with respect to the explanation recipients. For example, consider a system that explains its predictions using natural language sentences. Given the language skills of the recipient, the system can use text of varying lexical and grammatical complexity to facilitate its easier comprehension [174]. Finally, the system may be able to adjust the granularity of an explanation to suit the recipients’ needs based on their cognitive capacity, e.g., explaining a disease diagnosis to doctors as opposed to patients or their families. One of the goals of decreasing the complexity of an explanation (which is sometimes at odds with its fidelity, cf. **U1** and **U2**) may be making it easy enough for the explainees to comprehend it in full [96], hence enable them to simulate the decisive process *in vivo*, i.e., simulatability. (See the validation requirements discussed in the last paragraph of Section 2.2.5 for a detailed description of this concept.)

## **O7 Function of the Explanation**

Every explainability approach should be accompanied by a list of its intended applications [76]. Most of them are designed for transparency: *explaining* a component of the ML pipeline to an end user, whether it is to support decisions, compare models, elicit knowledge from a black box or the data used to build it, or extract a causal relation. Nonetheless, some of them can also be used to assess *accountability* of the underlying predictive model, e.g., debug and diagnose it to engender trust; or demonstrate its *fairness*, e.g., uncover disparate treatment with counterfactuals. It is important to provide the user with the envisaged (and validated) deployment context to prevent explainer misuse, which may lead to an unintentional harm when deployed in high-risk applications or autonomous systems.

## **O8 Causality vs. Actionability**

Most explanations are not of a causal nature. If this is the case, lack of causal relations needs to be explicitly communicated to the users so that they can avoid drawing incorrect conclusions. Given

an *actionable*, and not causal, explanation, the users should understand that the insight provided by the explanation will result in, say, a different classification outcome, however interpreting it causally, no matter how tempting, can lead to inaccurate conclusions [118]. Similarly, explanations that are derived from a full causal model should be advertised as *causal* and used to their full potential. This concept is closely related to **O1**, which specifies the explanation family.

## **O9 Trust vs. Performance**

All of the explainability approaches should be accompanied by a critical discussion of performance–explainability trade-offs that the user has to face. By and large, explainability can improve the user’s *trust* in a predictive system, but sometimes the decrease in *predictive performance* (if any) that is associated with making a particular system more explainable may not be worth it. For example, consider a case where making a predictive algorithm explainable renders its predictions to be incorrect most of the time. Therefore, the user of an explainability method needs to decide whether the main objective of a predictive system is to make it more efficient or learn something from data. While the existence of a clear-cut performance–explainability dichotomy has recently been questioned [133], an insightful discussion on this topic published along each explainer could provide a rich source of data and evidence for a scientific evaluation of this phenomenon and its existence.

## **O10 Provenance**

Finally, the operational requirements should record the provenance of an explainability system and of the explanations that it produces, i.e., be translucent [131] about the information that contribute to them. Most often, provenance of an explanation can be attributed to its reliance on: a *predictive model* – achieved via interacting with the (black-box) model or using its internal representation (glass-box) [76]; a *data set* – introduced by inspecting or comparing data points originating from one or a mixture of the training, evaluation and validation data sets; or, ideally, *both* a predictive model and a data set. An example of a purely model-driven explanation is interpreting a *k*-means model with its centroids. An exclusively data-driven explanation is, for example, explaining predictions of a *k*-nearest neighbours model by accounting only for the *k* neighbours closest to the data point being explained. If possible, every explanation should be accompanied by an *explainability trace* indicating which training data points were influential for a prediction [74] and the role that the model and its parameters played. In most of the cases, a model-specific explainability algorithm (**F6**) will rely heavily on internal parameters of the underlying predictive model, whereas a model-agnostic approach will depend more on data and behaviour of a predictive model.

### 2.2.3 Usability Requirements

Here, we discuss eleven *properties of explanations that are important from an explainee’s point of view* (U1–U11). Many of these are grounded in social science research and aim at – whenever applicable – making algorithmic explanations feel more natural to the end users regardless of their background knowledge and prior experience with this type of a system or technology in general.

#### U1 Soundness

This property measures how *truthful* an explanation is with respect to the underlying predictive model [81] (sometimes called concordance). Its goal is to quantify local or global adherence of the explanation to the black box. If the explanation is of the *ante-hoc* type, this property does not apply as both the explanation and the prediction are derived from the same model. However, *post-hoc* (or *mimic*) explanations should measure and report this property to quantify the error introduced by the explainability technique. This can be done by comparing a selected performance metric between the outputs of predictive and explanatory models, e.g., average rank correlation between the two. A high value of such a metric would assure the user that an explanation is consistent and aligned with predictions of the underlying model. This requirement can also be understood as “truthfulness” of an explanation – Grice et al. [51] have noted in their *maxim of quality*, which is one of the rules for coöperative communication, that a user should only be presented with claims supported by evidence. Soundness is one of the two explanation *fidelity* measures, with the other one being *completeness* (U2).

#### U2 Completeness

For an explanation to be trusted, it also needs to *generalise* well beyond the particular case for which it was produced. This mostly applies to *local* and *cohort* explanations as the user may want to apply insights learnt from one of these explanations to a “similar” case: *pars pro toto*. Completeness measures how well an explanation generalises [80, 98, 107], hence to what extent it *covers* the underlying predictive model. This property can be quantified by checking correctness of an explanation across similar data points (individuals) across multiple groups within a data set. In particular, cardinality of a support set – the number of instances to which the explanation applies divided by the total number of instances – can be used to measure completeness. Given the context-dependent nature of this metric, there is no silver bullet to assess how well an explanation *encompasses* the model. In addition to *soundness*, this is the second explanation *fidelity* metric.

### U3 Contextfulness

If there are known issues with completeness of an explanation, a user may not trust it. To overcome this and help the user better understand how an explanation can be generalised, it can be framed in a *context*, thereby allowing the user to assess its *soundness* and *completeness*. For example, the user will better recognise the limitations of an explanation if it is accompanied by all the necessary conditions for it to hold, critiques (i.e., explanation oddities) and its similarities to other cases [68, 70, 81, 106]. Contextfulness can help to make a local explanation either explicitly local, allow the explainee to safely generalise it to a cohort-based explanation, or even indicate that despite it being derived for a single prediction it can be treated as a global one. A specific (quantitative) case of this property, called *representativeness*, aims to measure how many instances in a (validation) data set does a single explanation cover, akin to the support set cardinality. Another aspect of contextfulness is the degree of importance for each factor contributing to an explanation. For example, if an explanation is supported by three causes, how important are they individually? One observation worth making is that an order in which they are presented rarely ever indicates their relative importance, e.g., a list of conditions in a decision rule. This usability requirement can also be compared to the *maxim of manner* (from the rules for coöperative communication [51]) that entails being as clear as possible in order to avoid any ambiguity that may lead to confusion at any point.

### U4 Interactiveness

As explainees may have a broad range of experience and background knowledge, a single explanation rarely satisfies a potentially wide spectrum of their expectations. To improve the overall user experience the explanation process should be controllable. For example, it should be *reversible* (in case the user inputs a wrong answer), respect *user's preferences and feedback*, be “*social*” (bidirectional communication is preferred to one-way information offloading), allow to adjust the *granularity* of an explanation, and be *interactive* [68, 70, 80, 81, 106, 167, 178]. This means that, whenever possible, the users should be able to *customise* and *personalise* the explanation that they get to suit their needs [138]. For example, if the system explains its decisions with counterfactual statements and a foil used in such a statement does not contain information that the users are interested in, they should be able to request an explanation conditioned on the desired foil (if one exists).

### U5 Actionability

When an explanation is provided to help users understand a reason behind an algorithmic decision, then the users prefer explanations that they can treat as *guidelines* towards the desired outcome [77, 166]. For example, in a banking context, given an explanation based on counterfactual statements, it is better (from the user's perspective) to get a statement conditioned

on a number of active loans rather than the user's age. The first one provides the user with an action towards the desired outcome (i.e., pay back one of the loans before reapplying), while the latter leaves the user without any options.

### **U6 Chronology**

Some aspects of an explanation may have inherent time ordering, for example, loans taken by a borrower. In such cases, if one of the reasons given in an explanation has a timeline associated with it, users prefer explanations that account for *more recent* events as their cause, i.e., proximal causes [106]. For example, consider multiple events of the same type contributing equally to a decision: an applicant has three current loans and will not be given a new one unless an outstanding one is paid back. Repaying any of the three loans is a sufficient explanation, however from the user's perspective taking the most recent loan is a more natural reason behind the unfavourable loan application outcome than having any of the first two loans.

### **U7 Coherence**

Some users of explainability systems may have prior background knowledge and beliefs about the matter that is being predicted and explained – their mental model of the domain. In such cases, any resulting explanation should be *consistent* with the explainee's prior knowledge [37, 98, 107], which can only be achieved when the explainee's mental model is part of the explainability system (U10) as otherwise there is nothing to be coherent with. (A mental model can either be *functional*, i.e., shallow, in which case the end users know how to interact with something but not how it works in detail; or *structural*, i.e., deep, in which case they have a detailed understanding of how and why something works [79].) If a prediction of a black-box model is consistent with the users' expectations, then the reasoning behind it will not be contested most of the time unless its logic is fundamentally flawed (internal inconsistency) or it is at odds with the general knowledge [174] – humans tend to ignore information that is coherent with their beliefs (confirmation bias). If the output of a predictive model is unexpected, on the other hand, the users will contrast the explanation against their mental model to understand the prediction, in which case the explainer should identify and fill in these knowledge gaps [64]. Therefore, if an explanation uses arguments that are consistent with the users' beliefs, they will be more likely to accept it. While this property is highly subjective, basic coherence with the universal laws, e.g., number ordering, should be satisfied.

### **U8 Novelty**

Providing users with a mundane or expected explanation should be avoided. Explanations should contain surprising or abnormal characteristics (that have low probability of happening, e.g., a rare feature value) to point the user's attention in an interesting direction [19, 80, 106]

(recall **U7** where anomalies prompt the user to request an explanation). However, this objective requires balancing the trade-off between coherence with the explainee’s mental model, novelty and overall plausibility [174]. For example, consider an ML system where explanations help to better understand a given phenomenon. In this scenario, providing the users with explanations that highlight relations that they already know should be avoided. The explainees’ background knowledge should be considered before producing an explanation to ensure that it is novel and surprising; at the same time, consistency with their mental model should be preserved as long as it is correct. Again, this usability criterion can only be built on top of the explainees’ mental models since this knowledge is essential for assessing novelty of causes.

### **U9 Complexity**

Given a wide spectrum of explainees’ skills and background knowledge, the *complexity* of explanations should be tuned to the recipients [106, 167]. This can be an operational property of an explainable system (**O6**), however it is also important to consider it from a user’s perspective. If the system does not allow for explanation complexity to be *adjusted* by the user, it should be as *simple* as possible by default (unless the explainee explicitly asks for a more complex one). For example, given an explanation of an automated medical diagnosis, it should use observable symptoms rather than the underlying biological processes responsible for the condition. Choosing the right complexity automatically may only be possible given the availability of the explainee’s mental model.

### **U10 Personalisation**

Tuning an explanation to its intended recipients requires the explainability technique to approximate explainees’ background knowledge and mental model [138, 178]. This is particularly important when attempting to adjust the complexity of an explanation (**U9**) as well as its novelty (**U8**) and coherence (**U7**). An explanation can either be personalised *on-line* via an interaction or *off-line* by incorporating the necessary information into the model (e.g., parameterisation) or data. Personalising an explanation is related to yet another rule of cooperative communication [51]: the *maxim of relation*. According to this rule, a communication should only relay information that are relevant and necessary at any given point in time. Therefore, an explainability system has to discern what the user knows and expects in order to determine the content of the explanation [37].

### **U11 Parsimony**

Finally, explanations should be *selective* and *succinct* enough to avoid overwhelming the explainee with unnecessary information, i.e., fill in the most knowledge gaps with the fewest arguments [81, 98, 106, 107, 174]. This is somewhat connected to explanation novelty (**U8**) as it can be partially attained by avoiding premisses that an explainee is already familiar with.

Furthermore, parsimony can be used as a tool to reduce complexity (**U9**) of an explanation regardless of the explainee’s background knowledge. For example, *brevity* of a counterfactual explanation can be achieved by giving as few reasons (number of conditions) in the statement’s foil as possible. This requirement is also related to another rule of coöperative communication presented by Grice et al. [51]; the *maxim of quantity* states that one should only communicate as much information as necessary and no more (a partial explanation).

## 2.2.4 Safety Requirements

Explainability can become a toolkit for improving trust and validating safety of black-box AI systems [54], however in certain cases it may also cause unintended harm and have adverse consequences. Explainers tend to reveal partial information about the data used to train predictive models, these models’ internal mechanics or parameters, and their prediction boundaries. Therefore, our taxonomy considers *the effect of explainability on robustness, security and privacy aspects of predictive systems* which they are built on top of as well as robustness of explainers and explanations themselves [157] (**S1–S4**), outlining all the known hazards that come into play when they are deployed.

### S1 Information Leakage

Every explainability approach should be accompanied by a critical evaluation of its privacy and security implications and a discussion about mitigating these factors. It is important to consider how much information an explanation reveals about the underlying model and its training data, as well as consequences of this leakage. For example, consider a counterfactual explanation generated for a logical machine learning model; given that this model family applies precise thresholds to data features, this type of an explanation is likely to disclose them. Similarly, explanations of a  $k$ -nearest neighbours model can reveal training data points and explaining a support vector machine classifier can leak data points constituting the support vectors. Another example can be a security trade-off between *ante-hoc* approaches that reveal information about the predictive model itself and local *post-hoc* explanations that can only leak behaviour of the decision boundary in the neighbourhood of a selected point. We could partially mitigate these threats by increasing the parsimony of explanations (**U11**), producing explanations for aggregated data or obfuscating the exact thresholds or data points. This can be achieved by  $k$ -anonymising [134] the data, outputting fuzzy thresholds in the explanations or providing general directions of change (e.g., “slightly more”) to avoid giving out the exact values, among many others.

## S2 Explanation Misuse

With information leakage in mind one can ask: How many explanations and of how many different data points does it take to gather enough insight to steal or game the underlying predictive model? This can be a major concern especially if the model is obfuscated to protect a trade secret. Furthermore, explanations can be used by adversaries to game a model; consider a case where a malicious user was able to find a bug in the model by inspecting its explanations, hence is now able to take advantage of it. While the explainer may have been designed for debugging a predictive model and identifying its vulnerabilities, e.g., discovering its non-monotonic behaviour, the net outcome of these actions ultimately depends on the explaine. In particular, this observation indicates a close relation and fine line between explanations and adversarial attacks [49]. Therefore, having a clear target audience in mind (**O6**) is crucial since explanation misuse is closely linked to a consideration of the intended application of an explainability system (**O7**). For example, a system designed as a certification tool will usually reveal more information than one providing explanations to customers.

## S3 Explanation Invariance

Given a phenomenon to be modelled by a predictive system, (training) data that we gather are just a way to quantify its observed effects (see **F3** for more details). Therefore, the objective of a predictive system should be to elicit insights about the underlying phenomenon and the explanations ought to be a medium to foster their understanding in a human comprehensible context. Ideally, explanations should be based on a property of the underlying phenomenon rather than an artefact of a black-box model, which may require a proper causal model [119]. In this setting it is natural to expect an explainability system to be [59]:

**consistent** Explanations of “similar” data points should be similar for a fixed model (training procedure and data) and explanations of a fixed data point should be comparable across different predictive models or different training runs of the same model (trained using the same data).

**stable** An explainability approach should provide the same explanation given the same inputs (model and/or data point). This can be measured by investigating variance of an explanation over multiple executions of an explainability algorithm.

Furthermore, explanations produced by one method should be comparable to those produced using another explainability technique (given fixed training data). If one of these properties does not hold, the designer of an explainer should investigate how model configuration and parameterisation influence its explanations. Such inconsistency – where the same event is given different, often contradictory, explanations by different actors (explainable algorithms in our

case) – is well documented in the social sciences as “The Rashmon Effect” [31] and should be avoided.

#### **S4 Explanation Quality**

The final safety requirement concerns evaluating the quality and correctness of an explanation with respect to the “confidence” of the underlying predictive model and the distribution of its (training) data. This criterion is in place as poor predictive performance, whether overall or for specific data points, usually leads to uninformative explanations. After all, if a prediction is of subpar quality, it would be unreasonable to expect its explanation to be sensible. We suggest for each explanation to be accompanied by a list of uncertainty sources, one of which may be the predictive confidence [121] assigned by the underlying model to the explained instance. For example, if a method relies on synthetic data (as opposed to real data), this should be clearly stated as a source of variability, hence randomness and uncertainty. Another example of an explanation that does not convey its possibly inferior quality is a counterfactual that lies in a sparse region of the (training) data distribution – since we have not seen many data points in that region, we should not trust the explanation without further investigation [90, 123]. Explanation multiplicity – existence of numerous explanations for each model or prediction – can also add to this problem as the explainee needs to prioritise such insights, especially when some of them are divergent or contradictory.

#### **2.2.5 Validation Requirements**

Finally, *explainability systems should be validated* with user studies (**V1**) or synthetic experiments (**V2**) *in a setting similar to the intended deployment scenario*. This research area has seen increasing interest in the recent years with Doshi-Velez and Kim [34] providing evaluation criteria and proposing various approaches to validate explainers. Other researchers [124, 167] highlighted the importance of considering the stakeholders of an explanation before validating it, akin to the intended application (**O7**) and audience (**O6**) sections included in our taxonomy. Nevertheless, this research branch lacks a universal consensus regarding a validation protocol, which hinders the progress of explainable AI research by making explainability methods incomparable. A commonly agreed validation protocol could help to:

- eliminate *confirmation bias* – when two explanations are presented side by side,
- mitigate *selection bias* – when a study is carried out via Amazon Mechanical Turk all of the participants are computer-literate,
- avoid *outcome bias* – when an explanation supports a given prediction or it agrees with the explainee’s mental model regardless of the prediction, and

- fight a phenomenon called *The Illusion of Explanatory Depth* [132] – to overcome explanation ignorance.

For example, when users are asked to choose the best explainability approach out of all the options presented to them, they should not be forced to choose any single one unless they consider at least one of them to be useful. When validating explainers, one should also be aware of a phenomenon called *Change Blindness* [145] – humans’ inability to notice all of the changes in a presented medium – which is especially prominent for visual explanatory media (**O2**), e.g., images. A good practice in such cases is ensuring that two explanations, or the instance to be explained and the explanation, are clearly distinguishable by, for example, highlighting the differences (**U9**).

Furthermore, a well-designed user study could also provide a clear answer to some of the (qualitative) explanation properties listed in the previous sections. For example, it can help to evaluate the effectiveness of an explanation for a particular audience (**O6**), assess the background knowledge necessary to benefit from an explanation (**O5**) or check the level of technical skills required to use it (**O2**) as not all explainees may be comfortable with a particular explanatory medium. Since user studies are the most acceptable approach to validate the explanatory powers of a new method, an XAI-specific protocol – such as randomised controlled trials in medical sciences – should be developed to make the evaluation results comparable and prevent influence of various biases. For example, Doshi-Velez and Kim [34] identified three types of evaluation approaches:

**application-level** Validating an explainability approach on a *real task* with user studies (**V1**), e.g., comparing explanations given by doctors (*domain experts*) against an explainability algorithm for X-ray imaging.

**human-level** Validating an explainability approach on a *simplified task* (within the same domain) and a lay audience (to avoid using domain experts whose time is often scarce and expensive) with user studies (**V1**), e.g., an Amazon Mechanical Turk experiment asking the explainees to choose the most appealing explanation from a range of different techniques.

**function-level** Validating an explainability approach on a *proxy task*, i.e., synthetic validation (**V2**). For example, given an already proven explainability method such as explaining decision trees by visualising their structure, a proxy can be its measure of complexity given by the tree depth or width.

A different set of, mostly synthetic (**V2**), validation approaches was proposed by Herman [59]:

- using simulated data with known characteristics to validate correctness of explanations, and

- testing stability and consistency of explanations – see the invariance safety requirement (**S3**) for more details.

The latter validation strategy can be either *quantitative* (**V2**), given a well-defined metric, or *qualitative* (**V1**), given user’s perceptual evaluation.

Lastly, Lipton [96] has come up with three distinct approaches to evaluate how well an explanation is understood based on user studies (**V1**):

**simulatability** Measuring how well a human can recreate or repeat (simulate) a black-box computational process based on its explanations, for example, by asking the explainee a series of counterfactual what-if questions.

**algorithmic transparency** Measuring the extent to which a human can fully understand a predictive algorithm: its training procedure, provenance of its parameters and the process governing its predictions.

**decomposability** Quantifying the ability of an explainee to comprehend individual parts (and their functionality) of a predictive model: understanding input features, model parameters (e.g., a monotonic relationship of one of the features) and outputs of the model.

Lipton’s [96] motivation is to gauge explainees’ understanding, which resembles our definition of explainability presented in Chapter 1 (transparent insights leading to understanding), however his evaluation criteria expose that the two are based on fundamentally different premisses. Lipton’s first two metrics arguably fail to measure (human) understanding, whereas the last one highlights various explanation targets (data, models and predictions according to **F3**), their assumed transparency (**O5**) as well as model and explanation provenance (**O10**), all of which are covered by our taxonomy. The type of understanding that follows from *simulatability* is embodied by a functional mental model (Section 1.1), which is relatively superficial and mechanistic. This criterion measures whether a human can replicate behaviour of a predictive black box, which we deemed insufficient (Section 1.1.3) based on The Chinese Room Argument [139]. Similarly, *algorithmic transparency* entails an in-depth appreciation of how a black box operates, i.e., developing a structural mental model (Section 1.1), which may require machine learning expertise and leads to understanding the technical process but not necessarily the reason for outcomes.

## 2.3 Taxonomy Trade-offs

Multiple questions arise when developing explainers or evaluating them based on our list of desiderata. Are all of the properties equally important? Are they compatible with each other or are some of them at odds? In practice, many of these characteristics cannot be attained at the same time and their respective importance and prioritisation often depend on the *application area* [37]. Even though certain explanation types may be flexible enough to comply with most

of the requirements in theory, e.g., counterfactuals, some of these properties can be lost in implementation due to particular algorithmic choices. While both *functional* and *operational* requirements are properties of a specific explainability approach and its implementation, the *usability* desiderata are general properties and any explainer should aim to satisfy all of them. For example, making an explainability system *model-agnostic* forces it to be a *post-hoc* (or a *mimic*) technique and prevents it from taking advantage of specifics of a particular black-box model implementation. Furthermore, in contrast to *ante-hoc* techniques, such approaches create an extra layer of complexity on top of the predictive model, which can be detrimental to explanation *fidelity* – a trade-off between *completeness* and *soundness* that is common to *model-agnostic* explainers.

Some of the trade-offs affecting design and implementation of explainability systems have already been observed in the literature. Lombrozo [98] points out that explanations that are *simpler* (U11), i.e., with fewer causes, *more general* (U2) and *coherent* (U7) are usually more appealing to humans, however depending on the application (O7) and target audience (O6), this may not always be desirable. Moreover, when considering *coherence* of an explanation, we may run into difficulties defining the complement of the concept being explained, which may simply be ill-defined – consider a *non-concept* [112] such as not-a-car. Kulesza et al. [80], on the other hand, show that both *completeness* (U2) and *soundness* (U1) are important, however if faced with a trade-off, one should choose the former over the latter. Notably, Eiband et al. [37] point out that this is not a universal principle and the selection largely depends on the application domain. Similarly, Walton [174] argues that users prefer explanations that are *more plausible* (U1), *consistent with multiple outcomes* (U2), i.e., explain many things at once, and *simple* (U9, U11). While daunting, all of these dichotomies are important to consider as they can help to identify and make informed choices about the trade-offs that every explainability method is facing.

In particular, vanilla counterfactual explanations prioritise *completeness* over *soundness* as they are always data point-specific. Nevertheless, Miller [106] shows that, in theory, counterfactuals – which he considers the most human-friendly explanation type since they are contrastive and answer a “Why?” question – can satisfy most of the desiderata and the aforementioned observation is an artefact of algorithmic implementations. Moreover, he shows that based on social sciences research some of the properties of explainability systems should be prioritised:

- *necessary causes* (U2, U3) are preferred to *sufficient* ones;
- *intentional actions* (U10, U7) form more appealing explanations than those taken without deliberation;
- the *fact* and the *foil* of a (counterfactual) explanation should be clearly *distinguishable* (U9);
- *short* and *selective* (U11) explanations are preferred to *complete* ones;

- the *social context* (**U10**, **U9**, **O6**) should drive the content and the nature of an explanation; and
- one explanation covering *multiple phenomena* (**U2**) is preferred to a collection of unique explanations.

Some of these properties can be employed to achieve more than one goal. For example, *completeness* can be partially fulfilled by having *contextful* explanations. If an explainability system is not inherently *interactive*, this desideratum can be accomplished by deploying the explainer within an interactive platform such as a dialogue system for explanations delivered in natural language or an interactive web page for visualisations. *Actionability* and *chronology* are usually data set-specific and can be achieved by manually annotating features that are actionable and ordering the time-sensitive attributes. *Personalisation* – along with *coherence*, *novelty* and *complexity*, which all depend on it – is the most difficult criterion to be satisfied. On the one hand, we can argue that the *complexity* (as well as *novelty* and *coherence*) of an explanation may be adjusted by personalising it via system design (**O2**, **O6**, **O7**), through user interaction (**O3**, **U4**) or with parsimony (**U11**). Alternatively, we can imagine encoding a hierarchy of explanation complexity (based on the user’s mental model) and utilising this heuristic to serve explanations of desired complexity.

## 2.4 Applying the XAI Taxonomy

Systematically evaluating properties of explainability techniques can be a useful precursor to user studies (e.g., aiding in their design) to show their capabilities and compliance with the best practices in the field. Furthermore, despite theoretical guarantees of selected desiderata for some explainability systems, these properties can be lost in implementation. For example, model-agnostic explainers can render some desiderata difficult to achieve since these approaches cannot take advantage of model-specific aspects of predictive black boxes. LIME [129] has recently been subjected to studies aiming to validate its usability [89, 158, 183], which discovered that its explanations lack stability (**S3**) and struggle to capture locality (**U1**), thereby raising questions about the validation methods (**V2**) used to evaluate this technique in the first place. We concur that had a taxonomy such as one presented in this chapter been available, some of these issues could have been addressed early in the design and avoided. To support this claim we show examples of taxonomy-inspired explainability *Fact Sheets* in Appendix A, which closely inspects properties of LIME, CtreeX and LIMETree along all the five dimensions.

### 2.4.1 Target Audience

A comprehensive list of requirements expected of explainability systems spanning both their technical and social aspects is needed amid a lack of general consensus among researchers and

practitioners in this space. Some papers discuss a subset of the properties presented in Section 2.2, with many of them scattered throughout explainability literature, rendering it difficult to get a coherent and complete view on the matter. Having a taxonomy that collects and organises all of the requirements in one place will empower *designers* (researchers) and *users* of explainability systems to:

- guide the development, implementation, evaluation and comparison of explainers in a systematic and consistent manner;
- identify gaps in (theoretical) capabilities of explainers and divergence from those in their respective implementations; and
- quickly grasp properties of a given method to choose an explainer that is appropriate for the desired application – similarly to the food nutrition labels [180], the users know what to expect and how to navigate our taxonomy.

In addition to serving these two communities and their use cases, our list of desiderata can be utilised as a reporting tool aimed at regulators and certification bodies by providing them with the necessary information in a standardised and comparable format. Given the topic separation within the structure of our taxonomy, browsing through it should be sufficient to make explainability systems more appealing and transparent to a wider public, e.g., a non-technical audience.

### 2.4.2 Delivery Format and Medium

We chose to present our explainability requirements in the form of a taxonomy to empower the designers and users of explainability systems to make better choices and consider a wide spectrum of functional, operational, usability, safety and validation aspects when building or using explainers. Such a flexible and comprehensive structure makes the taxonomy suitable for a wide range of applications and allows its users to take as much, or as little, as they want from it rather than feel obliged to report on all the requirements in full length and complexity. We opted for introducing our desiderata of explainability systems as a “taxonomy” to convey this intrinsic flexibility of our framework, which otherwise could have been overlooked if presented as “fact sheet”, “work sheet”, “check list”, “standard”, “guideline” or “recommendation”. We acknowledge that our requirements list can form the basis of future standards or recommendations, but not being in a position to enforce this, we leave this task to bodies such as IEEE and their Global Initiative on Ethics of Autonomous and Intelligent Systems, which has already produced recommendations for Transparency and Autonomy [115]. Furthermore, posing our taxonomy as a “standard reporting tool” could undermine its adoption since enforcing standards may impede the pace of explainability research.

In addition to clear, transparent and well-defined structure and composition, framing our requirements list as a taxonomy has one major advantage: it can be adapted to serve as any of the aforementioned tools. For example, it can be used as an explainability *Fact Sheet* (examples included in Appendix A), which is a very common framework for self-reporting inspired by food nutrition labels [11, 15, 47, 60, 67, 108, 126, 180] – we discuss the adaptation of this concept in AI and ML more broadly in Section 2.5. Alternatively, the taxonomy can be used as a development and deployment *checklist* for explainability approaches as it has been shown that even experienced practitioners can make obvious mistakes despite their presence of mind, especially if working under stressful conditions or simply due to the repetitiveness of a task. While this use case is not very common in AI and ML, checklists have been demonstrated to eradicate most of such trivial, and often dangerous, human errors, e.g., checklists that help to account for all the tools used during a surgical procedure after it is finished [178]. A similar line of reasoning can be applied to designing and deploying explainers, but instead of a checklist, the user is provided with a taxonomy to aid critical evaluation of their capabilities and draw attention to their features that otherwise may have been overlooked.

Given the evolving nature of the requirements included in our taxonomy and the structure of its dimensions, we propose to host it on-line, accompanied by a collection of explainer-specific reference materials, i.e., *manuals*, within a single repository that will serve as their catalogue and a reference guide. Since the requirements can change over time with publication of new research, hosting the taxonomy and explainability method-specific manuals on-line will enable their natural evolution and encourage versioning, dissemination and revision supported by the community effort. Such contributions can be peer-reviewed and scrutinised following a process similar to OpenReview or review of open source software submitted to code hosting and versioning repositories such as GitHub. We anticipate that our taxonomy accompanied by such a collection of explainer-specific manuals and guidelines for their creation will become a go-to resource for learning about explainability of predictive AI systems.

### 2.4.3 Operationalisation

The effort required to apply our taxonomy, e.g., to create an explainer-specific manual such as a *Fact Sheet*, may seem prohibitively time consuming, thereby hindering its widespread adoption. However, doing so can be selective and incremental, especially that the process is not limited to the creator of an explainability technique and it is not required during the development of an explainer since it can be applied post-hoc. Nevertheless, we suggest consulting our taxonomy throughout the development of explainability systems as a reference material, a guideline and a *checklist* to adhere to best practices. The process of creating explainer-specific manuals can be further sped up by allowing the entire explainable AI community to contribute, which may even lead to improving the method itself by identifying its shortcomings and straightforward extensions. All of this is possible because our taxonomy can be applied retrospectively since it

only requires familiarity with the explainability approach or its implementation, unlike similar solutions for data sets [15, 47, 60] (which require knowledge of the data collection process) or AI services [11] (which are usually opaque to protect trade secrets). All things considered, we argue that researchers designing explainability methods and software engineers implementing them are best suited (and would benefit most) from applying our explainable AI taxonomy.

#### 2.4.4 Selecting Dimensions and Requirements

Certain requirements included in the taxonomy may appear very similar or strongly related to one another at first glance, hence their choice may seem arbitrary, with some arguing to merge or reorganise them. One reason for the repetitiveness of selected concepts is the fact that the requirements span five distinct dimensions some of which are presented from a social sciences perspective, e.g., the perception of users, while others are rather technical, e.g., deployment or performance. Another reason for the fine detail is to ensure versatility and flexibility of our taxonomy: in its current form it is applicable to both *inherently explainable predictive algorithms* as well as *standalone explainers*. One could imagine a taxonomy-based manual discussing how to interpret linear models based on their parameters, thereby highlighting caveats such as feature normalisation and independence assumptions. Such an elaborate structure also allows the users to quickly browse through the hierarchy, e.g., just inspecting the headings, without delving into details, thus making it more appealing and accessible.

Many requirements included in the taxonomy originate from diverse XAI literature – academic and on-line articles – and have proven to be of value in multiple instances, both theoretical and practical. The outer-level categorisation (dimensions) is role-driven, e.g., deployment, certification and users’ perception. When composing the list we were as comprehensive as possible (to avoid bias) and our intervention was limited to grouping together similar concepts presented under different names. Even so, we acknowledge that our taxonomy by no means should be treated as final and definitive. We plan to validate and revise it over time based on the feedback provided by its users (who create and deploy explainability solutions) and the XAI community.

Despite a well-defined list of requirements, exhaustively applying the taxonomy to any single explainability approach is a labour-intensive and time-consuming challenge. While some of these properties are purely *analytical*, others are *empirical*. We hence identify two approaches to validate them:

**quantitative** for properties that should be measured or can be precisely and definitely answered by assertion; and

**qualitative** for properties that should be defined in the context of a given explainability approach and either justified by a critical discussion (informal argument) or validated with user studies since they may lack a unique answer, be subjective or be difficult to measure.

This lack of a “correct” answer, however, should not be held against the taxonomy as even a qualitative discussion – of fuzzy properties that cannot be directly operationalised – is of considerable importance in advancing transparency of explainability approaches. Among other benefits, it encourages reporting on properties that are ambiguous and clarifying aspects of explainers that cannot be precisely measured, e.g., describing and justifying the chosen validation procedure allows the users to judge suitability of a given explainer for their individual use case.

## 2.5 Systematic Evaluation Approaches in AI

The recent surge in interpretability and explainability research in AI and ML may suggest that this is a new research topic, but in fact (human) explainability has been an active area of study for much longer in the humanities [106]. This observation encouraged Miller [106] to review XAI and IML approaches in the technical literature using insights from the social sciences, which shows how human-centred design can benefit AI explainability [107]. To date, a scattered landscape of explainability systems desiderata has been presented in a wide range of publications [56, 70, 77, 80, 81, 96, 127, 138, 167] that propose to evaluate explanations by defining their properties and interaction protocol. Despite many of these researchers converging towards a coherent list of requirements expected of explainability approaches, none of them collected and organised such a systematic list to serve as a guideline for other people interested in the topic.

Some studies [178] discuss a subset of requirements included in our taxonomy and support them with illustrative examples, but their main aim is to *familiarise* the readers with such concepts and not provide them with an evaluation framework. Other research avenues use a selection of these properties to *evaluate* a given explainability approach for a fixed task. For example, Kulesza et al. evaluate interactive explanatory visualisations for: music recommendations [80] with respect to explanation *fidelity* (*soundness* and *completeness*), and naïve Bayes classification of emails [81] with respect to *fidelity*, *interactiveness*, *parsimony* and *actionability*. Lakkaraju et al. [86] mathematically define some of the desiderata, e.g., *soundness* and *completeness*, to facilitate quantitative evaluation of the explainability method that they propose. Alternatively, organisations such as IEEE attempt to develop standards (either imposed or self-regulated) for transparency of autonomous systems [115].

Another, related branch of explainability research deals with user studies, which in this field are often considered the gold standard of validation. Doshi-Velez and Kim [34] come up with guidelines and best practices for evaluating effectiveness of explainers with user studies and synthetic verification, whereas others [115, 124, 167] consider validating explainability techniques by focusing on their audience. In particular, Tomsett et al. [167] suggests that effectiveness of explainers should be demonstrated separately for each stakeholder:

- system creators,

- system operators,
- executors making a decision on the basis of system outputs,
- decision subjects affected by an executor’s decision,
- data subjects whose personal data is used to train a system, and
- system examiners such as auditors or ombudsmen.

Similarly, IEEE [115] proposes to consider: users, safety certification agencies, accident investigators, lawyers or expert witnesses, and wider society. Nonetheless, some research argues that user studies cannot fully assess the effectiveness of an explanation due to a phenomenon called The Illusion of Explanatory Depth [132], or that their results can yield unjustified positive results (confirmation and outcome biases) as simply offering an explanation makes its recipients believe that the underlying event is more likely to be true than not [73].

A different approach towards clarifying explainability properties in ML and AI is self-reporting and certification. Techniques such as “data statements” [15], “data sheets for data sets” [47] and “nutrition labels for data sets” [60] can help to characterise a data set in a coherent way. Kelley et al. [67] argue for a similar concept (“nutrition labels for privacy”) to assess privacy of systems that handle personal (and sensitive) information. All of these methods revolve around recording details about the data themselves, e.g., the units of features, the data collection process and their intended purpose. Other researchers propose an analogous approach for predictive models: “model cards for model reporting” [108], “nutrition labels for rankings” [180] and “algorithmic impact assessment” forms [126]. Finally, Arnold et al. [11] suggest “fact sheets” for ML and AI services (also called “Supplier’s Declarations of Conformity”) to communicate their capabilities, constraints, biases and *transparency*, which are predominantly aimed at casual users since access barriers to these tools are relatively low.

## 2.6 In Search of the Explainer: Surrogates Desiderata

Our XAI taxonomy collates and discusses a list of *functional*, *operational* and *usability* characteristics of explainability techniques designated for predictive systems. It also examines *safety* (security, privacy and robustness) properties of explainers and explanations, in addition to reviewing their *validation* methods such as user studies and synthetic verification. For each individual desideratum, we show how it can be used to systematically characterise AI and ML explainability approaches, point out their caveats, and highlight disagreements and trade-offs between their theoretical algorithms and available implementations. The taxonomy was designed to be as comprehensive as possible, spanning both technical and social properties, thus serving diverse audiences and creating a basis for building various tools such as fact sheets, manuals, work sheets and check lists. Based on this flexibility, we propose that explainability approaches

are accompanied and assessed by means of applying our taxonomy, for example in a form of explainability *Fact Sheets*, three of which are included in Appendix A.

Having collected, organised and reviewed a wealth of XAI desiderata to build our taxonomy, we identified *post-hoc* and *model-agnostic* explainers as the most promising research direction given their universality and a potential for satisfying many of these diverse properties. Furthermore, juxtaposing our *usability* requirements with Miller’s [106] review of *contrastive* explanations highlights their appeal when employed as AI and ML explainers. Since suitability of an explainability system depends on a wide range of factors, including the intended audience and application domain, developing a *process* for building explainers that can be adapted to the requirements of a particular use case seems more beneficial than searching for a fixed algorithm. All of these observations led us towards *surrogate* explainers – approximating a black box with an inherently interpretable model in a selected data subspace – which we identified as the most promising candidate. As a starting point, we looked into the algorithmic design and implementation of LIME [129], which is one of the most prominent examples of surrogate explainers – in Appendix A.1 we review it based on our taxonomy. This investigation became the foundation of our follow-on research in which we modularise surrogates, study rules of their composition, identify common pitfalls and propose their improved design, all guided by our taxonomy.

## BLIMEY: MODULAR SURROGATE EXPLAINERS

Surrogate explainers play an important role in explainable artificial intelligence since they are post-hoc, model-agnostic and compatible with a variety of data types. They gained considerable popularity in recent years following publication of the Local Interpretable Model-agnostic Explanations (LIME) algorithm, which instantly became their main representative. While it is the most prominent surrogate explainer, a much broader family of these methods remains to be explored. Without understanding their inherent properties, such techniques may result in subpar explanations due to a fundamental mismatch between individual transparency requirements and properties of the employed explainer. To fill this gap, we propose a principled algorithmic framework and a meta-algorithm for composing a wide range of tailor-made local surrogate explainers of black-box models and their predictions in an effort to empower the community to “build LIME yourself” (bLIMEy).

In particular, we demonstrate how to decompose surrogates into independent and interoperable modules – interpretable feature representation, data sampling and explanation generation – and discuss the influence of these component choices on the functional capabilities of the resulting explainer. We then investigate individual properties and compatibility of these various building blocks to discover: **a)** fragility of occlusion-based interpretable representations of images with respect to their segmentation granularity and occlusion colour; **b)** information loss when explaining binary interpretable representations of tabular data with linear models; and **c)** evidence suggesting superiority of decision trees as a replacement for both the interpretable representation and explanation generation steps when dealing with tabular data. We support these findings with experimental results and a formal derivation presented in Appendix C. With such a high degree of customisability and numerous pitfalls, this chapter introduces a conceptual framework, outlines desiderata and provides guidelines for building bespoke surrogate explainers. On a practical

level, it mainly deals with tabular data; this analysis is complemented in Chapter 5 by a quantitative evaluation of tree-based surrogates applied to images (and by extension text). bLIMEy is accompanied by an open source implementation distributed within the FAT Forensics Python package (described in Appendix B), which includes a selection of algorithms for each component of surrogate explainers.

### 3.1 The Family of Surrogate Explainers

Achieving state-of-the-art predictive performance with AI techniques typically requires considerable time and effort to wrangle data, engineer informative features and fine-tune the machine learning algorithm of choice. To guide the development, deployment and maintenance of such systems, researchers and practitioners have established standardised, iterative, multi-stage processes for knowledge discovery, e.g., KDD [38], CRISP-DM [26, 103, 141] and BigData [6]. However, as Rudin [133] points out, complementary tasks – including explainability (see Section 1.5.1) – often lack such an operational framework. While in recent years IML and XAI research has yielded an abundance of promising algorithms, using them outside of a laboratory setting tends to be more difficult than portrayed by scientific publications. Similar to a machine learning pipeline focused on optimising for predictive accuracy, explainers of black-box models should be treated as *modular frameworks* requiring configuration and tuning to ensure meaningful explanations.

**Missing Link** Our XAI taxonomy introduced in Chapter 2 is a step in this direction, but it cannot replace a formal process for designing, deploying and maintaining explainers of data-driven systems. Given the relatively young age and ongoing development of explainability research, creating a generic *XAI process* appears beyond our reach at this stage; nonetheless, limiting its scope to a selected family of explainers is more promising. Such a framework can prove especially useful for methods that are:

**post-hoc** can be retrofitted into any predictive pipeline,

**model-agnostic** work with any black-box system, and

**data-universal** are compatible with diverse data types such as image, text and tabular

since they can be applied to an array of diverse black boxes. These properties, however, create a risk of considering such explainers as panaceas that work straight out of the box; yet in reality they may not always be suitable, leading to inconsistent performance and subpar explanations. We argue that decomposing and modularising such techniques will improve their customisability, functionality and overall quality, especially if accompanied by comprehensive design guidelines. Applying these composition principles and process to post-hoc and model-agnostic approaches, in particular, will address a major concern with fidelity of their explanations [133].

While the ultimate goal is to develop an XAI process that is agnostic of the selected explainer, this chapter forays into designing a *principled algorithmic framework* and a *meta-algorithm* for composing a wide range of bespoke local **surrogate explainers** of black-box models and their predictions. Such methods construct an interpretable “surrogate” model, e.g., a shallow decision tree, to mimic and explain the behaviour of (a part of) an independently trained black box. Explainers from this family are in general post-hoc, model-agnostic and data-universal, thus versatile and flexible, but also complex from an engineering standpoint. With many components to choose from and parameters to tune, supporting the user in making informed choices to build a custom surrogate explainer that is optimal for a specific task can considerably improve the quality of the resulting explanations [158] and promote a wider adoption of this set of techniques.

The aforementioned properties of surrogate explainers lower the technological barrier to their adoption, making them an important and highly impactful technique in XAI and IML. Surrogates have also the potential to become the go-to explainer by pairing their versatility with improved accessibility, reliability, accountability and fidelity, all of which stem from enhanced customisability formalised through a *surrogate XAI process*. By empowering the users to take advantage of this modularity, they can compose bespoke configurations tailored to their individual use cases, building the best possible explainer that the surrogate family has to offer. This process can be further improved by educating the creators about the available component choices, their properties, advantages, caveats, inter-compatibility and influence on the produced explanations. This intuition is in line with the proverbial “no free lunch” theorem – described in Section 1.1.1 with respect to explainability – which, to reiterate, implies that a single explainer cannot outperform all the other approaches across the board. Finally, addressing all of these challenges by formalising a *surrogate XAI process* has the potential of becoming a trailblazer for creating such frameworks for other explainer families (possibly based on our taxonomy), and ultimately generalise to XAI and IML as a whole.

**LIME** The idea of using surrogates to explain a black-box model can be traced back to Craven and Shavlik [28], who approximated a neural network with a decision tree. It was recently re-introduced and popularised by Ribeiro et al. [129] through Local Interpretable Model-agnostic Explanations (LIME), which extend surrogates with the concept of *interpretable representations* and gear them towards black-box predictions. While appealing, their sometimes inconsistent behaviour, instability and low fidelity are a well documented problem [84, 87, 89, 133, 183]. Notably, this body of research is predominantly concerned with such undesired and detrimental artefacts pertaining to explanations produced with LIME’s official open source implementation<sup>1</sup>, thus largely limiting their applicability to this particular software. Such insights are very informative, however they often do not pinpoint the root causes of the observed issues, with individual papers [89] addressing some of these challenges with (algorithmic) stopgap measures.

---

<sup>1</sup><https://github.com/marcotcr/lime>

For example, Laugel et al. [89] attempted to “fix” LIME for tabular data by replacing its data augmentation method with an explicitly local sampler. Nonetheless, their experiments used LIME with disabled discretisation – a step responsible for generating the interpretable representation of data. This modification unintentionally compromised the integrity of the algorithm, rendering the two methods incomparable and the reported improvements inapplicable to more general cases beyond the specific ones presented in their research. This study is not an isolated case; the prevailing research theme appears to investigate explainers as *monolithic* tools by probing them with doctored inputs in order to identify their weaknesses [84, 183], or applying quick-fixes to their individual algorithmic components without considering the broader implications of such actions [89]. (A more detailed discussion of relevant work is presented in Section 3.5.) While this type of (mostly empirical) reports may be of some use, it falls short of a fundamental analysis of these techniques, thus overlooking their possible modularity and an opportunity to learn how each individual component operates on its own and in relation to each other, both of which affect the final “product”. These observations inspired us to look for solutions beyond modifying the original LIME algorithm, which can be applied to the entire family of surrogate explainers.

**bLIMEy** LIME is a very popular technique for explaining predictions of black-box machine learning models and a prototypical IML surrogate, but it is just one possible realisation of the highly modular *family of surrogate explainers*. Nonetheless, taking advantage of this observation in practice is far from trivial on both conceptual and functional levels due to LIME’s design: it is introduced as a self-contained algorithm and modifying its default behaviour (beyond simple parameterisation) often requires tinkering with its source code. We address these challenges by establishing a design process and a complementary meta-algorithm that formalise the task of building bespoke surrogate explainers, such as LIME, within a unified algorithmic framework. Our approach, which we call **bLIMEy (build LIME yourself)**, takes advantage of the inherent modularity of surrogates, enabling systematic development and evaluation of techniques from their extensive family. In particular, it supports creation of a suite of customisable surrogate explainers by employing a range of diverse algorithms for each of their modules.

We present bLIMEy as a *meta-algorithm* for surrogate explainability of tabular, text and image data, improving upon LIME’s fixed structure – a generalisation outlined in Section 3.2. It can explain a black box in its entirety by mimicking its behaviour with a simpler, human-comprehensible model; or instead explain a user-selected prediction by fitting the surrogate in its neighbourhood. bLIMEy consists of three building blocks: *interpretable data representation*, *data sampling* and *explanation generation*, each one offering a wide selection of algorithms and parameterisation possibilities. Since the varying capabilities and restrictions of these components greatly influence the resulting surrogate explainer, we argue that each of them should be accompanied by a critical discussion and operationalisation suggestions. The most important choices and their properties are examined in Section 3.3, which can be treated as guidance for researchers and practitioners to help them navigate around frequent issues, the most prominent

of which are summarised below.

**Interpretable Data Representation** The determinism of this transformation plays an important role in preventing the interpretable representation from causing the resulting explanations to be unstable. This property is also crucial in preserving the local faithfulness of interpretable representations, especially for tabular data. Its parameterisation tends to bias explanations since it determines their type, content and quality – we demonstrate this phenomenon experimentally for image data in Appendix C.1.

**Data Sampling** Local and class-aware sampling tends to be superior for tabular data, whereas exhaustive binary sampling has advantages for image and text data.

**Explanation Generation** The choice of the surrogate model strongly influences the type, content and meaning of the output explanations. Logical models, such as decision trees, exhibit many properties that are desirable in this setting, which we demonstrate experimentally for tabular data in Appendix C.2.

While individual properties of each surrogate building block are important, these modules also need to be compatible with each other to ensure quality and faithfulness of the resulting explanations – we examine this relation in Section 3.4. In particular, we analyse a popular pairing of an interpretable data representation with a surrogate model (used to generate explanations) for tabular data, which is employed by LIME: quartile-based discretisation and binarisation of numerical features explained with coefficients of a linear model. An analytical derivation of such explanations for Ordinary Least Squares (OLS) – outlined in Appendix C.3 – demonstrates that this type of a surrogate has inherent limits to its explanatory power. We then show how to mitigate this issue by using *surrogate decision trees* instead. Such a configuration combines these two independent steps in a way that solves many problems pertinent to the pairing of a discretisation-based interpretable representation with a linear surrogate.

Since some of our discoveries are based on rather specific theoretical findings, they may not be applicable to individual use cases, thereby limiting their utility. To address this problem, we conclude our analysis of surrogate explainers by introducing a range of diverse quantitative measures of explanation faithfulness. Each one is suitable for a different type of a high-level evaluation objective such as fidelity of the black-box decision boundary approximation (model-driven) or the quality of predictive mimicry in the neighbourhood of the explained instance (data-driven). We conclude our work on modular surrogates in Section 3.6 with a summary of our key findings and a motivation for further investigation of decision tree explainability (Chapter 4) spurred by the aforementioned evidence of their wide-reaching advantages across the family of surrogate explainers.

Since bLIMEy decomposes surrogates into independent functional building blocks, we publish their open source implementation to complement our conceptual meta-algorithm. For each of them, we coded a choice of algorithms in Python and released them under the BSD 3-Clause

licence – which allows commercial use – within the FAT Forensics<sup>2</sup> package [159, 160], as described in Appendix B. Our implementation is accompanied by a collection of on-line “how-to” guides<sup>3</sup> and a hands-on tutorial<sup>4</sup> [162] explaining how to compose custom surrogates and discussing pros and cons of individual component choices. It is also capable of recreating the LIME algorithm for tabular, image and text data in a way that mitigates most of the issues reported in the literature [84, 87, 89, 133, 183]. Lastly, it can be used to reproduce all of the experiments and results presented in this thesis.

## 3.2 From LIME to bLIMEy: Beyond Linear Surrogates

The versatility of LIME is impressive, especially with respect to compatible types of data – a significant improvement over preceding explainers that were mostly limited to a single domain. This generalisation leap is enabled by the introduction of *interpretable data representations* [129], which make the explanatory process agnostic of data and their unique characteristics. This proxy between data and explanations greatly extends the capabilities of surrogate explainers by making them applicable to images and text in addition to tabular data, while maintaining a familiar explanation appearance, thereby creating a uniform user experience. This flexibility (disguised in apparent simplicity) inspired us to investigate the inner workings of surrogate explainers and push their capabilities to their limits. To this end, we first take a closer look at the design principles and roles of interpretable representations. Having a functional understanding of this concept allows us to review individual steps of the LIME algorithm through the lens of the surrogate framework, hence identify distinct and self-contained operations of this particular method. These insights contribute to the foundation of bLIMEy, which generalises LIME to a customisable meta-algorithm that serves as a workflow for building a wide range of surrogate explainers.

### 3.2.1 Benefits of Interpretable Representations

Interpretable Representations (IRs) are the foundation of many explainability methods for black-box machine learning [45, 100, 129], particularly so for surrogate explainers. They facilitate translating the “language” of ML models – low-level data representations required for good predictive performance, such as raw feature values and their complex embeddings – into high-level concepts that are intelligible and relatable for humans. Therefore, IRs establish an *interface* between a computer-readable encoding of a phenomenon (collected data) and cognitively digestible chunks of information, creating a suitable medium for conveying explanations. Notably, the explanation type and its (perceived) complexity are directly controlled by the underlying

---

<sup>2</sup><https://fat-forensics.org/>

<sup>3</sup>For example, [https://fat-forensics.org/how\\_to/transparency/tabular-surrogates.html](https://fat-forensics.org/how_to/transparency/tabular-surrogates.html).

<sup>4</sup><https://events.fat-forensics.org/>

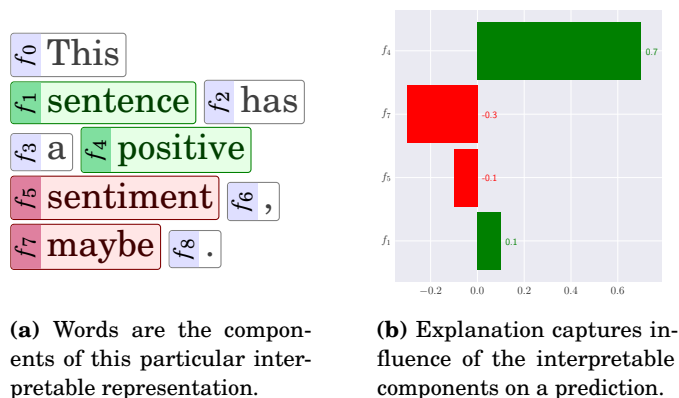
interpretable representation, making IR-based explainers highly flexible, versatile and appealing. By personalising an interpretable representation we can adjust the content of the resulting explanations, thus allowing its creators to target a selected *audience* and use case.

In most cases, the function transforming data from their original domain into an interpretable representation is defined by the user and built into the surrogate. Uniquely for tabular data, however, it can be learnt as part of the *explanation generation* step depending on the choice of the surrogate model – see Section 3.3.2 for more details. An IR of images, for example, can be created with a super-pixel segmentation, i.e., partitioning images into non-overlapping *segments*, each one representing an object of interest or pieces thereof. Similarly, text can be split into *tokens* denoting individual words, their stems or collections of words that are not necessarily adjacent. Tabular data containing numerical features can be discretised to capture meaningful patterns, e.g., people in different age groups. Notably, the first two types of representation change facilitate explainability of sensory data [129]. The operationalisation of IRs also vary across different data types – tabular, image and text – but their machine-readable format is usually consistent. The most common approach is to use a binary vector to encode presence (*fact* denoted by 1) or absence (*foil* denoted by 0) of certain human-understandable concepts captured by an interpretable representation generated for a selected data point. Importantly, choosing the foil may not always be straightforward or even feasible in certain domains, requiring a problem-specific proxy.

### **Interpretable Representations for Text and Image Data**

The interpretable representations of image and text data come naturally, are intuitive and share many properties. Images are partitioned into non-overlapping segments called super-pixels, which are then represented in the interpretable binary space as either present or absent. Similarly, text is split into tokens that can encode individual words, their stems or collections of words, the presence or absence of which is expressed in the IR. These two interpretable representations are relatively easy to generate automatically and, when configured correctly, capture (computationally) meaningful concepts. Notably, the high dimensionality of raw data does not impact comprehensibility, as is the case with tabular data where we are generally confined to three dimensions given the spatio-temporal limitations inherent to human visual system. Moreover, dimensionality reduction for images and text is unnecessary or even harmful: removing super-pixels from images is an ill-defined procedure and would result in “holes”, whereas for text it can be used to discard stop words and punctuation, but such pruning is often incorporated into the preceding tokenisation step.

**Text** The interpretable domain based on presence and absence of tokens in text is very natural and appealing to humans. Individual words and groups thereof encode understandable concepts, the absence of which may alter the meaning of a sentence, reflecting how humans comprehend text. A naïve IR can represent text as a bag of words – where each word becomes a token – thereby

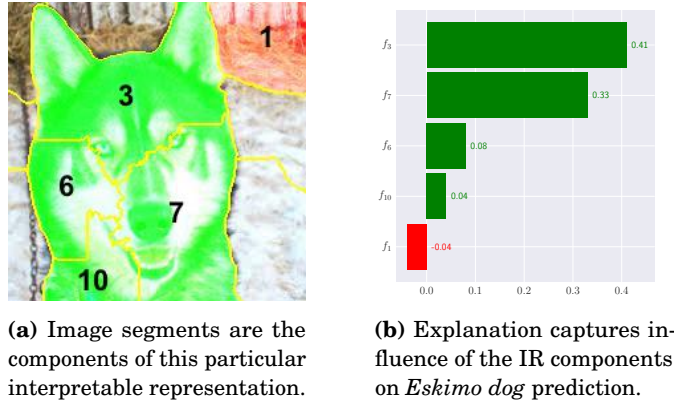


**Figure 3.1:** Example of an influence-based explanation of text with a *bag-of-words* interpretable representation. Panel (a) illustrates a sentence whose (positive) *sentiment* is being decided by a black-box model. The colouring of each word in Panel (a) conveys its influence on the prediction, with Panel (b) depicting the corresponding magnitudes.

forfeiting the influence of word ordering and the information carried by their co-appearance. We can easily improve upon that and capture the dependencies between words by including  $n$ -gram groupings. Applying other pre-processing steps, e.g., extracting word stems, can also be beneficial for the human-comprehensibility of such interpretable representations. Machine processing of text is a well-established research field [102], providing plenty of inspiration for the design of appealing IRs.

Once text is pre-processed and *tokenised*, it is **deterministically** transformed into the binary interpretable representation. To this end, a sentence is encoded as a Boolean vector of length equal to the number of unique tokens in the IR, where 1 indicates presence of a given token and 0 its absence; the original sentence is therefore represented by an all-1 vector. By flipping some components of this vector to 0, we effectively remove tokens from the underlying sentence and create its variations. Notably, the high dimensionality of this IR does not undermine the comprehensibility of the resulting explanations since altered text cannot have more tokens than the original sentence. Explanations based on token influence can be overlaid on top of text by highlighting each token with a different shade of green (positive) or red (negative) given their respective impact on the explained class – see Figure 3.1 for an example.

**Images** The interpretable representation of image data operates similarly to text IRs – see Figure 3.2. Images are algorithmically *segmented* into super-pixels, often using edge-based methods [129] such as *quick shift* [171], but the resulting partition may not convey (cognitively) meaningful concepts from a human perspective. *Semantic segmentation* or delegating this task to the user usually yields better results [150, 151]. Next, the segments are represented as a binary vector encoding presence (1) or absence (0) of information in each super-pixel; an all-1 vector corresponds to the original image. However, removing a super-pixel from an image when setting



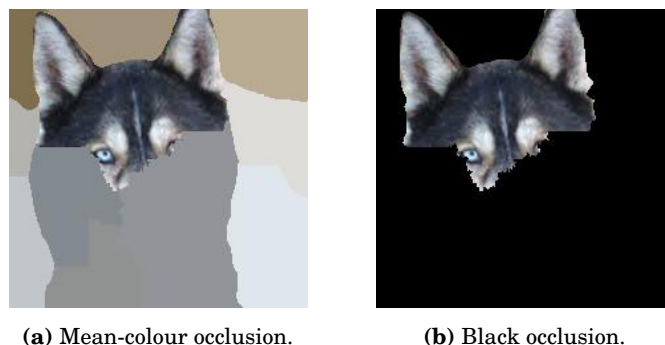
**Figure 3.2:** Example of an influence-based explanation of image data with the interpretable representation built upon *segmentation*. Panel (a) illustrates an image that is being classified by a black-box model. The colouring of each super-pixel in Panel (a) conveys its influence on *Eskimo dog* prediction, with Panel (b) depicting the corresponding magnitudes.

one of the interpretable components to 0 is an ill-defined procedure. The most appealing and semantically-meaningful solution would be to “delete” the content of a segment by occluding it with another object, akin to Benchmarking Attribution Methods (BAM [181]), or retouching it in a context-aware manner, e.g., with what is anticipated in the background, thus preserving the colour continuity of the image. Both of these approaches are intuitive, but they are difficult to automate and scale since they are mostly limited to image partitions where each super-pixel represents a self-contained and semantically-coherent object.

Instead, a computationally-feasible proxy is commonly employed to hide the information carried by the super-pixels: segments are occluded with a solid colour. For example, LIME uses the mean colour of each segment to mask its content [129] – see Figure 3.3a. Alternatively, a single colour can be applied to all of the super-pixels that are assigned 0 in the interpretable domain to cover them up, for example with the black colour as pictured in Figure 3.3b. In addition to choosing the occlusion *colour*, we can also adjust the *granularity* of the segmentation, effectively increasing or decreasing the size of super-pixels. Both of these parameters influence the resulting explanations to varying degrees; for example, two different colouring strategies may yield disparate black-box predictions (probabilities) of the same class for a partially occluded image as shown in Figure 3.3. We discuss these phenomena in more detail in Section 3.3.2, supporting our findings with experimental results (presented in full in Appendix C.1), all of which lead to a collection of recommendations for designing sound interpretable representations of images.

### Tabular Interpretable Representations

In contrast to raw pixel values and word embeddings, tabular data do not require an interpretable representation to be explainable since their features are often human-comprehensible. However,



**Figure 3.3:** Image occlusion strategy influences the resulting explanations (see Appendix C.1). The picture shown in Figure 3.2a is classified by a black box as *Eskimo dog* with 83% probability. Mean-colour occlusion of all the segments but one (a) results in 77% and black occlusion (b) in 9% probability of the same class, showing that the former approach cannot effectively remove information from this particular image.

if the explanation is to answer a specific question – as was the case for images and text – using an IR may be necessary. Continuing with the *interpretable sensitivity analysis* setting (i.e., computing influence of selected factors on black-box predictions) for tabular data, we are interested in how presence and absence of certain *binary* concepts pertaining to the explained instance affects its prediction. One approach is to treat the specific feature values of the explained data point as concepts: if an attribute value of an instance is identical to the value of the same feature for the explained instance, the concept is *present* (1), otherwise it is *absent* (0). While appealing for categorical attributes, considering each and every unique value of a numerical feature is counter-intuitive given their inherent continuity. Moreover, doing so may not reflect the corresponding human thought process, e.g., “high sugar content” in contrast to “70g of sugar per 100g of a product”, with both 0g and 100g in the latter case encoded as an *absent* concept in the underlying IR.

Building up on this observation, a natural extension of such an IR is to consider intervals of numerical features as interpretable concepts: if an attribute value of a data point is within the same range as the value of the same feature for the explained instance, the concept is *present* (1), otherwise it is *absent* (0). To this end, tabular data with numerical attributes need to be *discretised*, creating a *hyper-rectangle partition* of the space. The binary interpretable representation is unique to each hyper-rectangle, hence one of the partitions has to be chosen as the explanation target. The binary IR of a selected data point is then computed by comparing the hyper-rectangle it belongs to along every feature (i.e., its discrete encoding) with the equivalent representation of the explained instance. This procedure results in a binary on/off vector that for each discrete dimension (captured by the underlying IR) indicates whether the chosen data point lies in the same partition as the explained hyper-rectangle or not.

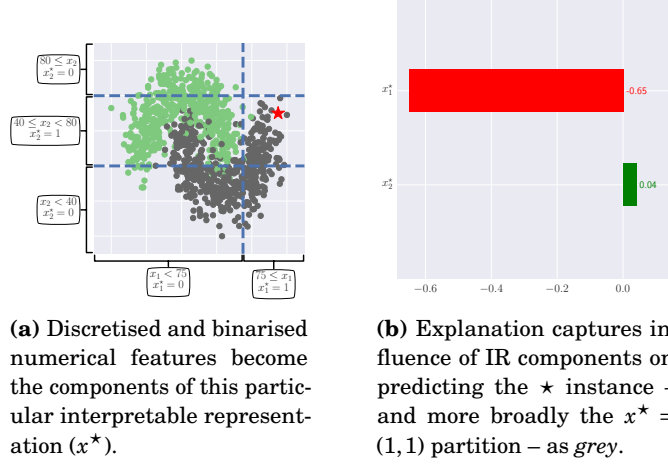
Therefore, IRs of tabular data are generated by preserving categorical features and discretising numerical attributes into “categorical” bins, for example,  $x_2 < 5$ ,  $5 \leq x_2 < 7$  and  $7 \leq x_2$ . Next,

the binary on/off representation is computed based on the data point selected to be explained, which facilitates expressing influence of each interpretable concept on the black-box prediction of this instance. Since a binary representation allows to encode only two events for each attribute, it commonly indicates a feature value belonging to the same (1) or different (0) numerical bin as the explained data point. For example, if the second feature  $x_2$  of the explained instance  $\hat{x}$  is  $\hat{x}_2 = 6.5$ , based on the aforementioned bin boundaries any data point  $x$  whose second attribute is within the  $5 \leq x_2 < 7$  range is assigned  $\hat{x}_2 = 1$  in the underlying binary IR, and 0 otherwise. Notably, if any feature in the discretised representation has more than two unique values, this procedure will “merge” some of the hyper-rectangles – see the  $(x_1^*, x_2^*)$  coordinates in Figure 3.4a, where two (out of three) partitions of the  $x_2$  attribute (y-axis) are assigned  $x_2^* = 0$  in the binary IR.

By default, LIME uses *quartile*-based discretisation followed by the aforementioned binarisation procedure. The former step is done once based on a data set chosen for training the LIME explainer, thus building a reusable binning of numerical features that becomes the foundation of all subsequent binary IRs for the problem at hand. While the discretisation is shared between such interpretable representations, each explained data point receives an individual IR that is determined by the hyper-rectangle it belongs to. However, the final binary encodings lose information since all of the instances residing in the same partition of the feature space receive an identical interpretable representation (assuming that the modelled data set has discretised numerical features). When paired with a surrogate linear model, this IR allows us to investigate how each attribute value of the chosen data point being within its respective partition (the concept “switched on”) *influences* predicting a selected class. (Such insights reflect the behaviour of the underlying black box, which is captured by its predictions across different hyper-rectangles.) Therefore, the non-uniqueness of binary IRs within a single data set (given presence of discretised numerical features) also applies to the resulting explanations, which are specific to a hyper-rectangle (likely containing multiple instances) rather than an individual data point – see Figure 3.4 for an example (note that the discretisation presented therein is not based on quartiles). In Section 3.3.2, we investigate other properties of tabular interpretable representations obtained through a discretisation step followed by a binarisation procedure; we propose to replace them with a tree-based partition of the feature space, which achieves better faithfulness as shown by experimental results outlined in Appendix C.2.

### 3.2.2 LIME: A Surrogate Explainer of Black-box Predictions

On a conceptual level, LIME strives for *low complexity* of the surrogate model and *high fidelity* of the resulting explanations with respect to the black box; this is achieved by optimising the objective function  $\mathcal{O}$  given in Equation 3.1. Complexity  $\Omega$ , in the case of linear models, is given by the number of non-zero (or significantly larger than zero) coefficients of the surrogate model  $g \in \mathcal{G}$ , where  $\mathcal{G}$  is the set of all the possible (sparse linear) surrogate models. High fidelity is attained by *minimising* the loss  $\mathcal{L}$  (defined in Equation 3.2) calculated between the outputs of the black box



**Figure 3.4:** Example of an influence-based explanation of tabular data with the interpretable representation built upon *discretisation* and *binarisation*. Panel (a) illustrates an instance (red  $\star$ ) that is being predicted by a black-box model. The dashed blue lines mark feature partitions; grey and green denote two predicted classes; and  $x^*$  is the binary IR created for the  $\star$  data point. Panel (b) depicts the magnitude of the influence that  $x_1^* : 75 \leq x_1$  and  $x_2^* : 40 \leq x_2 < 80$  have on predicting the *grey* class for the  $\star$  instance (as well as any other data point located within the same hyper-rectangle).

$f$  and the surrogate model  $g$  – it measures how well the latter approximates the former.<sup>5</sup> In our notation,  $x \in \mathcal{X}$  is a data point described in the original domain, which can be transformed into an interpretable representation  $x' \in \mathcal{X}'$  with a user-defined mapping  $IR : \mathcal{X} \rightarrow \mathcal{X}'$  (the inverse transformation is denoted by  $IR^{-1}$  if it exists). The explanation locality is enforced by weighting the individual loss incurred by each instance (drawn from the validation set) with a kernelised  $k : \mathbb{R} \rightarrow \mathbb{R}$  distance  $L : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$  computed in the interpretable domain  $\mathcal{X}'$  in relation to the explained data point  $x' = IR(x)$ . More details about this optimisation procedure can be found in the LIME paper [129].

$$\mathcal{O}(\mathcal{G}; f) = \argmin_{g \in \mathcal{G}} \underbrace{\Omega(g)}_{\text{complexity}} + \underbrace{\mathcal{L}(f, g)}_{\text{fidelity}} \quad (3.1)$$

$$\mathcal{L}(f, g; x, X') = \sum_{x' \in X'} \underbrace{k(L(IR(x), x'))}_{\text{weighting factor}} \times \underbrace{(f_c(IR^{-1}(x')) - g(x'))^2}_{\text{individual loss}} \quad (3.2)$$

Here, however, we are interested in the algorithmic approach to this optimisation procedure, seeking to obtain enough insights to build modular surrogate explainers in practice. LIME assumes that the explained model is a probabilistic classifier (or regressor) and fits a sparse linear regression (surrogate) to the interpretable representation of data sampled in the neighbourhood of the instance selected to be explained, with every sample weighted by its kernelised distance to

<sup>5</sup>The subscript  $c$  in  $f_c$  – used in Equation 3.2 – indicates the probability predicted for a selected class  $c \in C$ .

the explained data point. The target of this regression are probabilities of a user-selected class predicted by the black box for the augmented data. In this setting<sup>6</sup>, LIME explains the specified data point with respect to the chosen class – usually the one assigned to it by the black box – in terms of interpretable feature influence, which quantifies the positive or negative effect of each concept being switched “on” as shown by Figures 3.1, 3.2 and 3.4. Therefore, for a data point  $\hat{x} \in \mathcal{X}$  selected to be explained and a given *probabilistic* black-box model  $f : \mathcal{X} \rightarrow [0, 1]^{|C|}$  – where  $C$  is the set of modelled classes and  $|C|$  is their number – the vanilla LIME algorithm proceeds as follows.

1. Determine the human-interpretable representation  $\hat{x}' \in \mathcal{X}'$  of the data point  $\hat{x}$  chosen to be explained, where  $\mathcal{X}'$  denotes the interpretable domain and  $IR(\hat{x}) = \hat{x}'$ .
2. Sample data  $X'$  from the interpretable domain  $\mathcal{X}'$  in the neighbourhood of  $\hat{x}'$ .
  - For image and text data, this is done by uniformly replacing 1s in  $\hat{x}'$  with values from the  $\{0, 1\}$  set to generate new data points in the “neighbourhood” of  $\hat{x}$ . This step produces variations of the image or sentence being explained, e.g., by randomly occluding segments in the image or removing tokens (words) from the sentence.
  - For tabular data, the sampling is performed on the discretised but *not* binarised representation to ensure that each instance is assigned to a unique hyper-rectangle. The data sample is then transformed into the binary interpretable representation  $\mathcal{X}'$ . (The need for sampling from this intermediate representation is explained in more details in Section 3.3.3.)
3. Map the sampled data  $X'$  from the interpretable representation  $\mathcal{X}'$  back to their original domain  $\mathcal{X}$  using the inverse function  $IR^{-1}$ . (See the discussion of the *determinism* of this transformation procedure and operationalisation of the  $IR^{-1}$  function in Section 3.3.2 for more details.) Expressing the sampled data in the original feature space  $\mathcal{X}$  is required to predict their probabilities for a selected class  $c \in C$  using the black-box model  $f$ . Usually,  $c$  is chosen to be the class assigned to the explained data point  $\hat{x}$  by the black-box model  $f$ , i.e.,  $c = \arg\max f(\hat{x})$ , however the user can selected a different class. The probability of class  $c$  predicted by the black box  $f$  for an instance  $x$  is denoted with  $f_c(x)$ .
4. Calculate the distances between the sampled data and the explained instance in the binary interpretable representation  $\mathcal{X}'$ , e.g., using the Manhattan, Euclidean or cosine distance. Next, kernelise these distances – e.g., using the exponential kernel – to transform them into proximity scores. These similarity measurements are then used to weight the sampled data when training the local surrogate model, thus reinforcing locality of the explanation.

<sup>6</sup>More details about this algorithm can be found in the official LIME implementation linked in Footnote 1.

5. Use a dimensionality reduction technique, e.g., K-LASSO [36], to limit the number of the interpretable representation components that are used to compose the explanation (i.e., train the local surrogate model). This step is especially useful for high-dimensional tabular data, where it decreases the explanation complexity by only considering the most prominent interpretable concepts – recall that the original domain  $\mathcal{X}$  and the interpretable representation  $\mathcal{X}'$  have the same dimensionality. For image and text data, however, the feature selection step is omitted as it would result in pictures or sentences with missing parts, which is undesirable in general. Notably, the high dimensionality of these two data types is not a problem as the explanation is inherently tied to the selected instance, which had to be intelligible in the first place. Figures 3.1 and 3.2, depicting a sentence and an image explanation respectively, are examples of this dependency.
6. Finally, fit a sparse linear regression to the subset of interpretable components selected in the previous step, weighting each sampled instance according to its kernelised distance from the explained data point. The target of this surrogate model are the black-box predictions calculated in step 3, i.e., probabilities of the previously selected class  $c$  computed with the black-box model for the sampled data. The coefficients of this linear regression (weights of the interpretable features) are then used to express the (positive or negative) influence of each human-comprehensible concept modelled by the surrogate.

In summary, LIME can be understood as a *sensitivity analysis* tool that operates on the interpretable domain  $\mathcal{X}'$ , judging the influence of presence and absence of human-intelligible concepts on a given class in the vicinity of the explained instance based on the behaviour of the underlying black-box model (i.e., its predictions). For image and text data, such explanations can be visualised either by merging them with the explained instance or as a self-contained bar plot – see Figures 3.1 and 3.2. Tabular data, on the other hand, can only be explained through a bar plot (Figure 3.4), with the exception of trivial cases where the data set has just two features (even such toy examples may be difficult to interpret for untrained explainees). This disparity in the presentation of explanations is mainly the result of a fundamental difference between operationalisation of sensory and tabular IRs: the former does not have to “simplify” the perceived complexity of the data since they are intelligible in their raw form, whereas the latter deals with a possibly large quantity of factors that need to be considered individually. While tabular interpretable representations are not necessary for explaining this type of data, they determine the *meaning* of explained concepts, which changes drastically if IRs are abandoned. Therefore, such an alteration requires additional processing steps to ensure trustworthiness of the resulting explanations, e.g., data features must be normalised to the same range for the coefficients of the surrogate linear model to be directly comparable.

### 3.2.3 bLIMEy: A Meta-algorithm for Building Modular Surrogate Explainers

The bLIMEy meta-algorithm decomposes surrogate explainers of black-box models and their (individual) predictions into three distinct steps:

- creation of an interpretable data representation;
- data sampling (augmentation); and
- explanation generation.

Each component is operationally independent, making them technologically cross-compatible regardless of their implementation details. Nonetheless, certain types of these modules may not be best suited for one another given that their assumptions, caveats and requirements may diverge on a conceptual level, thus resulting in subpar surrogates despite functionally sound building blocks. This duality is an important aspect of composing bespoke surrogate explainers with bLIMEy and we discuss it in detail throughout the remainder of this chapter.

**Interpretable Data Representation** This surrogate building block has already been introduced in Section 3.2.1. It transforms the low-level data representation necessary for good predictive performance (the original feature space or its embedding) into high-level, human-intelligible concepts (the interpretable representation) used to convey the explanations. Some surrogate explainers, LIME in particular, require this process to be *reversible*, and ideally *deterministic*. Interpretable domains tend to be *binary spaces* encoding presence and absence of human-comprehensible characteristics found in the data. This step is optional for tabular data, but necessary for images and text. Nonetheless, an interpretable representation may still be desired for tabular data since it determines the type of the resulting explanations as well as their content and cognitive complexity, thus allowing to target a particular audience and use case. It also implicitly defines the explanation scope: an interpretable representation can be specific to an individual instance, e.g., segmentation of an image, or it can be universally applicable to all (or a selection of) instances in a data set, e.g., logical rules for tabular data [45].

**Data Sampling** This module is responsible for generating data in the neighbourhood of the instance selected by the user to be explained. This set of data points captures the behaviour of the black-box model in the vicinity of the explained instance, therefore it determines the scope and coverage of the explanation. When explaining an individual prediction, data should be sampled in its direct neighbourhood. Alternatively, the sampling region can be expanded to cover a feature subspace spanning instances considered similar by the user, thus generalising the explanation from a prediction to a cohort. It is also possible to explain the entire black box by simplifying its decisive process with a (global) surrogate mimicking its behaviour, which requires training it on a sample that represents the whole data distribution.

For images and text, sampling must be performed in the interpretable domain since doing so in the raw feature space is either ill-defined, e.g., sampling individual pixel values for an image, or lacks meaningful correspondence to human-intelligible concepts. Tabular data, however, can be sampled in either of the domains, which provides a greater control over the explanation scope; when sampling in the original domain, this procedure explicitly defines the region from which data points are drawn. Regardless of the domain, the sampling procedure does not necessarily have to be random, especially if it is executed for a low-dimensional *binary* interpretable representation, in which case generating a complete set of instances is a viable alternative. The last step of this building block is predicting the data sample with the black-box model being explained to capture its behaviour in the decision space covered by these instances. If the data were sampled in the interpretable domain, they first need to be *converted* back to the original feature space using the inverse of the interpretable representation transformation function.

**Explanation Generation** The final building block of a surrogate explainer is an inherently transparent model, which is trained on the interpretable representation (if used) of the sampled data and a selected class of their black-box predictions. The built-in transparency mechanism of the employed surrogate model provides insights into the behaviour of our black box within the space encompassed by the data sample and expressed in terms of human-understandable concepts. In particular, the type of surrogate model determines the *category* of our explanations and the *medium* through which they can be delivered to the explainee. As we have seen with LIME, a binary interpretable representation encoding presence and absence of human-intelligible concepts in data paired with a surrogate linear model produces explanations based on influence of these concepts on predicting a specific class for a selected instance. For image and text data, these can be *visualised* in both the original data domain and as a self-contained bar plot, however tabular data only support the latter explanation type. A decision tree surrogate, on the other hand, can provide a diverse range of explanations – outlined in Chapter 4 – which, among others, include *textualisation* of counterfactual statements [169]. This building block can also alter the explanation characteristics by pre-processing the data used to train the surrogate model, e.g., explanation sparsity may be enforced by applying a dimensionality reduction technique. Furthermore, this step provides an opportunity to fine-tune the explanation scope by focusing the surrogate learning process on a selected subset (region) of the data sample. To this end, the data can be *weighted* according to the explainee’s interests when training the surrogate, for example, using similarity scores computed with kernelised distance between the explained instance and the sampled data (*in either representation*).

In summary, bLIMEy is a modular meta-algorithm built upon three components: interpretable representation, data sampling and explanation generation. Our framework – captured by Algorithm 3.1 – enables creation of bespoke surrogate explainers by introducing a sound engineering process that addresses multiple challenges arising for both individual building blocks and their

---

**Algorithm 3.1:** bLIMEy meta-algorithm. Sampling (Step 2) and weighting (Step 5) are done in the interpretable domain  $\mathcal{X}'$ . The order of Steps 5 and 6 – which are *optional* – can be reversed.

---

**Data:** • instance to be explained  $\hat{x}$  • class to be explained  $c$  • black-box model  $f$

**Result:** explanation  $e$

---

```

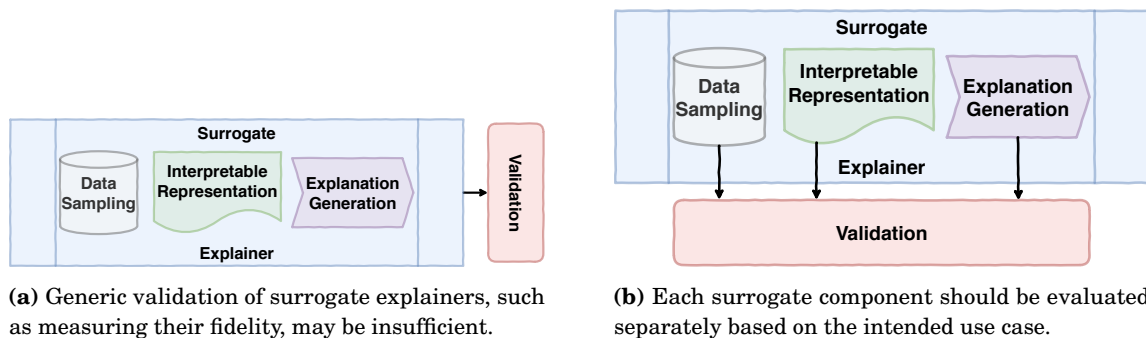
/* Interpretable Data Representation */
1  $\hat{x}' \leftarrow IR(\hat{x});$ 
/* Data Sampling */
2  $X' \leftarrow \text{sample\_data}(\hat{x}', \text{scope} = \{\text{local}, \text{cohort}, \text{global}\});$  /* Performed in  $\mathcal{X}'$  */
3  $X \leftarrow IR^{-1}(X');$ 
4  $y \leftarrow f_c(X);$ 
/* Explanation Generation */
5  $w \leftarrow k(L(\hat{x}', X'));$  /* Get weights via kernelised distance */
6  $\bar{X}' \leftarrow \text{reduce\_dimensionality}(X');$ 
7  $g \leftarrow \text{fit\_surrogate}(\bar{X}', y, \text{sample\_weight} = w);$ 
8  $e \leftarrow \text{extract\_explanation}(g);$ 

```

---

custom compositions. To this end, we identify roles of each module and discuss its influence on the resulting explanations, in both cases looking for the most universal techniques and best practices when choosing individual components and welding them together. Furthermore, we show how to generate insights of varying complexity by adjusting the content of explanations and selecting their appropriate type to present the explainee with useful artefacts such as interpretable feature influence and counterfactuals, among many others.

bLIMEy neither directly improves upon LIME nor competes with it since our method is a conceptual approach to building surrogates and not an explainer per se. However, it provides guidance for constructing a local surrogate explainer of black-box predictions that is based on a binary interpretable representation and a sparse linear model, which potentially improves on vanilla LIME and addresses its shortcomings. In particular, we propose to use an explicitly local sampler applied to the original domain of tabular data, and to generate a complete sample in the interpretable representation of sensory data. We also show how to reduce instability of LIME by replacing the quartile-based IR of tabular data with a feature space partition extracted from a decision tree, and changing the colouring strategy of the occlusion-based interpretable representation of images from mean colour to a single colour, e.g., black. We discuss these alternative building blocks individually in the following section, after which (Section 3.4) we inspect their interactions and influence on the resulting explainers in addition to reviewing methods for evaluating quality of surrogates.



**Figure 3.5:** Validating a surrogate explainer as a whole may be insufficient (a) given its diverse building blocks and their parameterisation. Instead, each individual component – data sampling, interpretable representation and explanation generation – should be evaluated on its own (b).

### 3.3 bLIMEy Modules

When composing a surrogate with bLIMEy, every module choice and parameterisation may limit the overall functionality of the resulting explainer. Moreover, fixing one building block can restrict the range of algorithms supported by other modules on a *conceptual* level – while they may be mechanistically compatible, their conjoined operation can be detrimental to the quality of explanations. This section discusses (often unintended) consequences of choosing a particular algorithm for each individual bLIMEy module (interpretable representation, data sampling and explanation generation) and offers best practices for this selection process. Throughout our discussion we refer to LIME – given its prevalence among surrogate explainers – identifying its possible sources of undesired behaviour and suggesting alternative modules that address many of such issues. We support our investigation with an in-depth theoretical analysis, empirical evidence and a collection of experimental results, providing a comprehensive view on the challenges of choosing appropriate surrogate building blocks.

#### 3.3.1 The Challenge of Building Surrogate Explainers

Given the complex nature of end-to-end surrogate explainers, many of them are built from generic and versatile components, focusing on the overall performance of such tools and not delving into selection and optimisation of their individual building blocks – see Figure 3.5. Understandably, these explainers seek to automate the whole process, which requires modules that can be operated without human guidance or intervention, thereby enabling their initialisation, deployment and evaluation at scale. This hands-off attitude is particularly detrimental to *interpretable data representations*, which tend to be based on (quantile) discretisation for numerical features of tabular data, (edge-based) super-pixel segmentation for images (e.g., quick shift [171]) and (whitespace-based) tokenisation for text. This design choice can be easily justified since creating an IR that is intelligible is often user- and application-dependent or even unique to the explained data point, therefore scaling it up is impractical without a concrete use case. However, the core

premise of interpretable representations is to encode concepts that are *meaningful* to the target audience, and so relying upon computer-generated IRs without understanding their behaviour and properties may be counterproductive.

As a result the important task of choosing appropriate surrogate components is often overlooked in the literature. It is common to assume that a particular module is given or to reuse one that was introduced in prior work without reviewing its suitability, (often implicit) assumptions, properties and caveats [84, 89, 183]. Such an attitude hampers creation of novel surrogate components and limits the scope of such explainers to measuring the *influence* of interpretable components on black-box predictions. Nonetheless, this type of sensitivity analysis geared towards explainability comes with numerous unaddressed issues, many of which originate from misconfiguration of individual surrogate building blocks. For example, to discern how a particular interpretable component (encoded by the IR) influences a black-box prediction, it needs to be “removed” and the resulting change in the model’s prediction quantified. Many black-box models, however, cannot predict incomplete instances, especially for tabular and image data, in which case this procedure becomes ill-defined and has to be replaced with a proxy – such as segment occlusion for images – possibly leading to biased and untrustworthy explanations.

The consequences of treating surrogates as end-to-end explainers without appreciating their individual modules reach beyond the aforementioned undesired behaviour of the interpretable representation, which is just one component needed to build them. When it comes to data sampling, it may be beneficial to perform this step in different representations depending on the data type. Recall that it is necessary to sample from the binary IR for sensory data (images and text), whereas for tabular data it can be done in either the interpretable or the original data domain. In particular, by sampling in the binary IR regardless of the data type, we forfeit control over locality of this procedure for tabular data – it was implicit for images and text (an individual image or sentence), however for tabular data it spans all of the hyper-rectangles, i.e., the entire feature space. Moreover, because of the ill-defined inverse transformation from the interpretable into the original domain of tabular data, it is difficult to avoid introducing explanation instability in this step; evaluating the quality of sampled data is also uncommon. The significance of the explanation generation steps is often overlooked as well – it defines the type of explanations and their presentation medium in addition to imposing limitations that are inherent to the chosen surrogate model, e.g., the assumption of feature independence for linear models. Notably, some types of local models may have benefits spanning multiple surrogate building blocks, for example, a tree-based surrogate can unify the interpretable representation and explanation generation steps for tabular data, thus decreasing the explainer’s complexity and introducing explanation types such as counterfactuals.

The versatility and adaptability of surrogate explainers, however, come at a cost: these tools are complex entities suffering from overparameterisation, which often manifests itself in multiple contributing sources of instability and low fidelity of the resulting explanations [84, 87, 133, 183].

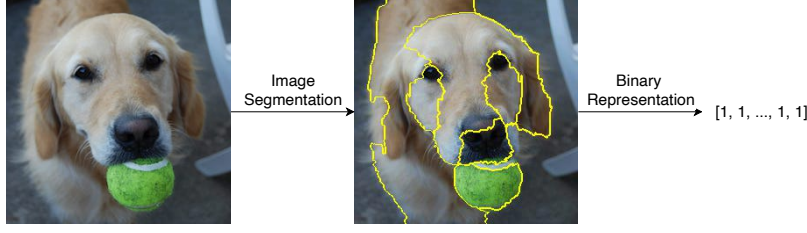
Nonetheless, by carefully choosing and configuring surrogate components we can alleviate most, if not all, of such issues. To this end, this section analyses individual building blocks in isolation; a view complemented by Section 3.4, which investigates how they work together and interact. Understanding both of these aspects is important as, for example, certain pairings of interpretable representations and surrogate model types can magnify otherwise insignificant problems or even render the entire explainer unreliable, especially when the implicit assumptions underpinning these components are at odds. These two areas of research are mostly under-explored for surrogate modules on their own and as a part of an explainer, potentially leading to sub-optimal design choices and inadequate explanations.

### 3.3.2 Interpretable Data Representation

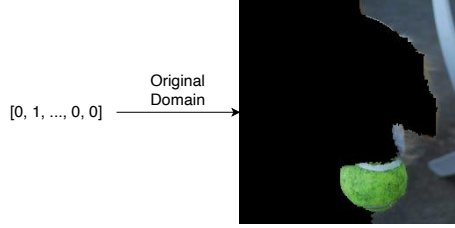
Interpretable representations – introduced in Section 3.2.1 – are arguably the most important component of surrogate explainers. To reiterate, they translate the low-level data domain necessary for good predictive performance into high-level human-intelligible concepts used to communicate the explanation. In particular, the explanation type and its cognitive complexity are directly controlled by the IR, allowing to target a particular audience and use case. These ramifications make the task of choosing an appropriate interpretable representation particularly important, but many explainers that rely on IRs overlook their merit and fall back on default solutions, which may introduce implicit assumptions, degrading the explanatory power of such techniques. To address this problem, we study properties and limitations of interpretable representations that encode presence and absence of human-comprehensible concepts with binary vectors, discussing their strengths and weaknesses for tabular, image and text data.

Our findings are a stepping stone towards building custom IRs that can be generated automatically while still representing (computationally) meaningful concepts and yielding faithful explanations. Among others, we discuss the implicit assumption of the *explanation locality* that is detrimental to its completeness, and a *transformation* between the original and interpretable domains that is *non-deterministic* in the opposite direction, which introduces unnecessary instability of the explanation, reducing its fidelity and soundness [149, 158]. We also touch upon discretisation-based IRs for tabular data, which impose an axis-parallel, grid-like structure, focusing on approaches such as quartile binning and feature space partition learnt with decision trees to better understand importance of black-box decision boundaries and possible information loss. Furthermore, we investigate implicit assumptions and consequences of using proxies when it is impossible to remove information from data as imposed by an IR. We support our claims with a range of experimental results that illustrate principles of designing reliable interpretable representations. In particular, we show how:

- parameterisation of the information removal proxies for images – such as segmentation granularity and occlusion colour – links to *explanation volatility*; and



(a) Transformation from the original domain into the interpretable representation  $\mathcal{X} \rightarrow \mathcal{X}'$ .



(b) Transformation from the interpretable representation into the original domain  $\mathcal{X}' \rightarrow \mathcal{X}$ .

**Figure 3.6:** Example of interpretable representation transformation in both directions for image data. Panel (a) depicts steps required to represent a picture as a binary on/off vector, and Panel (b) illustrates this procedure in the opposite direction. Both transformations are *deterministic* given a fixed image segmentation and occlusion colour.

- purity of feature space partition with respect to black-box predictions for tabular data affects *explanation faithfulness*.

This section provides an overview of our experimental findings, with additional results and derivations presented in Appendices C.1 and C.2.

### Understanding Interpretable Representations and Their Properties

The interpretable representations of image and text data (introduced earlier in Section 3.2.1) are *implicitly local* – they are only valid for the data point (image or sentence) for which they were created. Another shared property is *determinism* of the IR transformation procedure in both directions (within the scope of a single instance); there is a one-to-one correspondence between a data point and its interpretable representation given that the IR captures the structure of the explained instance (image or sentence) and operates within a fixed framework (segment occlusion with a predetermined colour or token removal) – see Figure 3.6. Transforming between the two domains deterministically therefore requires the IR to memorise adjacency of segments and their original pixel values for images, and order of tokens and their pre-processing for text. Notably, this property helps to ensure uniqueness of explanations, which is very important for their stability, hence preserving explainees’ trust [133, 149]. Out of the two IRs, the one for text has the advantage of allowing the interpretable components (tokens) to be *truly* removed from an explained instance (sentence) without a proxy that may be inconsistent with human perception

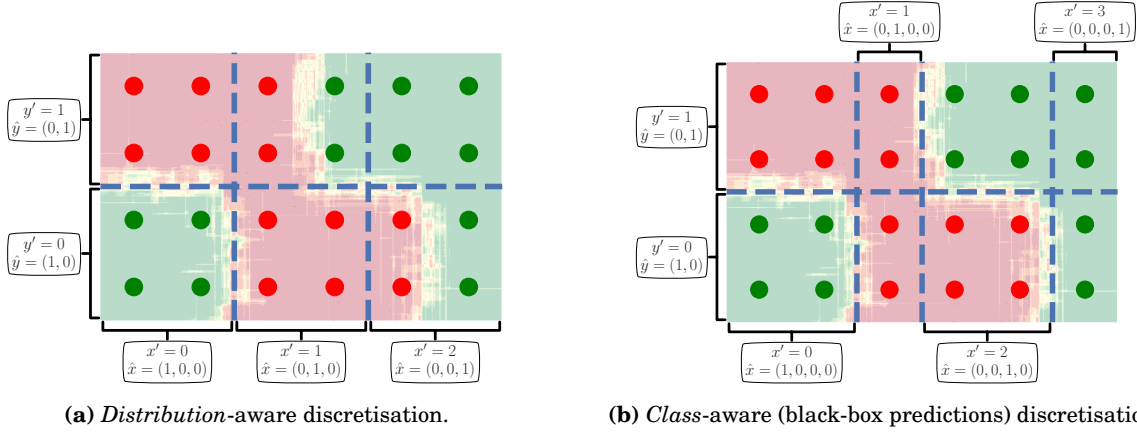
of meaningful changes applied to the relevant data domain. In this case, however, it is more of a property of the underlying predictive model rather than the IR itself – text classifiers are more flexible and do not assume input of a fixed length, while vision models cannot handle “missing” pixels.

Interpretable representations should encode concepts that are meaningful to humans as they are the foundation of the resulting explanatory artefacts. For image and text data, this is achieved by using (pre-processed) words instead of letters and ensuring that picture segments coincide with visually meaningful entities, e.g., ears, nose and tail in a dog photograph. For tabular data, instead of using each individual feature value of the explained instance as interpretable concepts, numerical attributes are discretised and categorical features can be optionally grouped, which is followed by binarisation – refer to Section 3.2.1 for more details. However, selecting the bin boundaries when discretising numerical attributes is non-trivial and biases the explanation akin to the influence of the segmentation granularity and occlusion colour on image explanations. Since generic, computer-generated interpretable representations may not capture meaningful concepts, explainee-driven interactive personalisation of IRs is an interesting avenue of research in this direction on the crossroads of interpretable ML and Human–Computer Interaction [151]. It has the potential to formulate guidelines for the design and operationalisation of IRs for individual applications – see Chapter 6 – but such a solution comes at the expense of a user-in-the-loop architecture that poses challenges for automation and scaling.

We can address this problem by borrowing from both approaches: building interpretable representations algorithmically with tools that are manually tuned (off-line) for each individual explanatory task. This research direction can create a considerable impact as it allows for *automatic* creation of IRs that encode (computationally) *meaningful* concepts. While interpretable representations for images and text naturally come with many of the desired properties, preserving them for tabular data is challenging – an observation that we explore throughout the remainder of this section. Notably, an IR defines the question that the explanation answers and restricts the types of explanation that can intelligibly communicate this information, e.g., importance and influence of interpretable concepts, counterfactuals or what-if statements. By understanding characteristics and behaviour of each interpretable representation and its influence on the resulting explanation – both on its own and in conjunction with a particular type of a surrogate model (see Section 3.4.1) – we can uncover the theoretical properties of such explainers and assess their applicability and usefulness for a problem at hand.

### **Faithfulness**

Interpretable representations of image and text data are computationally faithful since they are implicitly local, i.e., constructed with respect to the individual instance being explained. On the other hand, building IRs of tabular data that are conceptually equivalent and exhibit similar properties is an open challenge, which in the best case would require domain experts to manually



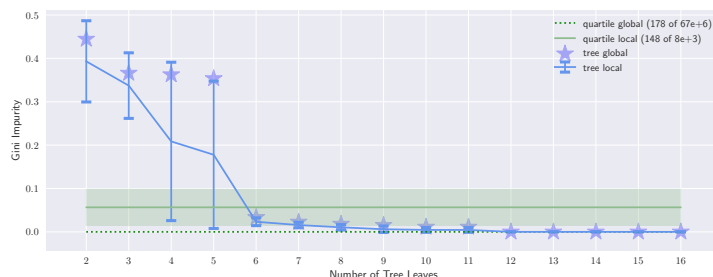
**Figure 3.7:** Discretisation is the main building block of interpretable representations of tabular data. It can either be learnt based on data features alone – Panel (a) – or additionally consider their black-box predictions (background shading) – Panel (b).

partition the attribute values to create meaningful concepts. Nonetheless, when composing them algorithmically, two factors determine their quality and faithfulness: the *data point* selected to be explained – which specifies the reference hyper-rectangle and neighbourhood – and the ability of the *discretisation* algorithm to locally approximate the black-box decision boundaries. Since the former is chosen by the explaine, only the latter can be controlled, making the discretisation either explicitly **global**, i.e., learnt with respect to a whole data set, or **local**, thus focusing on a specific region. Moreover, each type can either observe just the *data distribution*, or additionally take into account their *black-box predictions*, presenting us with two distinct approaches:

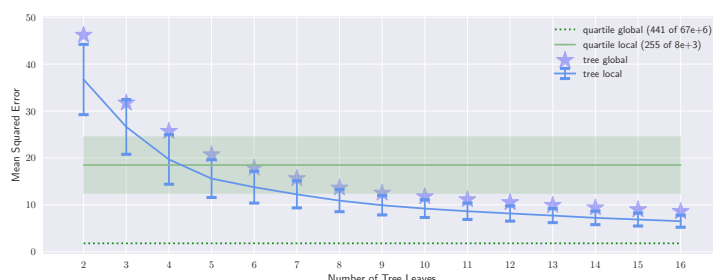
**distribution-aware** discretisation – Figure 3.7a – is based on the density of data in the local or global region being explained, e.g., quantile discretisation; and

**class-aware** discretisation – Figure 3.7b – partitions data according to black-box decision boundaries confined within the local or global region being explained.

Since the predominant role of *local* surrogate explanations is to approximate and simplify the behaviour of a black box near a selected instance, the latter type should be preferred. It is a stepping stone towards representing human-intelligible concepts that are coherent with predictions of the investigated model, thus producing faithful and appealing insights. However, to the best of our knowledge, class-aware discretisation approaches have not been explored in the XAI and IML literature. Computationally, their objective can be expressed as maximising the *purity* or *uniformity* of each hyper-rectangle with respect to the black-box predictions of data that it encloses, weighted by the proportion of these data to account for their uneven distribution across partitions. For example, for probabilistic and regression black boxes this criterion can be calculated as the Mean Squared Error (MSE), and for crisp classifiers as the Gini impurity. This observation suggests that **learning interpretable representations with**



(a) Weighted average of the Gini impurity computed for IRs generated for the *wine* data set (classification).

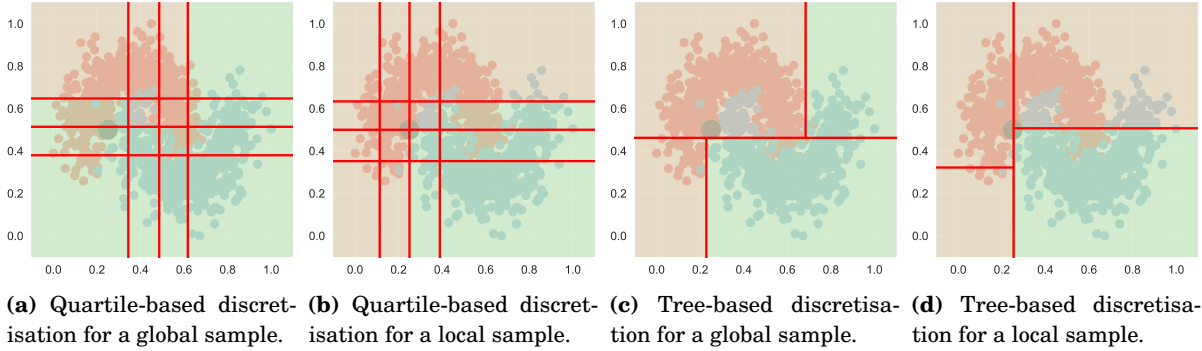


(b) Weighted average of the mean squared error computed for IRs generated for the *housing* data set (regression).

**Figure 3.8:** Interpretable representations based on decision trees result in purer hyper-rectangles (y-axis, lower is better) and fewer encodings (x-axis) when compared to equivalent quartile-based IRs, i.e., they are more flexible and expressive. The number of unique encodings used by quartile-based IRs is constant for a data set and is displayed in the legend (presented as the number of encodings used, out of the theoretical limit supported by the representation); whereas for tree-based IRs, it is equivalent to the number of leaves, which is recorded on the x-axis. See Figure C.2 and Appendix C.2 for more details.

**decision trees** (specifically, the data space partitions they create) is a promising approach to optimising the aforementioned objective. Our experiments to assess purity of IRs generated with various methods – summarised in Figure 3.8 – support this claim, with a more detailed analysis presented in Appendix C.2. Notably, while tree training procedures tend to be greedy, alternatives that consider multiple features at any given iteration could improve the quality of the resulting IRs even further.

Similar to images and text, the *binary* interpretable representation of tabular data is specific to the explained data point and, more generally, its hyper-rectangle. Nonetheless, the underlying *discretisation* can be reused for explaining any instance from the same data set. While a common practice [129], such an approach undermines faithfulness of the resulting explanations – the goal is to produce a *local* explanation of the selected data point, hence the discretisation should be truthful within the explained neighbourhood. Neither globally (based on an entire data set) nor locally (based on a local data sample) faithful discretisation can capture uniqueness of a black-box decision boundary universally well for all the possible data subspaces [158]. Therefore, reusing the same discretisation to generate individual IRs for tabular data can be compared to creating a

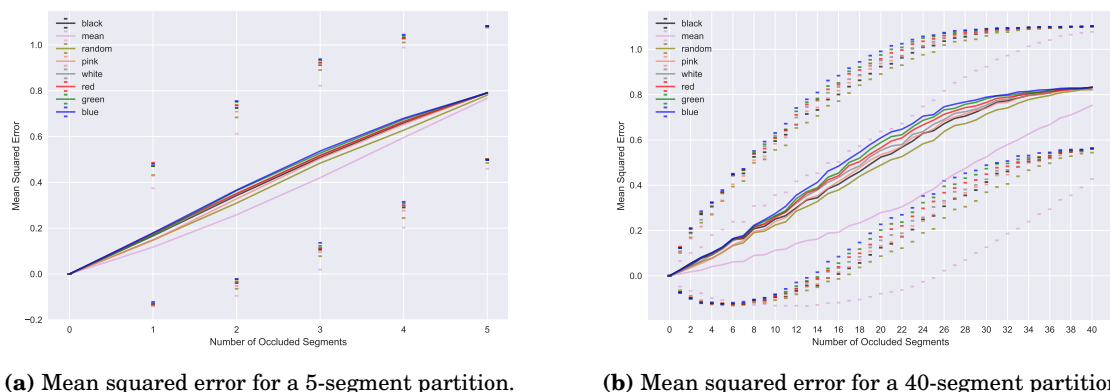


**Figure 3.9:** Interpretable representations learnt for the two-dimensional two moons data set. A global (a&c) or a local (b&d) data sample is used in combination with a quartile (a&b) or a decision tree-based (c&d) discretisation. Local approaches (b&d) are better at capturing the intricate behaviour of the black-box decision boundary in the neighbourhood of the explained instance (black dot). Additionally, tree-based interpretable representations (c&d) require less partitions and are more faithful since they account for the black-box predictions.

super-pixel partition of a specific image and reapplying it to other, unrelated images, yielding conceptually meaningless interpretable representations. This phenomenon can be observed in Figure 3.9, which depicts local – Panels (b) and (d) – and global – (a) and (c) – discretisations based on quartiles and decision trees. Additionally, Figure 3.9d shows how trees, which account for the black-box predictions, can faithfully approximate a decision boundary with a relatively few partitions in the explained neighbourhood.

### Information Removal and Loss

Any operationalisation of interpretable representations requires “switching off” human-intelligible concepts, which translates to removing tokens for text, however a similar procedure is impossible for image and tabular data, where a proxy is needed. To this end, image segments are often occluded with patches of a solid colour, but such a strategy comes with its own implicit assumptions and limitations, which are often overlooked. For example, LIME [128] replaces super-pixels with their mean colour to remove their content without acknowledging the unintended consequences of this choice. In such a setting, segments that have a relatively uniform colour gamut may, effectively, be impossible to occlude; this is especially common for segments that are in the background or out of focus, e.g., bokeh and depth-of-field effects. Furthermore, whenever the segmentation coincides with objects’ edges or regions of an image where colour continuity is not preserved (which is common for edge-based segmenters), occluding super-pixels with their mean colour causes (slight) colour variations between adjacent segments. These artefacts preserve edges in a (partially) occluded image, and they often retain enough information for a black-box model to correctly recognise its class (for an example refer back to Figure 3.3). Segmentation granularity is also important: the smaller the segments are, the more likely it is that their colour

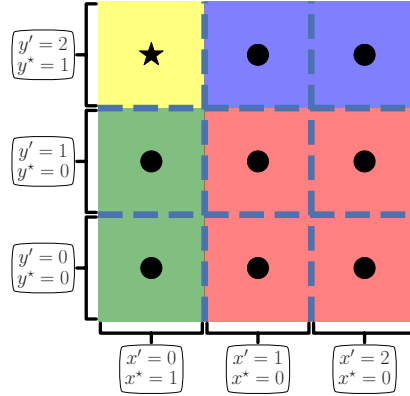


**Figure 3.10:** Mean squared error (y-axis) calculated between the top prediction of an image (probability estimate) and predictions of the same class when progressively occluding a higher number of segments (x-axis) with a given colouring strategy. The panels show that the mean occlusion strategy is not as effective at hiding information from the black box as using a single colour for all of the super-pixels (regardless of the colour choice). Similarly, randomising the occlusion colour for each individual segment does not seem to have the detrimental effect observed for the mean colouring. The plots also indicate that when an image is split into more segments, the ineffectiveness of the mean colouring approach gets magnified due to the increased colour uniformity of individual super-pixels – a “blurring” effect. See Appendix C.1 and Figure C.1 for more details.

composition is uniform given the “continuity” of images, i.e., high correlation of adjacent pixels, resulting in a similar effect as above.

Since most of these issues are consequences of using the mean-colour occlusion, it may appear that fixing a single masking colour for all of the segments would eradicate some of these problems. Such an approach hides the edges between occluded super-pixels and discards their content instead of just “blurring” the image, however the edges between occluded and preserved segments remain visible. Moreover, the choice of the masking colour may impact the explanations themselves regardless of the colouring strategy. This particular type of proxy for removing information from image segments implicitly assumes that the black-box model is *neutral* with respect to the occlusion colour, i.e., none of the modelled classes is biased towards it. Adjusting the granularity of the segmentation also plays an important role given the high correlation of adjacent super-pixels. We support these observations with a range of experiments done for occlusion-based interpretable representations of images, the results of which are detailed in Appendix C.1 and summarised in Figure 3.10. In particular, they exemplify the degree to which the segmentation granularity as well as the occlusion strategy and colour affect the resulting explanations.

For tabular data – in contrast to text and comparably to images – *removing* information from the original representation by setting a member of the binary IR to 0 is not possible and requires a proxy. Within our operationalisation of the interpretable representation of tabular data,

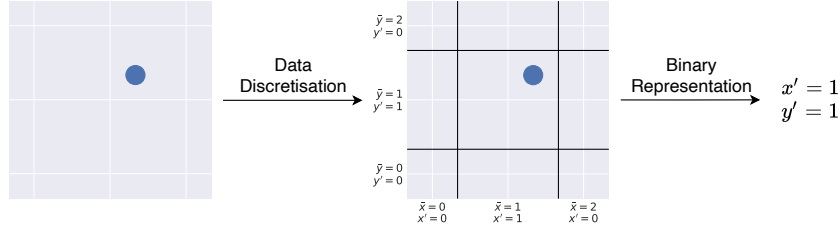


**Figure 3.11:** Some hyper-rectangles  $(x', y')$  – created with discretisation – become indistinguishable in the binary interpretable representation  $(x^*, y^*)$  of tabular data. The  $\star$  marker indicates the explained instance and the background shading illustrates unique binary (IR) encodings.

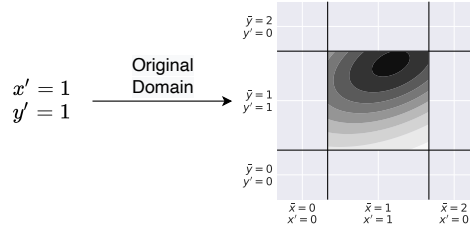
“switching off” an interpretable component is equivalent to placing the value of the corresponding numerical feature outside of the range encoded by this concept. Similarly, for categorical features this step is equivalent to choosing any other value not encoded by the corresponding interpretable concept. This procedure ensures diversity of data in the explained neighbourhood at a high price of introducing two sources of randomness when manipulating the IR, which is a consequence of information loss incurred by *non-deterministic* transformations between various data domains.

The intermediate discrete representation of a tabular IR uniquely encodes each hyper-rectangle – see the  $(x', y')$  coordinates in Figure 3.11. However, the same is not true for the *binary* interpretable representation if any categorical feature has more than two unique values or any numerical attribute is partitioned into more than two intervals. In such cases, the transformation between these two representations loses information as depicted by the background shading and the  $(x^*, y^*)$  coordinates in Figure 3.11. For each of these binary features, 1 is assigned to the partition that contains the explained data point and 0 to all the other intervals, effectively making them indistinguishable. Similarly, information is lost when transitioning from the original representation of data into their discretised form: each hyper-rectangle, which has unique coordinates in this domain, contains multiple data points that become indistinguishable – see Figure 3.7 for reference. These two many-to-one transformations – needed to create an interpretable representation of tabular data based on discretisation – contribute to non-determinism, which is discussed below.

The impossibility to distinguish data belonging to different hyper-rectangles in the binary interpretable representation is particularly detrimental to capturing the complexity of a black-box decision boundary. While the underlying discretisation may have closely approximated its intricacies, this information can be lost when transitioning into the binary representation, especially if the decision boundary runs across hyper-rectangles that were merged in this process. For example, consider the discretisation shown in Figure 3.7b assuming that the explained



(a) Transformation from the original domain into the interpretable representation  $\mathcal{X} \rightarrow \mathcal{X}'$ .



(b) Transformation from the interpretable representation into the original domain  $\mathcal{X}' \rightarrow \mathcal{X}$ .

**Figure 3.12:** Example of interpretable representation transformation in both directions for tabular data. Panel (a) depicts the discretisation and binarisation steps required to represent a data point as a binary on/off vector, and Panel (b) illustrates this procedure in the opposite direction. The forward transformation is *deterministic* given a fixed discretisation (binning of numerical features), however moving from the IR to the original domain is *non-deterministic* and requires random sampling.

instance resides in the  $(x', y') = (1, 1)$  hyper-rectangle – top row, second from the left. In the binary representation, the remaining top-row hyper-rectangles  $(0, 1)$ ,  $(2, 1)$  and  $(3, 1)$  would be grouped – akin to the process depicted by the background shading in Figure 3.11 – thus losing the information that the first one belongs to the red class and the latter two to the green class.

### Transformation Determinism

A direct consequence of the information loss suffered by *tabular data* is **non-determinism** of the transformation from the interpretable representation to the original data domain – see Figure 3.12 – which increases the volatility and instability of the resulting explanations. Note that this is not the case for image and text IRs, which we have discussed earlier – refer back to Figure 3.6. Recall that a sentence can be easily represented as a binary vector encoding presence or absence of unique word-based tokens and such a binary vector can be then deterministically transformed back into a sentence. In particular, by memorising the skeleton of the sentence, i.e., the pre-processing applied to words and their ordering, we can fully reconstruct the original text structure without selected words. Similar reasoning applies to images where a binary vector indicates whether a super-pixel should be occluded or preserved; capturing the location, adjacency and pixel values of each segment therefore allows us to deterministically recreate the original image without selected parts (through occlusions).

To avoid the (unnecessary) IR randomness adversely affecting the stability of explanations produced by surrogate explainers, transforming the data from their original domain  $\mathcal{X}$  into an interpretable representation  $\mathcal{X}'$  and the reverse procedure must both be **deterministic**, i.e., the mapping between  $\mathcal{X}$  and  $\mathcal{X}'$  has to be a *one-to-one correspondence*. However, transitioning from the original into a discrete representation for tabular data is a many-to-one operation if the underlying data set contains numerical attributes – see Figure 3.12. Furthermore, transforming the discrete representation into a binary IR is also a many-to-one mapping if the former has more than two unique values for any discretised feature. The original data point therefore cannot be reconstructed after passing through these two steps.

Recall that surrogate explainers tend to sample data from the (binary) interpretable representation, which implicitly introduces locality of these instances and the resulting explanations. While reasonable for images and text, following the same procedure for a tabular IR entails reversing the binary sample back to the original domain, which requires *random sampling*, making this transformation non-deterministic [152, 158]. This process involves, first, choosing at random one of the concatenated hyper-rectangles if the binary component is 0 (1 uniquely identifies a hyper-rectangle); next, sampling a numerical value from the range defined by this partition, e.g., using a Gaussian distribution (with clipping at bin boundaries) fitted to the (training) data enclosed by this hyper-rectangle (cf. Figure 3.12b). The value of a categorical feature, on the other hand, is uniquely identified by a hyper-rectangle, making the second – but not necessarily the first – part of this transition deterministic. Notably, the (numerical) sampling step embedded in this procedure is the unidentified source of randomness reported by Zhang et al. [183].

Image and text data require manipulating the binary IR since sampling from raw text or pixels is semantically an ill-defined procedure. However, tabular data can be augmented in their original representation, providing an opportunity for an algorithmic workaround of the non-deterministic transformation from  $\mathcal{X}'$  to  $\mathcal{X}$ . Data drawn from the original domain can be directly predicted by a black box to capture its behaviour, and then easily transformed into a discrete and binary representation (the IR) to train a surrogate model. Doing so no longer requires the  $\mathcal{X}'$  to  $\mathcal{X}$  transition, but in such a setting this procedure can be made (algorithmically) deterministic by memorising the correspondence of sampled data between different representations when executing the forward ( $\mathcal{X} \rightarrow \mathcal{X}'$ ) transformation. This matching can be compared to storing segment adjacency and original pixel values for images or a sentence skeleton and word pre-processing for text.

Therefore, by sampling in the original domain and memorising different representations of each instance, we avoid using the non-deterministic inverse IR transformation of tabular data, hence reduce randomness and improve stability of the resulting explanations. Importantly, if a need for this step arises, we can always exploit the data point mapping, making the  $\mathcal{X}' \rightarrow \mathcal{X}$  transition algorithmically deterministic. Nonetheless, this sampling strategy forfeits the implicit locality (albeit weak for tabular data) achieved by operating directly on the binary representation,

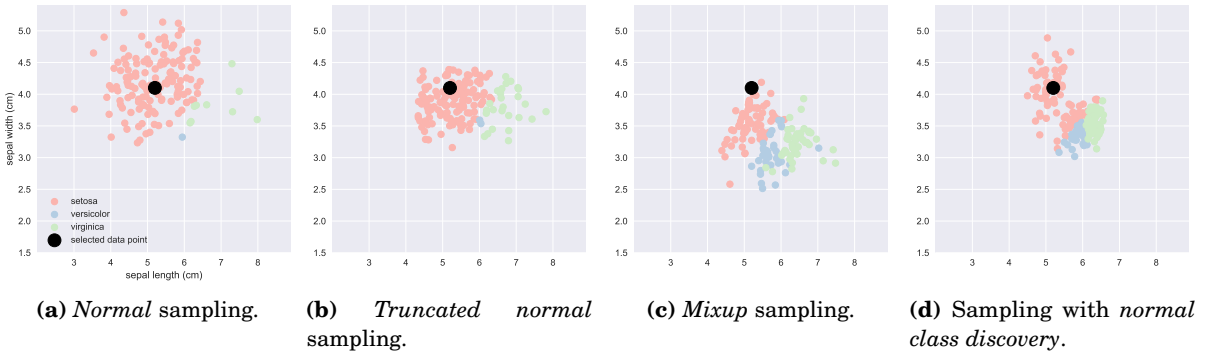
therefore the substitute sampling algorithm should be *explicitly local* to capture the explained subspace in detail [158]. While the binary representations of images and text are local, this is not entirely the case for tabular data where the all-1 vector encodes the explained hyper-rectangle but the remaining partitions span the *entire* feature space. In this scenario, additional measures to enforce explanation locality may be useful, for example, it can be controlled by weighting the sample by their closeness to the explained instance (in the original domain  $\mathcal{X}$ ).

### 3.3.3 Data Sampling

The sampler and its configuration determine the *breadth* and *scope* of data that the surrogate model is trained upon, which affects the faithfulness of the explanations. Recall that images and text require us to sample in the interpretable domain, which makes the augmented data implicitly local and of relatively small dimensionality since they are drawn from a binary space. In particular, a binary domain with  $d$  dimensions has  $|\mathcal{B}^d| = 2^d$  unique elements, therefore for small IRs it is often more beneficial to use the entire data space instead of a random sample. In our experience, images are usually partitioned into roughly 10 interpretable concepts, and plain English sentences have 15–20 words on average [29] including stop words; a random sample of 10,000 instances is a common practice [129] but an IR with 13 concepts has only  $2^{13} = 8,192$  unique data points. This observation suggests that generating a complete sample is an attractive alternative to random sampling, helping to improve the fidelity of surrogates, as we show in Chapter 5.

For tabular data, a similar approach – sampling in the binary interpretable domain – is feasible but impractical since the number of interpretable features is proportional to the number of attributes in the original domain, which tends to be high. Moreover, the *non-determinism* of the  $\mathcal{X}' \rightarrow \mathcal{X}$  transformation procedure and problems with *sample locality* are detrimental to the uniqueness and comprehensiveness of the augmented data, making this strategy less attractive. Instead, sampling can be performed in the original data domain, however the algorithm should guarantee a *diverse* and *explicitly local* sample. Generating data that extensively cover the neighbourhood of the explained instance helps to improve fidelity of the surrogate model, and ensuring that they span multiple classes (or a diverse range of class probabilities/regression values) allows to discover nearby decision boundaries, thereby fit a *meaningful* local surrogate. The former often depends on the parameterisation of the sampler, whereas the latter requires an algorithm that accounts for the data labels or their black-box predictions, with Mixup [182] and Growing Spheres [88] being good candidates. The sample diversity can be measured with the Gini impurity and mean squared error respectively for crisp and probabilistic classifiers (and regressors), which is an approach similar to monitoring purity of tabular IRs; however, here we strive for diversity, not purity.

Figure 3.13 explores the behaviour of different sampling strategies applied to the original domain of the Iris data set [41] (tabular), exemplifying the importance of choosing an appropriate



**Figure 3.13:** Effect of different sampling algorithms on the *locality* and *diversity* of the sample when applied to the original domain of the Iris data set. The panels are plotted along *sepal length (cm)* on the x-axis and *sepal width (cm)* on the y-axis, and the black dot represents the explained instance for which the sample is generated. Red, blue and green markers – the three classes of the Iris data set – capture black-box predictions. These experiments show the advantage of sampling algorithms that are aware of the class distribution – (c) *Mixup* and (d) *normal class discovery* [160] – which allows them to generate a diverse sample that discovers the local decision boundary. This information helps the surrogate to better approximate the behaviour of the black box in the explained neighbourhood, thus improving explanation faithfulness.

sampler when building surrogate explainers. It plots the data along two dimensions – *sepal length (cm)* on the x-axis and *sepal width (cm)* on the y-axis – to facilitate easy visual comparison. The three marker colours indicate the *setosa*, *virginica* and *versicolor* classes of the Iris data set. We initialise each sampler with the full data set and generate 150 instances around the explained data point, which is represented by the black dot. Some of the data samplers can have difficulties locating the closest black-box decision boundary when the explained instance is in a high-confidence region, which may be common for a large number of attributes due to the curse of dimensionality – see Panels (a) and (b) in Figure 3.13. When the black-box predictions of the generated data are relatively uniform, fitting a local surrogate model may be impossible or it may under-perform, thereby providing misleading or meaningless explanations. Another important aspect of the sampled data is their class imbalance, which needs to be accounted for when training the surrogate predictor.

Sampling from the original domain of tabular data can be improved further by adapting active learning algorithms. Such a strategy can explore the space around the explained instance according to some diversity criterion, treating the black-box model as an oracle, thus reducing the size and improving the quality of the generated data. While querying the black box may be expensive, the data sample has to be predicted anyway since this information is needed to train the local surrogate. Another caveat is consistency of the sample with respect to the density of the underlying data as out-of-distribution instances may be detrimental to the quality of the resulting explanations [123]. Alternatively, the sampling may be centred on a black-box decision boundary that is closest to the explained data point to shift the focus of the explanations more

towards the behaviour of the black box [89]. Regardless of the approach taken, the data sampling component of surrogate explainers built with bLIMEy tends to be *inherently random*. In such a setting, ensuring explanation reproducibility requires always outputting the same (local) sample, which can only be achieved by fixing the random seed, thereby creating an illusion of stability.

### 3.3.4 Explanation Generation

The final building block of surrogates is explanation generation, which consists of:

- computing similarity between the explained instance and sampled data;
- selecting a subset of interpretable features; and
- fitting a (local) surrogate model.

The first two steps are optional and are, respectively, responsible for focusing the explanations on a chosen neighbourhood and introducing explanation sparsity. The surrogate model is trained on the interpretable representation of the sampled data and their black-box predictions (for a selected class), which can be weighted by similarity scores. In the case of tabular data, where the number of (interpretable) features may be overwhelming, a dimensionality reduction technique can be applied to introduce explanation sparsity. The type of the surrogate model as well as the choice, configuration and parameterisation of these steps all affect the resulting explanation to a varying degree.

**Sample Similarity** By weighting the sampled data based on their similarity to the explained instance, we can introduce or enforce locality of the explanation and refine its scope. Explanations of images and text are already local since the underlying interpretable representation operates within a single picture or sentence. Nonetheless, weighting the binary data sample during the surrogate training procedure, e.g., by using the edit distance passed through a kernel, can induce preference for shorter explanations, i.e., a smaller number of tokens removed and super-pixels occluded. For tabular data, on the other hand, the utility of this step varies depending on the data domain in which sampling is performed. If data were augmented in the discrete or binary representation, the sample may span the entire feature space, making this step the *only* mechanism to introduce explanation locality. When sampling in the original domain, however, the scope of the resulting data depends on the properties of the sampling algorithm, therefore weighting can either be the *only* apparatus to introduce explanation locality, or it can *reinforce* the result achieved with an explicitly local sampler.

Image and text data present us with little choice with respect to the domain in which we can calculate the similarity scores. There is no meaningful distance metric (from an explainability standpoint) that works directly on individual pixels or letters, therefore it has to be computed on the binary interpretable representation. Selecting an appropriate distance function depends on

the application and expected results, hence this task remains an open research question; however, the *edit distance* is an appealing choice given the structure of the corresponding IRs, in which setting it counts the number of “switched off” concepts. For tabular data, on the other hand, this procedure is more flexible and the distance can be computed in the original, discrete or binary domain, using a chosen metric such as the Euclidean, Manhattan or cosine distance. Again, either of the choices depends on the intended use, the data domain and the selected surrogate model, without clear guidelines originating from our research. Finally, since we are interested in similarity – which is inversely proportional to distance – we need to transform these measures with a kernel, for example, the *exponential kernel* used by LIME [129]. Kernel choice and its parameterisation are subjective and application-dependent, and they should be adapted based on the desired degree of locality enforcement. In summary, there are no obvious choices for this step of the explanation generation module and any decision should be based on a thorough analysis of each individual use case.

**Explanation Sparsity** Reducing the dimensionality of the interpretable domain for image and text data is detrimental to the quality of explanations as it would result in “black holes” in images and missing words in sentences, therefore it should be avoided. While bar plots depicting such explanations may become overwhelming for a large number of interpretable components, the influence measures can also be superimposed on top of the original data point, which alleviates this problem – refer back to Figures 3.1 and 3.2. Nonetheless, tabular data can only be explained with bar plots when they have more than two attributes since it is impossible to visualise high-dimensional spaces. Therefore, when the number of (interpretable) features is large, a subset should be selected to shrink the collection of “influential factors” presented to the explainee, which leads to shorter and more comprehensible explanations. This type of explanation sparsity can be achieved by discarding uninformative (interpretable) attributes with methods such as *lasso path* (K-LASSO), *forward selection* or *highest weights* [129], and it should be considered a necessity for tabular data with many features.

**Surrogate Model** With all the other components in place, the final stage of building a surrogate explainer is fitting a local model, which can be done in a number of different ways:

- a regressor of probabilities output by a black-box classifier;
- a regressor of numerical values predicted by a black-box regressor; or
- a classifier trained for a crisp black-box classifier or a thresholded probabilistic model.

Another choice is the training scheme: the surrogate model can either be trained as a *multi-class* or *one-vs-rest* predictor, a distinction that applies naturally to surrogate classifiers but needs to be adjusted for surrogate regressors. A plain surrogate regressor mimicking a probabilistic black-box predictor is restricted to modelling, hence explaining, probabilities of a single class selected by the

user, which establishes a one-vs-rest approach since the complementary probability spans all of the classes that are not being explained. However, by using multi-output regression models [21] an arbitrary number of classes can be modelled and explained simultaneously, thus imitating a multi-class scenario – see Chapter 5 for more details. In our experience, surrogate regressors tend to perform better than surrogate classifiers for black-box probabilistic models as treating them as argmax classifiers may result in low fidelity of the surrogate whenever these models are overconfident or otherwise poorly calibrated [82].

Choosing the type of a surrogate model is also crucial as it determines the explanatory artefacts and their meaning. If local influence of (interpretable) concepts is desired, a linear model is a good pick as long as all of the features are normalised to the same range and they are “reasonably” independent. While such an approach is limited to measuring influence, a different type of explanations can be generated with a tree surrogate, e.g., counterfactuals and logical conditions outlining the behaviour of a black-box model in the neighbourhood of the selected instance. In the simplest form, the last split of the tree can be used to retrieve a class-contrastive (counterfactual) statement conditioned on a single (interpretable) feature, for example, “had this word not been in the sentence...” or “had this image segment been occluded... the prediction would be different”. The selection here should be motivated by the desired type of the explanation – e.g., “Why class A?” or “Why class A and not B?” – and its format – feature importance or influence depicted as a bar plot, or a conjunction of logical conditions – both of which are application-dependent.

### 3.4 Tailor-made Surrogate Explainers

Having discussed the ins and outs of each individual surrogate building block, we move on to analyse the properties of their various compositions. For example, a surrogate explainer can be built to inspect different aspects of a black box either *locally* for a selected prediction or *globally* by mimicking its behaviour in the entire data space. Depending on the intended audience, the expected level of domain expertise and familiarity with machine learning concepts, one type of explanation may be more appealing than another. Notably, the choice of interpretable representation and surrogate model family determines the *kind* and *meaning* of the resulting explanations. This is an important observation since conflicting pairings of these two components tend to be the main source of poor explanatory performance. Tabular data are particularly vulnerable to such effects given the intrinsic complexity of the underlying IR and all of its inefficacies, which are especially pronounced when modelling it with a linear surrogate. Therefore, data samplers, interpretable representations and surrogate models need to be studied on their own and as parts of an end-to-end explainer, providing invaluable insights for building, orchestrating and tuning surrogates.

In particular, this section examines how the choice of a surrogate model influences the

resulting explanations in view of the properties and limitations of the underlying IR. We study the characteristics and constraints of surrogate explainers originating from the properties and caveats of their building blocks, e.g., the assumption of (interpretable) feature independence imposed by linear models. To this end, we analyse tabular data with numerical features in a LIME-like surrogate setting, where influence-based explanations are determined by the coefficients of a linear model. Specifically, we illustrate the limited explanatory capabilities of an interpretable representation built upon discretisation of continuous attributes when paired with Ordinary Least Squares (OLS), a detailed derivation of which is given in Appendix C.3. Such explainers lose a precise encoding of the black-box decision boundary and can be manipulated by altering the distribution of the data sample used to train the OLS, undermining the reliability of the resulting explanations. As a solution, we propose using decision trees to both partition (discretise) the feature space and generate explanations, e.g., with counterfactuals [169], thus merging the creation of interpretable representation and explanation generation steps. We finish with an overview of fidelity-based evaluation strategies – measuring how well surrogates approximate the black box with a loss function such as the one given in Equation 3.2 – which are complementary to the inspection approaches that we proposed separately for each individual building block.

### 3.4.1 Compatibility of bLIMEy Modules

Binary interpretable representations can be paired with linear models [45, 129] to explain black-box predictions with concept importance or influence, meaning that such explanations are subject to assumptions and limitations of these models. In particular, the coefficients of linear models can be deceiving when the target variable is *non-linear* with respect to data features, the attributes are *co-dependent* or *correlated*, and the feature values are *not normalised* to the same range [150, 158]. Intuitively, the first two properties may not hold for high-level interpretable representations since their components are highly inter-dependent, e.g., adjacent image segments, neighbouring words and bordering hyper-rectangles, thus the resulting explanations can misrepresent the relations between these concepts. Friedman et al. [45] addressed some of these issues by using logical rules extracted from random forests as binary interpretable concepts, which they modelled with a linear predictor, but the overlap between these rules still violates the feature independence assumption. A lack of attribute normalisation, on the other hand, causes the model coefficients to be *incomparable*, rendering the explanation uninformative and misleading. LIME satisfies this criterion by using *binary* interpretable representations or otherwise explicitly normalising the features.

To overcome these limitations and facilitate explanations more diverse than influence of interpretable concepts, alternative surrogate models can be used [158]. Logical models, such as *decision trees*, are particularly appealing given that they provide a wide range of explanations (see Chapter 4) and do not introduce any restrictions on the features, although they do impose axis-parallel partitions of the data space [150]. Decision trees are particularly suited

for explaining tabular data, for which they alleviate the need for an independent interpretable representation as noted in Section 3.2.1. In particular, they can automatically learn a locally faithful, class-aware feature discretisation, with the added benefits of modelling combinations of hyper-rectangles, not suffering from information loss and not requiring the non-deterministic  $\mathcal{X}' \rightarrow \mathcal{X}$  transformation [158].

On the other hand, using linear models to capture influence of binary concepts for tabular data is particularly *problematic* when utilising the interpretable representation introduced in Section 3.2.1. The information loss suffered when transitioning from the discrete into the binary representation partially forfeits the preceding effort to faithfully capture the black-box decision boundary (during the discretisation step). We can demonstrate this by deriving an analytical solution to OLS, which is shown in Equation 3.3. Notably, it highlights an unexpected influence of the number of data points sampled in each hyper-rectangle on the resulting explanations (magnitudes of concept influence) and irrelevance of feature partitions other than the ones directly enclosing the explained instance.

$$\Theta = \begin{bmatrix} 1 & \frac{w_{11}+w_{10}}{\sum w_{ij}} & \frac{w_{11}+w_{01}}{\sum w_{ij}} \\ 1 & 1 & \frac{w_{11}}{w_{11}+w_{10}} \\ 1 & \frac{w_{11}}{w_{11}+w_{01}} & 1 \end{bmatrix}^{-1} \times \begin{bmatrix} \bar{y}_{\mathcal{W}} \\ \bar{y}_{\mathcal{W}_{11} \cup \mathcal{W}_{10}} \\ \bar{y}_{\mathcal{W}_{11} \cup \mathcal{W}_{01}} \end{bmatrix}. \quad (3.3)$$

Equation 3.3 shows the dependence of OLS coefficients on various characteristics of the binary interpretable representation of tabular data with two numerical features, both when the intercept is modelled (red & blue shading) and without it (blue shading). For comprehensibility, our analysis is limited to two-dimensional data such as those shown in Figure 3.11, but the result generalises to an arbitrary number of dimensions. Similarly, Equation 3.3 is specific to surrogate regressors, however it can be easily extended to linear classifiers. In particular,  $\mathcal{W}_{ij}$  is the set of (sampled) data points enclosed by the hyper-rectangle with  $(i, j)$  coördinates in the binary interpretable representation;  $\mathcal{W}$  is the set of all the (sampled) data;  $w_{ij}$  is the count of instances in the  $\mathcal{W}_{ij}$  partition (i.e.,  $|\mathcal{W}_{ij}| = w_{ij}$ ); and  $\bar{y}_{\mathcal{W}_{ij}}$  is the average black-box prediction of the instances within the  $\mathcal{W}_{ij}$  hyper-rectangle. Therefore, the influence of interpretable concepts is *solely* based on:

- **the proportion determined by the number of the data points** residing in the explained partition  $\mathcal{W}_{11}$  divided by the hyper-rectangles aligned with the explained partition along every axis, i.e.,  $\mathcal{W}_{11} \cup \mathcal{W}_{10}$  for the first feature and  $\mathcal{W}_{11} \cup \mathcal{W}_{01}$  for the second; and
- **the average value predicted** in the latter two subspaces –  $\mathcal{W}_{11} \cup \mathcal{W}_{10}$  and  $\mathcal{W}_{11} \cup \mathcal{W}_{01}$  – by the black box (scaled appropriately when the intercept is modelled).

Additionally, the intercept value is determined by:

- *the proportion* given by the number of data points in the hyper-rectangles aligned with the explained partition along every axis divided by the total number of data points; and

- *the average* value predicted by the black box for all the data points.

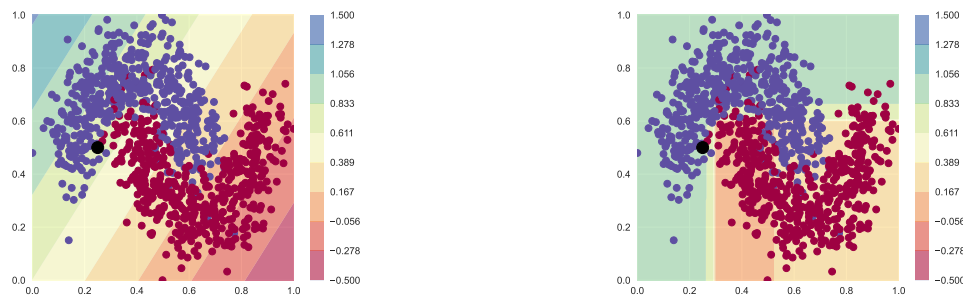
The full derivation of Equation 3.3 and an in-depth analysis of the consequences of this result are presented in Appendix C.3.

### 3.4.2 Tree-based Surrogates

LIME’s goal is to explain a prediction by quantifying its sensitivity with respect to changes in the interpretable domain such as removal of tokens (words) for text, occlusion of segments in images and jumping between hyper-rectangles for tabular data. This setting can be understood as a sensitivity analysis tool operating on an interpretable representation, thus motivating the use of a *linear model* as the local surrogate with its coefficients serving as the explanation. However, this is the only possible explanation type that can be extracted from such a surrogate, which may be inappropriate for a lay audience due to the technical background knowledge required to understand it, especially for tabular data. Notably, during our analysis of individual surrogate building blocks and their end-to-end compositions, decision trees exhibited favourable properties in numerous situations. Here, we investigate this advantage for tabular data, for which decision trees unify the interpretable representation and explanation generation modules, thus mitigating the need for the problematic discretisation and binarisation steps.

The (local) partition of a feature space learnt by a tree optimises for purity by default since each region (hyper-rectangle) is determined by a single leaf. The tree fitting objective is consistent with the goals of creating an interpretable representation for tabular data, thereby allowing to faithfully approximate a decision boundary of the explained black box. Furthermore, the quality of this IR can be controlled by relaxing various generalisation constraints (such as the depth or width limit of the tree and the number of training instance required to build a leaf), thus allowing the tree to overfit the explained region. Appendix C.2 gathers a collection of experiments documenting these benefits. Additionally, tree-based surrogates are capable of generating appealing class-contrastive (counterfactual) explanations [169], which are the foundation of human-centred explainability [106]. To compare this explainer type with a linear surrogate, accentuate the difference in their respective explanations and demonstrate the importance of selecting a good surrogate model, we explain a carefully selected instance from a two-dimensional toy data set.

For our investigation, we use the two moons data set, which is a synthetic collection of instances with a complex decision boundary intended for binary classification tasks. It is particularly suitable for this type of experiments as depending on which data point is chosen, the resulting explanations can be quite diverse. Figure 3.14 shows two different surrogates for explaining the instance marked with the black dot in this setting: (a) based on a linear model without an interpretable representation and (b) built upon a decision tree. It is clear that for complex decision boundaries a tree-based approach is superior in this particular case, where the dimensionality of the data is low and their density is high. In addition to better approximating the local decision boundary of the underlying probabilistic black-box random forest classifier, the



(a) Linear surrogate without an interpretable representation. If the classification threshold is set at 0.5, the yellow bar would be partially predicted as blue, therefore incorrectly classifying the upper-left part of the red cloud of points. The influence of the x-axis feature is  $-1.10$  and the influence of the y-axis feature is  $0.69$ . Since an interpretable representation is not used, these numbers are difficult to interpret beyond comparing their values in relation to each other.

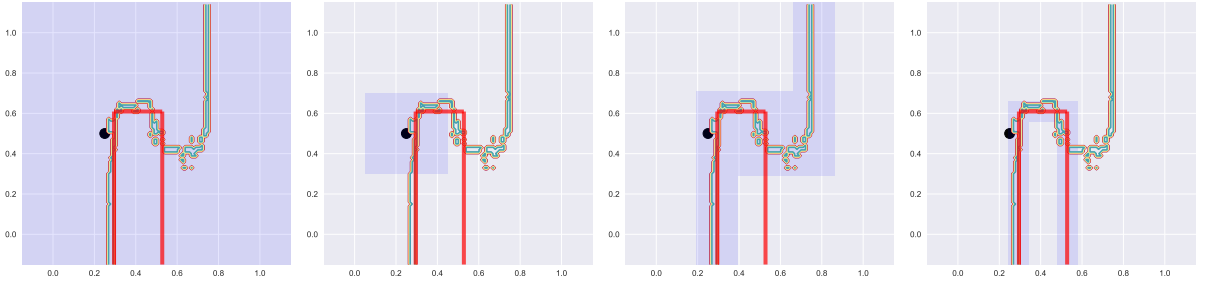
(b) Tree-based explainer for which the interpretable representation is the feature partition learnt by the surrogate itself. The green and light green background areas have high probability of the blue class, providing a good approximation of the fairly complex local decision boundary. The orange and yellow background blocks have low probability of the blue class, offering a precise approximation of the distribution of the red class. A possible explanation that can be derived from the local decision tree is: “Predict the blue class for the x-axis attribute  $\leq 0.3$  or the y-axis attribute  $> 0.6$ ; predict the red class for the y-axis feature  $\leq 0.6$  and the x-axis feature bounded between  $(0.3, 0.5]$ .” Notably, such rules can also be generated for high-dimensional data sets that cannot be easily visualised.

**Figure 3.14:** Comparison of linear and tree-based local surrogates for a toy tabular data set. The background shading represents the probability of the blue class predicted by the local (a) ridge regression and (b) regression tree surrogate models built to explain the instance marked with the black dot. The encoding of values predicted by these surrogates is given by the adjacent colour-bar. Since the output of a linear model (a) is unbounded, the predicted values may be outside of the expected  $[0, 1]$  range.

tree-based surrogate generates a locally-faithful interpretable representation from the feature splits that it learns. This can be particularly useful for high-dimensional tabular data, for which explanation visualisations are impossible but rule-based explanations consisting of a conjunction of logical conditions extracted from root-to-leaf paths are still viable. While this section (and the chapter as a whole) focuses predominantly on explaining tabular data, Chapter 5 is devoted to surrogate explainers of images (and by extension text) based on multi-output regression trees. To this end, explainability of decision trees as a standalone model is discussed in Chapter 4.

### 3.4.3 Evaluating Surrogate Explainers

Two of the main barriers for a wider adoption of surrogates are the instability and low fidelity of their explanations [133]. The former can be reduced by following best practices for building interpretable representations and sampling data, which are outlined in Sections 3.3.2 and 3.3.3. The fidelity of surrogate explanations, on the other hand, can be improved by composing bespoke algorithms tailored to the problem at hand and ensuring that each individual building block



(a) Global faithfulness of a surrogate model. (b) Local faithfulness of a surrogate model. (c) Faithfulness of the global decision boundary approximation. (d) Faithfulness of the local decision boundary approximation.

**Figure 3.15:** Various approaches to quantitative evaluation of surrogate models based on their faithfulness with respect to the underlying black box are possible. These metrics determine the ability of the surrogate (the red lines in the panels above) to mimic the predictions of the explained model (the green contours) by measuring its *fidelity* within a certain region. We can either compute *global* (a&c) or *local* (b&d) faithfulness with respect to the location of the *explained instance* (a&b) or the (*closest*) *decision boundary* (c&d).

performs as expected based on the proposed metrics: sample diversity and discretisation purity for tabular data, and segmentation granularity and colouring strategy for images. Nonetheless, to better understand whether such explanations can be trusted, the fidelity of the surrogate model should be measured with respect to the underlying black box. The evaluation metric used to this end depends on the type of both the black box and the surrogate, but usually it is a variation of the loss function given in Equation 3.2. For example, when both models are crisp classifiers, *predictive accuracy* can be used; when they output numbers, *squared error* is a good choice; and when the black box is a crisp classifier and the surrogate is probabilistic, *log-loss* is an option. Evaluating surrogates based on the XAI taxonomy introduced in Chapter 2 is also possible – it can provide a holistic view on the explainer and its functionality.

Depending on the intended use of a surrogate explainer, different strategies can be taken to measure its fidelity since its objective may either be to locally or globally approximate the black box decision boundary, or instead be faithful to a local or global data sample. These approaches are either data-driven and measured based on the scope of a relevant data sample (Figures 3.15a and 3.15b), or they are model-driven and determined by the shape of a decision boundary (Figures 3.15c and 3.15d). Satisfying each of these four competing objectives may require a unique surrogate whose explanations will only be truthful for the specific type of approximation:

**global data-driven** – Figure 3.15a – measures how well the surrogate performs in the entire data space;

**local data-driven** – Figure 3.15b – measures how well the surrogate performs in the vicinity of the explained instance;

**global model-driven** – Figure 3.15c – measures how well the surrogate can mimic the black-box model globally; and

**local model-driven** – Figure 3.15d – measures how well the surrogate approximates the black box locally.

Notably, the two local variants can be at odds if the closest black-box decision boundary is far from the explained data point, which is likely in high-dimensional spaces due to the curse of dimensionality. A high fidelity score is a sign of a well-crafted surrogate that is fit for purpose, thus engendering trust in explainees.

### 3.5 Perspectives on Surrogate Explainers

To better understand the role of surrogate explainers in XAI and IML, we briefly review relevant literature. Many disciplines benefit from various types of (explanatory) surrogates, which can be used as fast and low-complexity approximators of processes and functions that are too difficult to model or too complicated to comprehend [125]. In machine learning, Craven and Shavlik [28] proposed TREPAN, which is an algorithm for explaining neural networks by extracting their intelligible symbolic representation with a global surrogate tree. A similar approach, called RuleFit, was proposed by Friedman et al. [45], who used a surrogate linear model to assign importance to logical rules extracted from root-to-leaf paths of a (locally fitted) random forest. Friedman et al.’s idea to treat logical rules as explanatory artefacts can be thought of as a very early example of an interpretable representation (nonetheless not labelled with this name) that is limited to tabular data. Ribeiro et al. [129] then improved upon these concepts and offered a unified local surrogate explainer of black-box predictions for tabular, image and text data called LIME – it builds a local sparse linear model to explain human-intelligible concepts encoded by an interpretable representation, which allows explaining diverse data types.

When examining the individual components of surrogate explainers, choosing the interpretable representation and surrogate model for our analysis was motivated by the popularity of these particular approaches in the literature. Namely, LIME [129] and RuleFit [45] use a surrogate *linear model* to estimate importance or influence of interpretable concepts. Additionally, LIME and SHAP [100] employ an interpretable representation that *encodes presence and absence* of intelligible concepts to formulate their explanations. Similarly, RuleFit automatically learns a more complex IR by training a random forest and extracting rules from therein, which are then treated as binary concepts whose importance is determined by coefficients of a linear model, thus improving the expressiveness of discretisation-based interpretable representations. More recently, Garreau and Luxburg [46] analysed theoretical properties and parameterisation of vanilla LIME for tabular data, including its interpretable representation and surrogate linear model, however their work treated the explainer as an end-to-end algorithm and operated under quite restrictive assumptions, e.g., linearity of the black-box model.

Other (mostly empirical) research on surrogate explainability investigates the occasional instability of LIME explanations [87, 183], however it does not pinpoint the root causes of these undesired glitches. Lakkaraju and Bastani [84] show experimentally that surrogate explainers, such as LIME, can be tricked into producing misleading explanations by modelling the difference between the distribution of the explained data and the local sample used to train the explainer. Nonetheless, they do not thoroughly examine the affected explainers to understand the origin of such behaviour, which in case of LIME is caused by sampling tabular data from the discrete representation, making the distributions of the two sets relatively distinct. Similarly, Zhang et al. [183] show empirical evidence of LIME’s unexpected incoherence in certain cases, however they do not explore the origin of these issues, which based on our investigation are the artefact of the official LIME implementation and not the explainer per se (again, sampling from the discrete representation). Such observations prompted Laugel et al. [89] to propose minor modifications to the LIME algorithm, which have positive effects on the quality of its explanations but are limited to tabular data. Moreover, these alterations unintentionally compromise the integrity of LIME, making the two methods incomparable and the improvements not applicable to more general cases beyond the specific ones presented in Laugel et al.’s research.

Another relevant area of research aims to consolidate various explainers within a single schema, an abstract example of which is the XAI taxonomy presented in Chapter 2. Similarly, Henin and Le Métayer [58] introduced a unified (theoretical) framework that allows for systematic comparison of black-box explainers by characterising them along two dimensions: *data sampling* and *explanation generation*. In contrast, our (practical) approach is focused on algorithmic and implementation aspects of the subset of black-box explainers that operate as *surrogates*. Our meta-algorithm extends Henin and Le Métayer’s decomposition with a third dimension: *interpretable representation*, allowing our framework to be applied to a broader spectrum of data types. It also offers an in-depth analysis of individual surrogate building blocks and examines issues with their parings, providing a practical guideline for composing bespoke explainers. In particular, such an approach allows bLIMEy to bridge the gap between LIME and the family of surrogate explainers, thus unleashing their full potential.

## 3.6 If You Were to Choose One

In this chapter we introduced bLIMEy: a modular meta-algorithm for composing bespoke surrogate explainers of black-box models and their predictions. The bLIMEy framework consists of three building blocks – interpretable representation, data sampling and explanation generation – offering a range of algorithmic choices for each individual component. We analysed these modules both independently and when joined together as an end-to-end explainer to understand their properties, strengths and weaknesses. The results of our investigation provide guidance on choosing building blocks that are suitable for the problem at hand, which helps to construct the

best explainer that the surrogate family has to offer by avoiding common pitfalls. bLIMEy is accompanied by an open source implementation distributed within the FAT Forensics [159, 160] Python package (cf. Appendix B), which includes a selection of algorithms for every module of the meta-algorithm, therefore empowering the community to build tailored surrogate explainers.

Among others, our findings show the importance of building semantically and computationally meaningful interpretable representations, and their role in defining the question answered by the resulting explanations. We demonstrated that generic algorithms for building IRs may be insufficient, and that the intended application domain and audience, as well as interactive customisation and personalisation, should be considered. In particular, we discussed a popular operationalisation of IRs for image, text and tabular data, where they are used as binary indicators of presence and absence of interpretable concepts. This framework is then used in conjunction with surrogate models to quantify influence of such concepts on individual black-box predictions. In this setting, we identified challenges such as information removal proxies, parameterisation, faithfulness and determinism (of interpretable representations) – which are particularly prominent for tabular data – and explained how to overcome them.

Moreover, we showed that the choice of the data sampling algorithm and the representation in which this procedure is performed depend on the type of data. For relatively small binary interpretable representations that are often used with images and text, it is usually advantageous to generate a complete sample instead of employing a sampling method. When it comes to tabular data, however, it is more beneficial to randomly sample from the original data domain with an explicitly local and class-aware algorithm. Next, we analysed properties of different types of surrogate models and showed how their inherent constraints influence the resulting explanations. This investigation has not resulted in any clear selection guidelines, however we provided evidence that certain module pairings may unintentionally hurt the explainer. In particular, we demonstrated the limitations of explaining binary interpretable representations of tabular data with linear models and suggested logical models (such as decision trees) as a viable alternative. While throughout this chapter we encouraged the users to evaluate all modules individually, we also discussed various conceptual approaches to measuring fidelity of end-to-end surrogates, each one serving a different purpose.

Since bLIMEy consists of a conceptual framework for building surrogate explainers and a corresponding meta-algorithm, its utility lies entirely in the XAI process that it enables. The main message underlying our approach is that bespoke explainers must be tailored to the problem at hand despite such techniques being post-hoc, model-agnostic and data-universal. In particular, being able to build a superior surrogate explainer for one data set gives no precedent for the same strategy performing comparably well for similar data, let alone a completely distinct task. Therefore, showing a qualitative or quantitative dominance of a single realisation of bLIMEy on a collection of benchmarks with a chosen evaluation strategy would be counterproductive and conflict with the idea of championing tailor-made surrogates. All things considered, the

insights presented in this chapter clearly highlight the promises and perils of using out-of-the-box, one-size-fits-all, silver-bullet approaches such as surrogate explainers without much afterthought.

While a panacea is nowhere to be found, throughout our investigation we noted a pattern suggesting that decision trees are a good candidate for the surrogate model. We observed that they neither impose feature independence nor a linear relation between the intelligible concepts and the explained quantity, which are essential for high-level interpretable representations. For tabular data, they can combine two of the surrogate building blocks – creation of interpretable representations and generation of explanations – in an arguably optimal way since decision trees strive for leaf purity, which translates into a faithful approximation of the black-box decision boundary. Moreover, they are compatible with a diverse range of black-box models, supporting binary and multi-class classification as well as classic and multi-output regression. While they offer a wide range of meaningful and appealing explanatory artefacts – a big advantage over linear models, which are restricted to (interpretable) feature importance or influence – we need to be able to reliably extract them to utilise the aforementioned advantages. The next chapter addresses this problem since the inherent transparency of decision trees may not always be sufficient to warrant their explainability, for example, deep and wide trees are transparent but not explainable.



## CTREEX: CONTRASTIVE EXPLANATIONS FOR DECISION TREES

Having seen the wide-reaching benefits of classification and regression trees when used as surrogate explainers, in this chapter we investigate shortcomings of their *inherent* interpretability. Decision trees are often presented as a prime example of transparent machine learning models, in contrast to, for example, black-box deep neural networks. They classify a data point by following a hierarchy of logical conditions applied to its features, which can be mechanistically simulated by humans. This argument is often used to label them transparent, interpretable and explainable, which shifts the focus of the XAI research community away from this family of models. Replicating their decisive process *in vivo*, however, does not necessarily lead to understanding, which, as we argued in Chapter 1, is required for explainability. While indeed transparent, large trees are neither particularly comprehensible nor explainable using their inherent explanations – such as visualisations of the tree structure and logical rules extracted from root-to-leaf paths – whose complexity grows proportionally to the model size. We address these challenges with **CtreeX**: an algorithm for generating *class-contrastive* and supportive explanations of decision tree predictions, which are commended in the literature for their appeal, brevity and actionability. Our explanations outline a succinct reason behind a prediction instead of listing the underlying chain of logical conditions, thereby decoupling the complexity of the explanation from the tree size. In Appendix A.2, we analyse and describe our method in form of a Fact Sheet based on the XAI taxonomy introduced in Chapter 2.

### 4.1 Decision Trees: Transparent but Not Explainable

Decision Trees (DTs) [23] are efficient, flexible and transparent machine learning models, making them a popular choice. These algorithms are considered glass-box predictors mainly because they *sequentially* partition the *raw input feature space*, thus narrowing down the region of interest.

This easy-to-trace procedure allows one to inspect the logical conditions leading to a particular prediction and to simulate this *in silico* decision process *in vivo*. Simulatability [96] makes them especially appealing for high-stakes applications where transparency is of paramount importance or simply a legal requirement. This property, however, does not ascertain interpretability or explainability in all cases – a crucial distinction outlined in Section 1.1.3. Manually replicating the decision process of a tree does not warrant any understanding of the reason behind its predictions, which follows from The Chinese Room Argument [139].

Decision trees, as a model, are commonly “explained” by visualising their structure; their predictions, on the other hand, are justified with lists of logical conditions extracted from relevant root-to-leaf paths. However, as data grow in size – both in the number of instances and features – the trees and their explanations can become overwhelmingly complex, thereby rendering these transparent models and their decision logic incomprehensible. As the depth of a tree increases, the number of logical conditions that must be satisfied to classify a data point increases as well. When used as an “explanation”, such a long decision trace only tells the user that all the listed logical conditions are responsible for a prediction, without attributing it to any individual one in particular. This large amount of information combined with a lack of domain expertise may undermine the user’s ability to narrow down the reason behind the prediction to a subset of these predicates without an in-depth analysis. Therefore, we argue that inspecting the internal structure of decision trees in an effort to trace and mimic their decision logic for a particular prediction is insufficient for their explainability. Making sense of such information requires **logical reasoning** – exercised either by the explainee (human) or the explainer (machine) – to achieve *proper* interpretability and explainability built on top of the inherent transparency of decision trees; for example, the splitting criteria learnt by a tree can be processed to generate easy-to-digest explanations.

While the transparency of DTs does not imply their interpretability, we can leverage access to their internal logical structure to compose insightful and appealing statements. To this end, we investigate *contrastive* and *supportive* explanations, which are known for their human appeal grounded in social science research [106] and compliance with various legal requirements [173] including the “right to explanation” proposed by, but ultimately excluded from, the European Union’s General Data Protection Regulation [50, 172]. The most popular and informative variant of the former – *class-contrastive* explanations also known as *counterfactuals* – adheres to the following template:

“The prediction is <prediction>. Had a small subset of features been different <foil>, the prediction would have been <contrastive prediction> instead.”

Supportive explanations, on the other hand, prescribe, possibly one of many, minimal sets of requirements – i.e., logical conditions – *sufficient* for a particular prediction to hold. Their phrasing tends to conform to the following pattern:

“The prediction is <prediction>. Had *any* of the following conditions not been satisfied <support>, the prediction would have changed.”

These two explanation types respectively answer “Why not?” and “Why?” a predictive model behaved in a certain way, both of which have been shown to enhance explainee’s understanding, trust and acceptance of an intelligent system [94].

A straightforward class-contrastive explanation of a tree prediction – extraction of which does not require extensive processing of the tree structure – is a counterfactual based on the neighbouring leaf. It prescribes a change of just one feature value – captured by the last tree split – that results in a different prediction. A more principled approach to generating counterfactuals from the tree structure could nonetheless allow to customise them by imposing (user-defined) restrictions and requirements, thereby forcing some of the features and preventing others from appearing in such explanations, e.g., conditioning on a person’s age – which is non-actionable – can be avoided. Such (personalised) explanations can empower explainees without domain expertise to obtain an intelligible list of actions, helping them to *understand* predictions made by the tree and *guiding* them towards achieving a desired outcome. In particular, class-contrastive explanations are sparse, prescriptive and communicate the smallest change that results in a different prediction. The last property is, arguably, a simple form of logical reasoning that warrants the explanatory power of counterfactuals – a drastic improvement over the inherent transparency of decision trees that only enables listing a (possibly large) collection of logical conditions leading to a particular prediction. Notably, supportive explanations directly improve upon such an exhaustive list by reducing its size and complexity to only include *sufficient* conditions, thereby making such explanations more general, concise and comprehensible.

Because of their apparent transparency, research on interpretability of classification and regression trees, and more broadly logical ML models and their ensembles, receives relatively little attention. Nonetheless, an explainability suite tailored to decision trees can help practitioners to debug such models, discover their biases and identify their unfair behaviour (e.g., via counterfactual disparate treatment), in addition to explaining them and their predictions. The inherent transparency of decision trees facilitates a diverse range of model and prediction interpretability:

- visualisation of the tree structure;
- tree-based feature importance;
- conjunction of logical conditions extracted from a root-to-leaf path;
- exemplar explanations sourced from training data assigned to a single leaf;
- answers to what-if questions generated either based on the tree structure or by querying the model;

- **class-contrastive** explanations (counterfactuals) retrieved by applying logical reasoning to compare different tree paths; and
- **supportive** explanations composed by generalising logical conditions imposed on a root-to-leaf path.

Each explanation type serves a different purpose and answers a different question, with the first two concerning *model* transparency and the remaining five improving the interpretability of *predictions*. However, based on our distinction between *transparency* and *explainability* introduced in Chapter 1, the first five (discussed in Section 4.2) can only improve *transparency* of decision trees, whereas the final two (investigated in Section 4.3) are capable of *explaining* their predictions.

In this chapter, we propose a novel algorithm called **Contrastive tree eXplainer (CtreeX)** designed to generate *class-contrastive* and *supportive* explanations of decision tree predictions to improve their interpretability and comprehensibility by decoupling the tree size from the explanation length. We advocate a model-specific approach to take advantage of all the theoretical capabilities of such explanations [106], which can be easily lost in a more generic, model-agnostic setting. To this end, we exploit access to the internal structure of a decision tree, which allows us to extract a collection of logical conditions for each of its decision leaves. Next, we collate them and create a meta-feature space in which each leaf is described with applicable logical conditions. We use this new representation to compute a leaf-to-leaf distance matrix between every pair of leaves, with the underlying metric measuring the *number* of features whose values need to be changed to jump to another leaf. By tuning the distance function, the user can control which features appear in the explanation and which ones should be avoided. Moreover, it can account for the perceived importance of each attribute and, possibly, quantify the magnitude of the proposed changes for numerical features. We describe the algorithm, its guarantees and properties such as retrieval efficiency, customisability and completeness of generated explanations in Sections 4.3 and 4.4. Additionally, in Appendix A.2 we present a CtreeX Fact Sheet based on our XAI taxonomy (Chapter 2).

Our approach avoids a naïve search through the entire feature space when retrieving an explanation by representing all the meaningful attribute tweaks in a meta-feature distance matrix, thereby making the algorithm computationally feasible for any tree size. Furthermore, the length of the resulting class-contrastive and supportive statements is decoupled from the size of the underlying model, which in turn guarantees succinct and appealing explanations. They provide a (lay) explainees with a sparse rationale behind a particular prediction, which naturally leads to understanding since these explanation types do not presuppose any background in computer science or artificial intelligence. While CtreeX is specific to decision trees, its benefits can be generalised to a model-agnostic setting by using a tree-based surrogate [142] built upon the bLIMEy framework [152, 158] – see Chapter 3 for more details. An example of such an explainer is the LIMETree [150] algorithm presented in Chapter 5, where we show its superiority

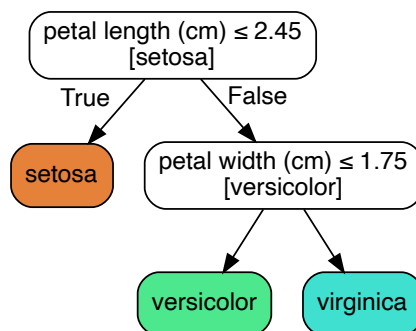
to the more established linear surrogate [129]. Moreover, our meta-feature representation creates an interoperable interface that allows to generalise our approach to other logical predictive models, such as rule lists and sets, and their ensembles, e.g., random forests.

Contrastive explanations are becoming the *de facto* standard for explaining automated decisions, mainly because of their aforementioned legal compliance [173] and social appeal [106]. While apt in theory, recent findings [123] have highlighted issues with algorithms used for their generation, as these approaches tend to produce impractical and non-actionable explanations – e.g., suggesting to change one’s age – or retrieve them from regions of low data density, making them volatile or impossible to achieve in the real life. Notably, our method is not susceptible to such problems since the class-contrastive explanations are extracted directly from the predictive model – an advantage of a tree-specific (ante-hoc) approach. Similar algorithms for generating counterfactuals from decision trees have been proposed in the literature [166, 169], however our method provides more flexibility, outputs a complete set of feasible explanations and supports their comprehensive customisation. Moreover, our class-contrastive statements are complemented by supportive explanations, which resemble the output of a model-agnostic explainer called *anchors* [130]. At the expense of being model-specific, Ctrees guarantees full fidelity of the supportive (and class-contrastive) statements, which is highly desirable [133]. Section 4.5 – which reviews research related to decision tree interpretability – discusses all of these algorithms in more detail.

## 4.2 Inherent Transparency of Decision Trees

Decision trees are a family of predictive models that is *inherently transparent*. This property allows us to inspect their internal structure and process it either manually or algorithmically. It also facilitates a variety of interpretability approaches, each one providing unique insights that serve a specific purpose and have a distinct set of limitations. Nonetheless, they share the same explanatory mechanism: the users drive the investigation and it is up to them to *understand* the behaviour of a tree and its predictions based on “manual” examination of the evidence. This delegation of processing (often technical) insights in search of explanations usually requires in-depth technical knowledge of decision trees, domain expertise and a well defined question, thus narrowing down the user base of these methods. In this section, we discuss the five most prominent types of insights stemming from the inherent transparency of decision trees, namely: tree structure visualisation, tree-based feature importance, root-to-leaf paths, exemplars and what-if statements. For each of them, we provide an example explanation extracted from a classification tree learnt on the popular Iris data set [41].

**Tree Structure Visualisation** is a *model* inspection approach in which the structure of a decision tree is plotted as a graph – see Figure 4.1. To benefit from the information represented therein, the explainee needs to understand the theory behind decision trees, therefore it may



**Figure 4.1:** Visualisation of a classification tree trained on the Iris data set. It is an example of a decision tree-specific transparency approach that helps the explainee to comprehend the (global) behaviour of such models. Its complexity grows with the size of the tree and it may not be suitable for a lay audience who lacks relevant machine learning knowledge.

be unintelligible to a lay audience. The visualisation complexity increases with the size of the tree, therefore tracing a prediction from the root to a leaf becomes challenging for large models. This issue can be partially overcome by making the visualisation interactive, however multiple conditions applied to the same feature may be scattered throughout a root-to-leaf path, thus concealing the bigger picture.

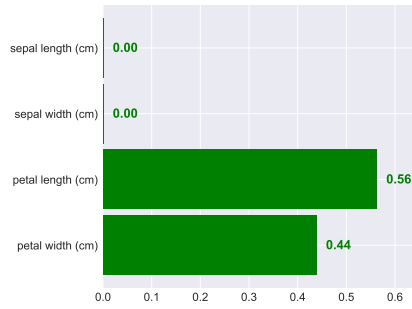
**Feature Importance** is another *model* inspection technique that communicates to the explainee which features are the most informative (in separating the classes) when modelling the underlying data – see Figure 4.2 for an example. It is relatively compact in comparison to a tree structure visualisation because its size is limited by the number of attributes in a data set. Interpreting feature importance does not require technical knowledge, making it more appealing to a lay audience at the expense of being less insightful and informative. Importance  $I(f_n)$  of the  $n^{\text{th}}$  feature  $f_n$  is computed as the sum of importance  $i(s)$  of every node  $s \in \mathcal{S}_{f_n}$  splitting on this feature divided by the total importance of every node  $s \in \mathcal{S}$  in the tree, i.e:

$$I(f_n) = \frac{\sum_{s \in \mathcal{S}_{f_n}} i(s)}{\sum_{s \in \mathcal{S}} i(s)}.$$

Importance  $i(s)$  of a splitting node  $s$  is calculated as *information gain* – the decrease in the node’s impurity  $\mathcal{L}$  (in relation to its children) weighted by the probability of reaching this node, which is often determined by the number of samples that reach the node divided by the total number of samples, i.e.:

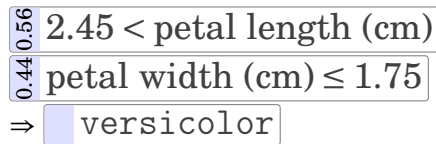
$$i(s) = \frac{|s|}{|X|} \mathcal{L}(s) - \frac{|s_{\text{left}}|}{|X|} \mathcal{L}(s_{\text{left}}) - \frac{|s_{\text{right}}|}{|X|} \mathcal{L}(s_{\text{right}}).$$

In this formulation  $|s|$  is the number of instances reaching the splitting node  $s$  and  $|X|$  is the total number of instances in the data set  $X$ . The definition of  $\mathcal{L}$  depends on the type of the tree: classification trees can use the Gini impurity or entropy, and regression trees can use the mean squared error.

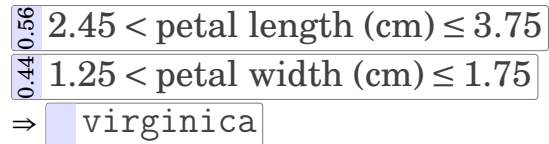


**Figure 4.2:** Bar plot depicting feature importance extracted from a classification tree trained on the Iris data set. It is an example of a decision tree-specific transparency approach that helps the explainee to comprehend the overall (global) importance of data attributes. In many cases, interpreting the plot only requires being familiar with the meaning of the underlying features.

**Logical Rules** are used to inspect *predictions* of a tree by extracting a conjunction of logical conditions from the corresponding root-to-leaf paths – see Figure 4.3 for an example. They may be suitable for lay explainees who understand the meaning of the underlying data attributes, however they can mislead the recipients into thinking that all of the conditions are equally important if their visualisation does not convey this information (i.e., feature importance). The length of logical rules can grow proportionally to the size of a tree, however, in contrast to tree structure visualisations, they can be compressed by merging multiple splits applied to the same attribute (cf. Figure 4.3b). This grouping approach improves their readability at the expense of forfeiting the original order of logical conditions, which implicitly indicates the significance of each split and its corresponding feature. This concern can be partially addressed by presenting *feature importance* beside logical rules – see the vertical numbers to the left of each box in Figure 4.3.

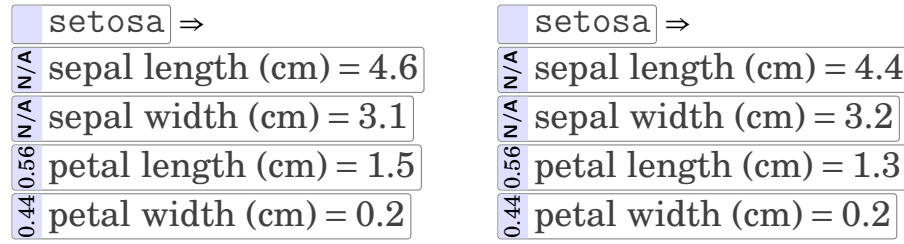


(a) Logical rule for the *versicolor* class extracted from the classification tree shown in Figure 4.1.



(b) Fictitious logical rule for the *virginica* class showing a simplistic compression of root-to-leaf conditions.

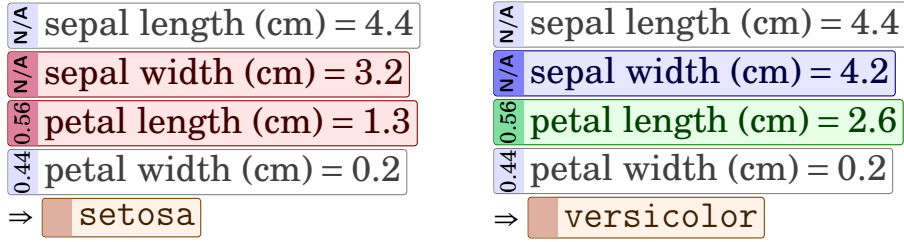
**Figure 4.3:** Visualisation of logical rules – presented as a conjunction of logical conditions – extracted from root-to-leaf paths of a classification tree trained on the Iris data set. It is an example of a decision tree-specific transparency approach that helps the explainee to understand conditions imposed on data attributes that lead to a particular prediction (a local or cohort explanation). In many cases, interpreting the figure only requires being familiar with the meaning of the underlying features. The vertical number to the left of each logical condition reports the importance of the corresponding feature.



**Figure 4.4:** Visualisation of two *setosa* class exemplars extracted from the training data assigned to a selected leaf of a classification tree fitted to the Iris data set. It is an example of a decision tree-specific transparency approach that helps the explainee to understand similarities between instances grouped together (within a single leaf) by the underlying tree (a local or cohort explanation). In many cases, interpreting the figure only requires being familiar with the meaning of the data features, however it is up to the explainee to reason about the connections between the output instances. Since some of the features may be redundant – i.e., not conditioned on the root-to-leaf path leading to the selected leaf – they should be marked as such (indicated with *N/A* in this figure) to avoid misleading the explainee.

**Exemplars** can be used to inspect a *prediction* of a tree by presenting the explainee with (training) data assigned to the same leaf as the explained instance – see Figure 4.4 for an example. When the model is overfitted, this transparency approach can become ineffective since each leaf may be built upon a single data point. Showing data that, according to the underlying tree, are related to the instance in question requires the explainee to reason about their similarities and differences, which may render this approach inappropriate for a lay audience. Exemplars have a size equivalent to the number of features in the training data set and they can carry redundant information since not all of the attributes may be conditioned upon on a given root-to-leaf path (cf. Figure 4.3a). Depending on the user’s expectations and privacy constraints, these explanations can either be based on real training data points or synthetic instances (which may be unrealistic). Exemplars and *logical rules* convey similar information – all of the exemplars comply with the logical conditions of the corresponding root-to-leaf path; the main advantage of the former is clear indication of feature values that are feasible in the real world as long as these explanations are based on training data.

**What-ifs** are both a *model* and a *prediction* inspection mechanism – see Figure 4.5 for an example. They are user-driven – the explainee defines the foil of a what-if question – and their effective use may require the explainee to have a search agenda or heuristic, which depends on the user’s understanding of the underlying problem and general structure of tree-based models. Nonetheless, their presentation is sparse and informative making them appealing to a lay audience, especially that they resemble class-contrastive (counterfactual) explanations. In case of decision trees, computing them would not benefit from a dedicated algorithm since the predictive function of trees has linear computational complexity with respect to the tree depth.



**Figure 4.5:** Visualisation of a what-if explanation extracted from a classification tree trained on the Iris data set. The left part of the plot depicts an instance selected to be explained, which is classified as *setosa*. The data point to the right has two of its attribute values modified by an explaine (red and blue/green shading), thus posing a what-if question leading to a *versicolor* prediction. This (local) explanation is an example of a decision tree-specific transparency approach that helps the explaine to understand influence of selected feature values on a tree prediction. In many cases, being familiar with the meaning of the data features is sufficient to interpret the figure. Often, only the differentiating factors are highlighted, i.e., the foil, to make the explanation sparse, however this example lists all of the features given their low number. Additionally, the visualisation indicates which feature value changes proposed by the user are meaningful – green shading – given that some attributes may not appear on the root-to-leaf path responsible for predicting the what-if instance or be used by the tree altogether (captured by blue shading and N/A markers to the left of each box). The change of prediction is shown by orange tint.

### 4.3 Tree-based Class-contrastive and Supportive Explanations

All of the transparency approaches discussed in the previous section place the explanatory burden on the user, who is required to *reason* about various model and prediction insights to elicit understanding. However, we can improve upon this procedure and make it more attractive (especially to non-technical users) by delegating the information processing responsibility to an algorithm while simultaneously preserving the explaine’s investigative power and control over the explanatory process via meaningful interactions. To this end, we propose CtreeX: a theoretically sound algorithm for generating *supportive* and *class-contrastive* (counterfactual) explanations of decision tree predictions. We take advantage of access to the internal structure of trees, which allows us to extract logical conditions leading to each decision leaf. We demonstrate how to process this information to construct a meta-representation of a tree in which each leaf is assigned a collection of meta-features that are based upon logical conditions extracted from the splitting nodes corresponding to its root-to-leaf path. This new encoding of the tree structure allows us to measure similarity of its leaves determined by the number of attribute values that need to be altered to jump to another leaf (of a different class). Additionally, it can account for other factors such as (user-provided) feature preference and quantification of tweaks applied to attribute values. Our approach avoids a naïve search through the entire feature space, making the explanation generation process computationally feasible regardless of the tree size.

When humans receive a decision supported by a collection of logical conditions, they often skip further analysis unless the outcome disagrees with their expectations or mental model [80, 106].

In such cases two types of *reasoning* are commonly used to help identify factors that influenced the automated decision the most:

**by example** where we identify similar instances and draw analogies to support our expectations; and

**by contrast** where, to argue our case, we create hypothetical conditions under which the desired outcome is achieved.

These two approaches are firmly grounded in the social sciences, where researchers have studied the human explanatory process for decades [106] – see Section 1.2.3 for more details about the origin and benefits of such explanations. The first type is based on *similarities*, e.g., exemplar transparency, and is the motivation for our *supportive* explanations, whereas the latter builds upon *differences*, e.g., what-if transparency, and is the foundation of our *contrastive* explanations. The second type can further be divided into [106]:

**class-contrastive** identifying a *similar instance* (a few attribute value changes) that yields a *different prediction*; and

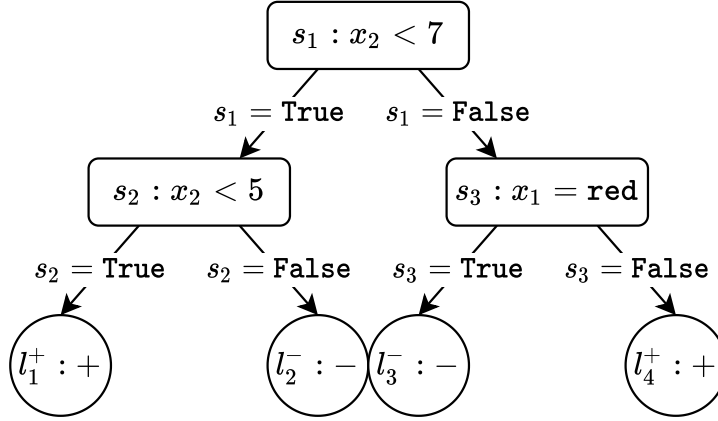
**instance-contrastive** determining a *different data point* (many attribute value changes) that yields *the same prediction* – it is similar to reasoning by example.

CtreeX focuses on the first type, i.e., class-contrastive (counterfactual) explanations.

### 4.3.1 Meta-feature Tree Representation

Assume a decision tree  $t : \mathcal{X} \rightarrow \mathcal{Y}$  of depth  $d$  fitted to a data space  $\mathcal{X}$  and its corresponding binary label space  $\mathcal{Y} = \{+, -\}$ . We denote the splitting nodes of the tree as  $\mathcal{S} = \{s_1, \dots, s_m\}$ , where  $m$  is the number of unique splits. We further denote its leaves as  $\mathcal{L} = \{l_1, \dots, l_w\}$ , where  $w$  is the tree width; additionally, we define function  $t_l : \mathcal{X} \rightarrow \mathcal{L}$  that assigns a leaf to a data point based on the underlying tree  $t$ . In this setting, each leaf  $l_i \in \mathcal{L}$  is uniquely identified by a root-to-leaf path of length  $n$  (with  $0 < n \leq d$ ) that can also be described with a collection of  $n$  logical conditions extracted by function  $t_p : \mathcal{L} \rightarrow \mathcal{S}^n$  from the tree splitting nodes. For convenience, we indicate the class  $\hat{y} \in \mathcal{Y}$  predicted by a leaf  $l_i$  in its superscript, e.g.,  $l_i^-$  if the  $i^{\text{th}}$  leaf predicts the negative class. Figure 4.6 depicts a toy tree with annotated splitting nodes and leaves.

In such a tree, the logical conditions  $\mathcal{S}$  create a **meta-feature space**  $\mathcal{B}$ ; this representation captures **all the non-overlapping attribute partitions** determined by the thresholds extracted from the tree splitting nodes. A meta-feature representation of a particular leaf is computed by function  $t_b : \mathcal{L} \rightarrow \mathcal{B}$  based on a collection of logical conditions extracted from the splitting nodes corresponding to its root-to-leaf path. The meta-features can take values  $b_i \in \{-1, 0, 1\}$  respectively for *failing* a logical condition, it being *undecided* (i.e., not used) or *satisfying* it; for reference see Table 4.1, which outlines the splits  $\mathcal{S}$ , meta-features  $\mathcal{B}$  and distance measurements



**Figure 4.6:** Balanced decision tree of depth  $d = 2$  with three splits  $s_i$  applied to two features  $x_i$  resulting in four leaves  $l_i$ . Feature  $x_1 \in \{\text{red}, \text{green}, \text{blue}\}$  is categorical, and feature  $x_2 \in \mathbb{Z}$  is numerical. Branching left corresponds to satisfying the split’s logical condition ( $s_i = \text{True}$ ), and branching right denotes failing it ( $s_i = \text{False}$ ).

between leaves for the decision tree shown in Figure 4.6. While we assume a two-class classification task and binary tree splits, our method generalises to multi-class classification (however, the user may need to explicitly specify the contrast class) and multi-way splitting nodes. It can also be applied to regression trees, where the contrast becomes predicting a different number.

For example, consider a data point  $\hat{x} \in \mathcal{X}$  that is assigned to the leaf  $t_l(\hat{x}) = l_3^-$  by the decision tree shown in Figure 4.6. This leaf can be represented with the logical conditions extracted from the applicable splitting nodes as:

$$l_3^- \equiv [7 \leq x_2 \wedge x_1 = \text{red}].$$

Alternatively, its root-to-leaf path can be given by the splitting nodes:

$$t_p(l_3^-) = [s_1 = \text{False}, s_2 = \text{Undecided}, s_3 = \text{True}],$$

	$s_1$	$s_2$	$s_3$	$b_{11}$	$b_{12}$	$b_{21}$	$b_{22}$	$b_{23}$	$l_1^+$	$l_4^+$	$l_2^-$	$l_3^-$
$l_1^+$	T	T	U	0	0	1	-1	-1	0	1	1	1
$l_4^+$	F	U	F	-1	1	-1	-1	1	1	0	1	1
$l_2^-$	T	F	U	0	0	-1	1	-1	1	1	0	1
$l_3^-$	F	U	T	1	-1	-1	-1	1	1	1	1	0

**Table 4.1:** Tree splits  $\mathcal{S}$  (left), meta-feature representation  $\mathcal{B}$  (middle) and pairwise leaf distance (right) for the decision tree given in Figure 4.6. The splits  $s_i \in \mathcal{S}$  are annotated as T for True, F for False and U for Undecided (i.e., not used/applicable). The meta-feature space built upon these splits is determined by  $\mathcal{B} = \{b_{11} : x_1 = \text{red}, b_{12} : x_1 = (\text{green} \vee \text{blue}), b_{21} : x_2 < 5, b_{22} : 5 \leq x_2 < 7, b_{23} : 7 \leq x_2\}$ . The right part of the table groups the leaves by their predicted class to highlight similarity of leaves with the same (diagonal quadrants) and opposite (off-diagonal) predictions.

where Undecided is used for splits that *do not appear* on (i.e., are not applicable to) a given path. Since a tree can split each numerical feature multiple times, we translate these thresholds into half-closed intervals; for example, splits  $s_1 : x_2 < 7$  and  $s_2 : x_2 < 5$  in Figure 4.6 partition feature  $x_2$  into  $b_{21} : x_2 < 5$ ,  $b_{22} : 5 \leq x_2 < 7$  and  $b_{23} : 7 \leq x_2$  bins in our meta-feature representation  $\mathcal{B}$ . Similarly, feature  $x_1$  (which is categorical) is partitioned into  $b_{11} : x_1 = \text{red}$  and  $b_{12} : x_1 = (\text{green} \vee \text{blue})$  by split  $s_3 : x_1 = \text{red}$ . Therefore, in the meta-feature space  $\mathcal{B}$  leaf  $l_3^-$  is represented as:

$$t_b(l_3^-) = [b_{11} = 1, b_{12} = -1, b_{21} = -1, b_{22} = -1, b_{23} = 1].$$

Notably, categorical features with *more than two* values may create disjunctions in the set of logical conditions for *binary* decision trees, e.g., feature  $x_1 \in \{\text{red}, \text{green}, \text{blue}\}$  and (binary) split  $s_3 : x_1 = \text{red}$  imply  $x_1 = (\text{green} \vee \text{blue})$  for leaf  $l_4^+$ . Had an extra categorical split  $s_e : x_1 = \text{blue}$  appeared in the tree, our meta-feature representation would be partitioned into  $\{\text{red}, \text{green}, \text{blue}\}$  instead of  $\{\text{red}, (\text{green} \vee \text{blue})\}$ , thus helping to resolve this issue. Similarly, two independent root-to-leaf paths can divide a single numerical feature into (partially) *overlapping* bins, e.g.,  $x_i < 42$  and  $24 \leq x_i < 88$ , which may be problematic. Such splits require special treatment when computing similarity between two affected leaves since a logical disjunction and overlapping numerical intervals are likely to cause the underlying distance metric to be ill-defined. Nonetheless, by *fixing a data point* prior to computing such a metric this issue can be easily resolved because the attribute values of the selected instance uniquely determine the meta-features – see Section 4.3.2 for more details.

### 4.3.2 Distance Metrics

Computing a distance between two arbitrary leaves  $l_i$  and  $l_j$  represented in the meta-feature space may not always be possible as noted in the previous section. Imagine  $(3 \leq x_m < 7) \in t_p(l_i)$  and  $(x_m < 7) \in t_p(l_j)$  for a numerical feature  $x_m$ , or  $(x_n = \text{red}) \in t_p(l_i)$  and  $(x_n = \text{red} \vee \text{blue}) \in t_p(l_j)$  for a categorical feature  $x_n$ . Without assuming an arbitrary heuristic, computing similarity between overlapping feature ranges or sets is an ill-defined procedure and as such a metric  $m : \mathcal{B} \times \mathcal{B} \rightarrow \mathbb{R}$  can become asymmetric, i.e.,  $m(t_b(l_i), t_b(l_j)) \neq m(t_b(l_j), t_b(l_i))$ . However, by specifying a reference data point  $x^*$  – which we want to explain – the meta-feature values become fixed, thereby grounding the logical conditions represented by  $\mathcal{S}$  and  $\mathcal{B}$ . This step allows us to compute a similarity or distance between any two leaves  $l_i$  and  $l_j$  as  $m(t_b(l_i), t_b(l_j); x^*)$ . For example, using the aforementioned root-to-leaf paths  $t_p(l_i)$  and  $t_p(l_j)$  represented in the meta-feature space as  $t_b(l_i)$  and  $t_b(l_j)$ , if  $x_n^* = \text{blue}$ , the corresponding splits are grounded with  $(x_n = \text{red}) = \text{False}$  and  $(x_n = \text{red} \vee \text{blue}) = \text{True}$ , thus making the metric symmetric, i.e.,  $m(t_b(l_i), t_b(l_j); x^*) = m(t_b(l_j), t_b(l_i); x^*)$ . Therefore, by choosing a data point – which the application domain requires anyway since we are explaining a prediction – a meaningful distance metric can be defined on the meta-feature representation of a tree.

To this end, we modify Hamming distance since it behaves similarly to the  $L_1$ -norm for binary vectors, thereby favouring sparsity. This property is desired when retrieving class-contrastive and supportive explanations as it ensures that the closest foils and furthest supports respectively differ in the fewest and overlap in the most attributes within the original feature space  $\mathcal{X}$ . The metric is extended to handle a representation that in addition to *satisfied* (1) and *unsatisfied* (−1) logical conditions also encodes meta-features that are *undecided* (0). Therefore, we define the distance between two arbitrary leaves  $l_m$  and  $l_n$  represented in the meta-feature space as  $b(m) = t_b(l_m)$  and  $b(n) = t_b(l_n)$ , and grounded with a data point  $x^*$  as:

$$m(b(m), b(n); x^*) = \sum_{i \in \mathcal{X}} \mathbb{1} \left( \sum_{j \in \mathcal{B}_i} \tilde{\mathbb{1}}(b_{ij}(m), b_{ij}(n)) \right), \quad (4.1)$$

where the first indexing  $i \in \mathcal{X}$  chooses a feature in the original data space and the second  $j \in \mathcal{B}_i$  iterates through all of the meta-features pertaining to this attribute – see the  $b_{ij}$  indexing in Table 4.1 for reference. In this equation, the indicator function  $\mathbb{1}$  is defined as:

$$\mathbb{1}(x) = \begin{cases} 1 & \text{for } 0 < x, \\ 0 & \text{otherwise;} \end{cases}$$

and the modified indicator function  $\tilde{\mathbb{1}}$  is defined as:

$$\tilde{\mathbb{1}}(x, y) = \begin{cases} 1 & \text{for } (x, y) \in \{(-1, 1), (1, -1)\}, \\ 0 & \text{otherwise.} \end{cases}$$

Recall that 0 in the meta-feature space denotes a logical condition that is undecided for (i.e., not applicable to) a given root-to-leaf path. Therefore, the latter formulation can be understood as returning 1 if such a condition is satisfied for one leaf and failed for the other, and 0 if it yields the same result on both branches or is undecided for one of them regardless of the outcome for the other. Therefore, the distance function defined in Equation 4.1 reaches its maximum when two leaves differ in the most number of features when represented in the original data domain  $\mathcal{X}$ , and it is at its minimum for similar leaves; note that the smallest distance is 1 since any two leaves must at least differ in the split imposed at the root of a tree.

Our metric quantifies the *number of tweaks* required in the original feature space  $\mathcal{X}$  but it does not consider the magnitude of these changes, therefore it prefers shorter, more comprehensible counterfactuals, and more general supportive explanations. It is a *pseudo-metric* in theory as points in this space are not necessarily distinguishable, i.e.,  $m(t_b(l_m), t_b(l_n); x^*) = 0$  for some  $t_b(l_m) \neq t_b(l_n)$ , for example,  $t_b(l_m) = (-1, 1)$  and  $t_b(l_n) = (0, 0)$ . In practice, however, such data points *cannot* exist as they are exclusively determined by the structure of a tree; for any  $t_b(l_m) \in \mathcal{B}$  and  $t_b(l_n) \in \mathcal{B}$ , the distance  $m(t_b(l_m), t_b(l_n); x^*) = 0$  for all  $t_b(l_m) = t_b(l_n)$  and  $m(t_b(l_m), t_b(l_n); x^*) > 0$  for all  $t_b(l_m) \neq t_b(l_n)$ . Table 4.1 lists distances computed according to this metric for the decision tree shown in Figure 4.6; note that in this case we can calculate the leaf-to-leaf metric without grounding it with an (explained) data point as the corresponding feature

partitions are non-overlapping. Alternative distance formulations (discussed in Section 4.4) can, for example, take into account user preferences or the importance of each feature as determined by the tree.

### 4.3.3 Explanation Generation

**Class-contrastive Explanations** Given a leaf-to-leaf distance matrix computed for a tree  $t$  and grounded with a data point chosen to be explained  $x^*$ , we generate class-contrastive (counterfactual) explanations by *minimising* the following objective:

$$l_{\min} = \underset{l_i \in \mathcal{L}}{\operatorname{argmin}} m(t_b(t_l(x^*)), t_b(l_i); x^*) \text{ for } t(x^*) \neq \text{class}(l_i).$$

To this end, given a data point  $x^*$  to be explained, we find its meta-feature representation  $b^* = t_b(l^*)$  based on the leaf  $l^* = t_l(x^*)$  assigned to it by the decision tree  $t$ . Next, we compute its similarity  $m(b^*, b_i)$  to all the other tree leaves  $l_i$  – represented in the meta-feature space as  $b_i = t_b(l_i)$  – that yield a different prediction class( $l_i$ ) than the one assigned by our tree  $t$  to the explained instance  $x^*$ , i.e.,  $t(x^*) \neq \text{class}(l_i)$ , thus identifying a leaf  $l_{\min}$  that minimises our distance. To derive a counterfactual explanation for the explained instance  $x^*$  based on the closest leaf  $l_{\min}$ , we compare their meta-feature representations  $b^* = t_b(t_l(x^*))$  and  $b^{\min} = t_b(l_{\min})$  respectively. By doing so we can identify feature value changes prescribed by the  $ij^{\text{th}}$  components  $b_{ij}$  of the meta-feature space  $\mathcal{B}$  and determined by the  $\tilde{\mathbb{I}}(b_{ij}^*, b_{ij}^{\min}) = 1$  factors of the distance metric.

The outcome of this procedure is a class-contrastive explanation of  $x^*$  that outlines a minimal change to its attribute values resulting in a different prediction. These counterfactuals are not necessarily specific data points, but rather feature tweaks expressed with valid ranges for numerical attributes and allowed values for categorical attributes. Such explanations are an improvement over vanilla counterfactuals since they additionally provide a context that implicitly generalises them – see Section 4.4 for more details. Moreover, our algorithm is guaranteed to return at least one explanation for each leaf as there always exists an explanation within distance 1, which is determined by the neighbouring leaf of  $t_l(x^*)$ . Our approach will also retrieve all other explanations within distance 1, which may be found for two mostly disjoint root-to-leaf paths leading to non-neighbouring leaves and agreeing on all features except the one at the root of the underlying tree. If several explanations within the same distance exist – see Table 4.1 for an example – we can choose one at random; alternatively, a heuristic based on feature importance or user preferences can be employed to break ties and address *explanation multiplicity* as discussed in Section 4.4.

**Supportive Explanations** The *minimal* and *most general* set of features for which specific ranges of numerical attributes and sets of values for categorical attributes guarantee a particular prediction can be computed for an arbitrary tree leaf and serve as a *supportive explanation*

(akin to the post-hoc anchors explainer [130]). In particular, such statements can be composed with *generalisation algorithms* [110], which are widely used in logic programming [42, 109]. To this end, we modify an appropriate off-the-shelf tool to support our meta-feature space, where the additional 0 value is introduced to indicate logical conditions that are undecided for a particular root-to-leaf path. Generating these explanations neither requires a similarity metric nor grounding the explained leaf with a fixed data point, making them unique to the structure of the explained tree  $t$ , for which they can be pre-computed and stored to facilitate a constant-time retrieval.

## 4.4 Making Trees Explainable

Our explanations exhibit a number of desired properties beyond their social [106] and legal [173] appeal reported by others. The explanatory power of contrastive statements (counterfactuals in particular) is widely documented in the XAI literature and verified by numerous studies in a variety of contexts [83, 107, 123, 155–157, 169, 173]. Therefore, in lieu of user studies we discuss explainability desiderata of CtreeX based on our taxonomy introduced in Chapter 2. This allows us to focus on inherent characteristics of class-contrastive and supportive explanations that are valued in the literature but may be lost unintentionally and unconsciously with their implementation, thus negatively affecting their effectiveness. For example, counterfactuals are deemed a natural explanatory mechanism because they are interactive and dialogue-like [106], however only a few explainers take advantage of this observation [151, 156]. Despite the mostly positive sentiment exhibited by the XAI community towards class-contrastive explanations, some of their (algorithmic) operationalisations simply lack what made them attractive in the first place. While not all of such desiderata are properties of the explainer per se, but rather the overall explanatory system within which it is deployed, our method seeks to enable and support their broadest possible range, which is partially made possible by its ante-hoc design (guarantee of retrieving a complete and faithful set of explanations) and duality of its output (counterfactual and supportive statements).

Counterfactual explanations appeal to humans because of their simplicity and versatility, whereas supportive statements provide a comprehensive overview of a black-box prediction, thus grounding it in a context. CtreeX uses both of them as their distinct scopes make them complementary: the former identifies *specific* and the latter *general* conditions leading to a particular prediction, helping to balance explanation fidelity. Both of our explanations are sound (**U1**, see Table 4.2 for reference) since they are derived with a tree-specific (**F6**), ante-hoc (**F7**) algorithm, thus guaranteeing full *faithfulness*. While it is common for class-contrastive explanations to lack *completeness* (**U2**) as their foil applies only to the explained instance and does not generalise to similar data points, CtreeX manages to partially overcome this limitation by *contextualising* them (**U3**) in three distinct ways.

Functional		Operational		Usability		Safety	
<b>F6</b>	Applicable Model Class	<b>O7</b>	Function of Explanation	<b>U1</b>	Soundness	<b>S1</b>	Information Leakage
<b>F7</b>	Relation to Predictive System			<b>U2</b>	Completeness	<b>S2</b>	Explanation Misuse
				<b>U3</b>	Contextfullness	<b>S3</b>	Explanation Invariance
				<b>U4</b>	Interactiveness	<b>S4</b>	Explanation Quality
				<b>U5</b>	Actionability		
				<b>U6</b>	Chronology		
				<b>U8</b>	Novelty		
				<b>U9</b>	Complexity		
				<b>U10</b>	Personalisation		
				<b>U11</b>	Parsimony		

**Table 4.2:** Summary reference of a subset of the XAI taxonomy (cf. Chapter 2) applicable to CtreetX.

1. Our foils are not specific data points (exact feature value tweaks) but instead ranges for numerical and sets for categorical attributes that lead to a different prediction, thereby providing a wider perspective that is valid for a whole sub-population instead of an individual prediction.
2. CtreetX can ground such explanations in a broader context with exemplars extracted from relevant fact and foil tree leaves (based on the underlying training data) or generate equivalent synthetic instances.
3. Class-contrastive statements can be accompanied by a supportive explanation that lays out a comprehensive view of the limitations pertaining to the counterfactual foil.

Supportive explanations, on the other hand, are not affected by *completeness* (**U2**) shortcomings since they explicitly state their generalisation limits – a region of the feature subspace for which a prediction is guaranteed. Unless the tree is overfitted, both of CtreetX explanation types are *stable* (**S3**) since they come from high-density (training) data regions [123] encapsulated by tree leaves – explanation quality can be measured by coverage and impurity of relevant leaves (**S4**). Furthermore, the explanations are always *short* and *sparse* since our approach optimises for brevity (**U9** & **U11**), which is improved even further by merging multiple logical conditions imposed on the same feature. In addition to enhanced interpretability, class-contrastive and supportive statements can also help to assess the model’s fairness [147], identify its biases [173] and uncover modelling bugs [166] (**O7**).

In Section 4.3.2, we proposed a basic distance metric that treats all of the attributes equally and does not quantify the size of change required in each dimension of the original data feature space  $\mathcal{X}$ . Furthermore, our metric selects an explanation at random when multiple other exist within the same distance. Both of these assumptions can be easily overridden by weighting

components of the meta-feature representation  $\mathcal{B}$  (e.g., according to the Euclidean distance between fact and foil) and applying a suitable counterfactual retrieval heuristic. Nonetheless, custom metrics may be specific to individual use cases; for example, *chronology* of events (**U6**), *novelty* of data points (**U8**, e.g., instances with rare or unexpected attribute values) and feature *importance* or *actionability* (**U5**, e.g., preferring foils based on income rather than ones conditioned on age) often need to be annotated separately for each data set.

Some of these objectives can be achieved algorithmically by tuning the distance function to explicitly *include* or *exclude* from the foil conditions imposed on certain attributes (**U10**). Alternatively, if deployed in a dynamic user interface, Ctrees can support explaineer-driven iterative personalisation of the explanations, which is particularly useful when adopting such an algorithm in an interactive setting (**U4**), e.g., a conversational agent [151, 156], thus making it more appealing to a lay audience who can directly influence the explanatory process – see Chapter 6 for more details. However, such an operationalisation raises *security* concerns if the model is proprietary [157] since multiple customised explanations can reveal (**S1**) exact splitting thresholds of the underlying decision tree, thus allowing an adversary to reverse-engineer and steal the predictive model (**S2**). A more comprehensive evaluation of Ctrees based on the XAI taxonomy outlined in Chapter 2 is presented in Appendix A.2 in form of an explainability Fact Sheet [149].

## 4.5 Tree Explainability in the Literature

Explainable AI and ML research is largely concerned with inherently opaque predictive models [71, 100, 129, 130] such as deep neural networks [91], which grew in popularity due to their impressive performance for a wide array of domains and applications. For example, Ribeiro et al. [128] introduced a model-agnostic explainer called LIME, which explains black-box predictions with influence of relevant interpretable concepts by locally approximating the behaviour of the black box with a simple and transparent surrogate model (see Chapter 3 for more details). Later, Ribeiro et al. [130] proposed anchors, which is a model-agnostic explainer that prescribes conditions guaranteeing a particular black-box prediction; the resulting explanations are similar to supportive statements generated by Ctrees since both of them determine a data subspace where a model prediction does not change. Similarly, Kim et al. [71] created TCAV, which identifies human-understandable concepts important for classification of images; and Lundberg and Lee [100] introduced SHAP, which computes contribution of individual features to black-box predictions.

Out of these four methods LIME, anchors and SHAP are model-agnostic and post-hoc, i.e., they are independent from the underlying black box and can be retrofitted to a preëxisting predictive model. However, this flexibility comes at a cost: post-hoc explainers create an additional layer of complexity that can be detrimental to the quality and faithfulness of their explanations [133]

– see Section 1.1 for an in-depth discussion of this topic. Therefore, in certain cases inherently transparent models such as rule lists and sets, linear models, naïve Bayes classifiers and decision trees may be preferred. While their operation can be traced and their parameters are meaningful to humans, explainability and interpretability of such models is questionable as we argued in Section 4.1. This presumption of intrinsic intelligibility makes such models easy to overlook for the explainable AI community, yet their transparency provides a solid foundation for creating faithful and robust ante-hoc *explainers*.

Explainability comes in many different shapes and forms, but contrastive statements – in particular class-contrastive, counterfactual, explanations – are proven to be the most natural and appealing for a lay audience and domain experts alike [106]. They can be used to audit fairness of ML models [83, 147] and they comply with various legal frameworks [173]. Notably, we should distinguish between two different types of counterfactuals. The notion proposed by Lewis [93] is based on an abstract idea of *similarity among hypothetical worlds*; these can be (synthetic) instances that produce different predictions for a *given* data-driven model – a paradigm dominating the XAI and IML landscape. Counterfactuals introduced by Pearl [117], on the other hand, are grounded in causality research and built directly upon the mechanism governing the underlying natural process or phenomenon, thus independent of any particular predictive algorithm used to model it.

Algorithmic approaches to counterfactual (class-contrastive) explainability include, among others, an optimisation technique compatible with differentiable predictive models such as neural networks and support vector machines [173], and a graph-based method called FACE (Feasible and Actionable Counterfactual Explanations [123]) for retrieving foils that are realistic with respect to the distribution of the underlying training data. Additionally, Tolomei et al. [166] offered a particular type of counterfactual statements – identifying features that can be tweaked to transform a true negative instance into one predicted as positive – designed for tree ensembles and used to improve on-line advertisements. However, their algorithm is limited to adjusting feature values in fixed intervals and perturbing attributes combinatorially, which makes it inefficient and prevents it from guaranteeing reproducibility with respect to the output explanations. Finally, van der Waa et al. [169] showed how to train surrogate binary foil trees, which locally explain a selected class (one-vs-rest) with counterfactuals, however the resulting explanations may not be symmetrical and the proposed method is focused more on the surrogate aspect of the technique than developing a principled tree explainer. Notably, all of these approaches, except FACE, boast benefits of contrastive explanations, but none of them considers their human aspects such as interactiveness, personalisation and actionability.

## 4.6 Explainable Tree-based Surrogates

This chapter introduced a novel approach to explaining decision tree predictions with sparse *class-contrastive* (counterfactual) and *supportive* statements composed by leveraging access to the internal structure of classification and regression trees. To this end, we proposed a meta-feature representation of the tree structure, allowing us to efficiently compute similarity between leaves of the tree using a custom distance function. Our approach is an improvement over the inherent tree transparency, which we argued is insufficient for true explainability since it lacks a reasoning mechanism that leads to understanding. We opted for counterfactual explanations given their strong theoretical foundations and practical capabilities grounded in the social science research on human explainability and its recent connection to artificial intelligence. Beyond improved interpretability of predictions, such explanations can be used to debug decision trees, e.g., identify a lack of predictive monotonicity for certain features, and expose their biases and unfair behaviour, e.g., a prediction change conditioned on gender.

Since CtreeX is ante-hoc, it exhibits a range of advantages and guarantees including full fidelity, explanation completeness, interactivity and customisability. These benefits, however, come at the expense of our method being tree-specific. In the next chapter we explore how to lift this limitation while preserving all of the desired properties. To this end, we revisit bLIMEy – our surrogate meta-algorithm introduced in Chapter 3 – which we employ to construct a faithful, model-agnostic and post-hoc surrogate explainer based on classification and regression trees. By using a decision tree-based surrogate, we can explain black-box predictions of image, text and tabular data with high-fidelity class-contrastive and supportive statements generated by CtreeX. Additionally, we show how such an explainer can address many issues exhibited by its more popular alternative that is built upon a linear surrogate, a prominent example of which is LIME. In particular, we focus on black-box image classification given the intuitiveness of such predictions and their (counterfactual and supportive) explanations.



## LIMETREE: TREE-BASED SURROGATES

Systems based on artificial intelligence and machine learning models should be transparent, in the sense of being capable of explaining their decisions to gain humans’ approval and trust. While there are a number of explainability techniques that can be used to this end, many of them are only able to output a single one-size-fits-all explanation that simply cannot address all of the explainees’ diverse needs. Having seen the power, flexibility and shortcomings of surrogate explainers as well as measures to mitigate them with the use of local decision trees (Chapter 3), we introduce a model-agnostic and post-hoc explainability technique for black-box predictions that employs surrogate multi-output regression trees. We validate our algorithm on a deep neural network trained for object detection in images and compare it against Local Interpretable Model-agnostic Explanations. Our method, which we call LIMETree, comes with local fidelity guarantees and can produce a range of diverse explanation types, including contrastive (counterfactual) and supportive statements recommended in the literature – the result of combining bLIMEy with our findings from Chapter 4, namely CtreeX. Some of these explanations can be interactively personalised to create bespoke, meaningful and actionable insights into the model’s behaviour – a topic which we discuss further in Chapter 6. While other methods may give an illusion of customisability by wrapping, otherwise static, explanations in an interactive interface, our explanations are truly interactive, in the sense of allowing the user to “interrogate” a black-box model. LIMETree can therefore produce consistent explanations, providing a solid foundation for an interactive exploratory process. The properties of our approach are summarised in Appendix A.3 in the form of an explainability Fact Sheet, which is based on our XAI taxonomy introduced in Chapter 2.

## 5.1 Surrogates for Humans

Transparency of predictive systems based on machine learning and artificial intelligence algorithms is desired for a variety of reasons. It can help to debug black-box models, inspect their fairness, evaluate their accountability and explain their decisions to relevant stakeholders. With this wide range of applications and diverse audiences, output of a single transparency algorithm cannot be expected to satisfy everyone's needs and expectations. While this may possibly be addressed by a dedicated team of data scientists responding to explainability requests by tweaking and tuning their toolkit, such an approach is inefficient. A more streamlined solution is to build *interactive* interpretability tools, through which the users can “ask” directly for the desired insights. This type of exploratory interaction gives users the flexibility to request customised and personalised analysis of a black box, possibly alleviating a need for technical skills and knowledge.

Interactive explainability in AI and ML is a somewhat overloaded term; it encompasses both explainability methods presented within *interactive interfaces* and truly *interactive explanations*. While the first kind may be desirable and is prevalent in the Human–Computer Interaction realm [77], many members of the explainable artificial intelligence and interpretable machine learning communities opt for the second, which, they argue, is the cornerstone of natural and human-like explanations rooted deeply in social sciences [106]. The latter approach bears promise for black-box predictive systems, which, fitted with such techniques, could interactively explain their nuances and decisions in a process that is intuitive to humans, for example, a voice-enabled natural language conversation. However, the interactivity of these explanations should extend beyond their delivery mechanism and allow the explainee to customise and personalise them by interrogating the black box. This aggregated approach marks a departure from one-size-fits-all explanation practices, accounting for the diversity of explainees' skills and backgrounds.

Designing such systems comes with two challenges: modelling the user interaction (an HCI component) and creating an explainability technique that can output personalised explanations based on user-provided information (an XAI component). Ideally, the approach should be independent of the underlying predictive algorithm and versatile enough to provide multiple explanation types of varying complexity. The latter property ensures coherence of the explanatory process as including explanations generated with different methods may lead to inconsistencies that can hurt users' trust [178]. Providing explainees with an opportunity to personalise the explanations empowers them to investigate properties of black boxes that fall beyond their transparency and interpretability. Bespoke explanations can inspect individual fairness of a prediction [83], e.g., counterfactual cues indicating disparate treatment, or help to debug the underlying black box [78].

Research into AI and ML transparency has recently seen major progress with numerous post-hoc and model-agnostic tools being proposed [28, 129, 142, 158, 169]. Some of these methods can implicitly produce customised explanations, achieved by off-line, non-interactive parameterisation.

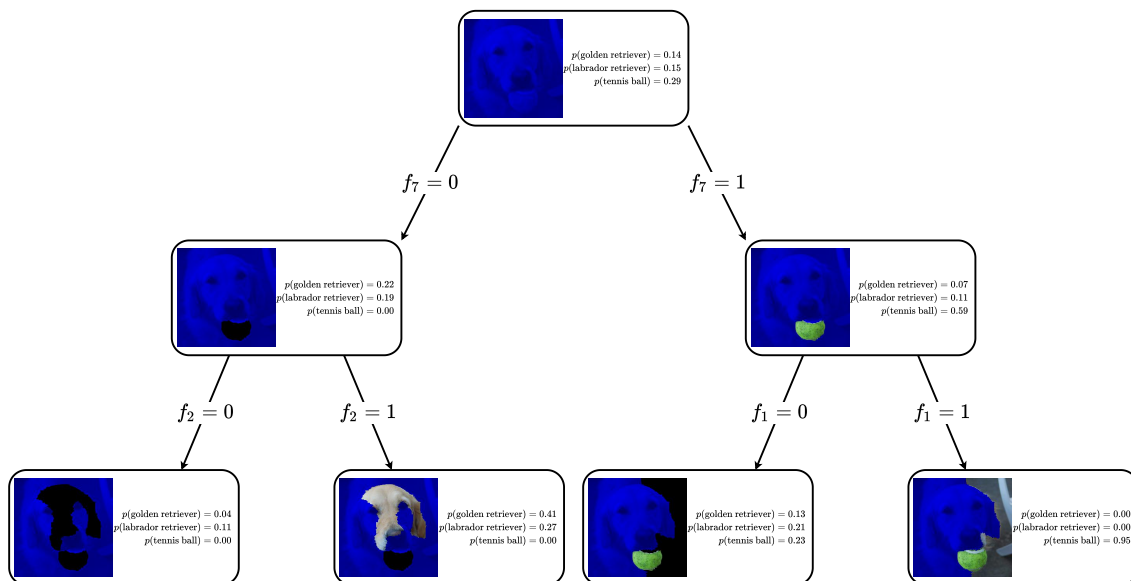
Work on counterfactual explanations [169, 173] is quite prominent as well since they are natural to humans [106] and compliant with various legal regulations [173]. In theory, they are also capable of interactive personalisation [151, 156], however this property has not been widely adopted.

Explainability methods that allow the end user, i.e., the explainee, to *customise* and *personalise* the explanation via an *interaction* are largely non-existent [138]. Some researchers [10, 101, 174] studied the formal communication and interaction protocols (e.g. in the form of a conversation) that in theory can facilitate an explanatory dialogue between two intelligent agents (humans, machines or one of each), but these concepts are yet to find applications in practical explainability tools. Non-personalised explanations and interactions with predictive systems have mainly come together to help the user debug [81] or customise and improve [69] the underlying ML algorithm. Interactive explainability systems allowing the user to request different types of static explanations have also been described [77, 178]. All of these techniques are discussed in more detail in Section 5.7.

In this chapter we draw inspiration from all these approaches and show how to achieve interactively customisable explanations of black-box predictions derived from surrogate multi-output regression trees (discussed in Section 5.3). Since surrogate explainers are post-hoc, model-agnostic and domain-independent (working with text, tabular and image data), our technique, which we call **LIMEtree**, can be retrofitted into any black-box predictive system. It enables explainees to interrogate an opaque ML model to understand and gain trust in its predictions, account for important decisive factors or prove fairness of its decisions. We chose trees as the surrogate model based on their ability to produce diverse and appealing explanations (cf. Chapter 4) such as:

- visualisation of the tree structure;
- tree-based feature importance;
- logical conditions extracted from a root-to-leaf path;
- exemplar explanations taken from training data falling into a single leaf;
- answers to what-if questions generated either based on the tree structure or by querying the model;
- **contrastive** explanations (including counterfactuals) retrieved by applying logical reasoning to compare different tree paths; and
- **supportive** explanations achieved by generalising logical conditions imposed on a tree path.

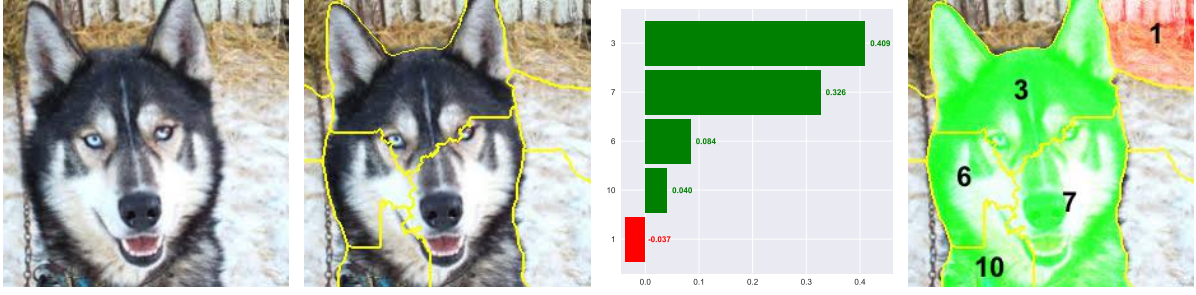
We already showed examples of these explanation types – accompanied by an extensive discussion – in Section 4.2 of the CtreeX chapter. The first two uncover the behaviour of a black box in a given



**Figure 5.1:** An example of a multi-output regression tree used to explain an image (taken from Figure 5.10) labelled as *tennis ball* by a black-box deep neural network image classifier. The super-pixels, i.e., segments, shaded in *blue* are not important to the explanation at any given tree node. A super-pixel which value is 0 in the interpretable representation is “removed” by occluding it with a solid black colour. A super-pixel assigned 1 in the interpretable representation is preserved. The probabilities estimated by the surrogate tree usually do not sum up to 1 in each tree node as these values may only represent a subset of modelled classes and are a result of a regression, thereby should not be treated as probabilities.

predictive subspace, whereas the remaining five target a specific prediction. While some of these explanations are inherently static, others can be embedded in an **interactive explanatory dialogue**, enabling the explaine to customise and personalise them in a natural way (more details in Section 5.5.1 and Chapter 6). We opted for **multi-output** regression trees – depicted in Figure 5.1 – to avoid common pitfalls associated with surrogate explainers and allow for modelling of multiple classes within the same surrogate model, thus creating a common source of explanations (see Section 5.3).

Our method builds upon *Local Interpretable Model-agnostic Explanations* [129] and *build LIME yourself* [158], described earlier in Chapter 3 and revisited in Section 5.2.1. LIMETree addresses many of LIME’s shortcomings and limitations (Section 5.2.2), and facilitates meaningful interaction with explanations to satisfy users’ expectations. By using a (shallow) regression tree as the surrogate model, we can guarantee its *full fidelity* with respect to the underlying black-box model under certain conditions. We demonstrate the explanatory power of our method with qualitative experiments and quantitative comparison on image classification tasks using a black-box deep neural network (Section 5.6). In Appendix A.3, we summarise properties of LIMETree within an explainability Fact Sheet, which is based on our XAI taxonomy introduced in Chapter 2.



(a) Image to be explained with LIME, predicted as *Eskimo dog* with 83% probability by a black-box model. (b) Interpretable representation of the image based on super-pixel segmentation. (c) *Eskimo dog* explanation presented as influence of the top five segments (regression coefficients). (d) *Eskimo dog* explanation shown in Panel (c) overlaid on top of the explained image from Panel (a).

**Figure 5.2:** Visual decomposition of a surrogate explanatory process for image data based on the LIME algorithm. The steps include generating an interpretable representation (b) and presenting an explanation in two different formats: a bar plot (c) and an image mask (d).

## 5.2 Surrogate Explainers, Revisited

The momentum behind surrogate methods can be attributed to several appealing properties that make them a universal explainability framework. They mimic behaviour of a complex black-box predictor either locally [129] or globally [28] with a simpler, inherently transparent model, thereby providing human-comprehensible insights into its operations. Surrogates are:

**model-agnostic** – can be used with any predictive system;

**post-hoc** – can be retrofitted into preexisting predictors; and

**data-universal** – are compatible with tabular, text and image data, which is enabled by *interpretable representations*.

LIME [129] is the most popular surrogate technique geared towards explaining predictions of black-box models – Figure 5.2 outlines its explanatory process for image data. Chapter 3 provides a more in-depth analysis of surrogate explainers in artificial intelligence and machine learning.

### 5.2.1 Local Surrogates of Images

LIME improves upon vanilla surrogate explainers by introducing an *interpretable data representation* (cf. Section 3.2.1). This concept extends their applicability beyond the inherently interpretable raw features such as *height* or *weight* for tabular data, allowing them to be used with sensory data such as images and structured data such as text. In this chapter we focus on applying surrogate explainers to *image recognition* tasks, which facilitates straightforward qualitative and quantitative evaluation of explanations by means of visual inspection, alleviating the need for technical background knowledge during user studies. Furthermore, a representation based on super-pixels, which is popular for images, exhibits properties that are necessary for

LIMEtree to achieve full fidelity. Nonetheless, all of our technical contributions can be applied to other data domains for which the interpretable representation satisfies the requirements outlined in Section 5.3.

The LIME algorithm trains a local surrogate used to explain an image  $\hat{x}$  for a black-box *probabilistic* model  $f$  by taking the following steps:

1. Find the human-interpretable representation  $\hat{x}' \in \mathcal{X}'$  of the data point  $\hat{x}$  by defining a mapping  $IR : \mathcal{X} \rightarrow \mathcal{X}'$  that transforms a data point from its original domain  $\mathcal{X}$  into the interpretable representation  $\mathcal{X}'$ . This mapping is usually provided by the user, although in certain cases it can be learnt, for example when the data is tabular and the surrogate model is a decision tree [158] – see Section 3.3.4 for more details. In the case of image data, the interpretable domain  $\mathcal{X}'$  is a (super-pixel) segmentation of the image  $\hat{x}$  represented as a binary vector  $\hat{x}' \in \mathcal{X}' = \{0, 1\}^d$ , where  $d$  is the number of segments. Such binary vectors  $x' \in \mathcal{X}'$  indicate whether a given segment should be preserved (1) or occluded (0), therefore the original image  $\hat{x}$  expressed in the interpretable representation is an all-1 vector  $\hat{x}' = [1, \dots, 1]$ . In practice, this is achieved with an image segmentation technique such as *quick shift* [171] implemented as part of the *scikit-image* Python package<sup>1</sup> [170].
2. Sample  $n$  data points uniformly at random from the interpretable representation  $\mathcal{X}'$  to get an  $n \times d$  binary matrix  $X' \subseteq \mathcal{X}'$  describing the neighbourhood of the explained image  $\hat{x}$ . Transform each data point (row) in this matrix back into the original representation  $\mathcal{X}$  using the inverse of the  $IR$  function –  $IR^{-1} : \mathcal{X}' \rightarrow \mathcal{X}$ . In practice, this is achieved by generating images that preserve the pixel values from the original image in the  $i^{\text{th}}$  segment if the  $i^{\text{th}}$  component of a binary vector  $x' \in X'$  is 1, i.e.,  $x'_i = 1$ , and replacing all of the pixels in this segment with the mean RGB colour of this segment if  $x'_i = 0$ . Next, the images recovered from the sampled data are classified with the black-box probabilistic model  $f$  to get an  $n \times c$  matrix holding probabilities for every class modelled by  $f$ , where  $c$  is the number of modelled classes.
3. Calculate a distance<sup>2</sup>  $L : \mathcal{X}' \times \mathcal{X}' \rightarrow \mathbb{R}$  between the explained data point and the sampled data in the interpretable representation  $\mathcal{X}'$ . Next, compute proximity/similarity scores by kernelising these distances using the exponential kernel  $k : \mathbb{R} \rightarrow \mathbb{R}$  defined as  $k(s; w) = \sqrt{\exp\left(-\left(\frac{s}{w}\right)^2\right)}$ , where  $w$  is the kernel width set to 0.25 by default.
4. Train a linear *regression*  $g : \mathcal{X}' \rightarrow \mathbb{R}$  as the surrogate model. A *sparse* regression is favoured to identify as few as possible important factors contributing to the explanation, thereby making it more comprehensible. The model is fitted to the data sampled in the binary interpretable representation  $\mathcal{X}'$  weighted by the kernelised distances (similarity scores).

---

<sup>1</sup>The `skimage.segmentation.quickshift` function.

<sup>2</sup>LIME suggests using either the Euclidean ( $L_2$ -norm) or cosine ( $L_{\cos}$ ) distance. We will use the cosine distance since our experiments suggested that it yields more intelligible explanations for images.

The target of the regression are probabilities – computed with the black-box model for the data  $X'$  sampled in step 2 – of a class  $\hat{c} \in \{1, \dots, c\}$  selected by the user to be explained, which tends to be the top prediction output by the black-box  $f$  for the explained image  $\hat{x}$ . The coefficients of this model are then used to quantify and interpret the positive or negative influence of each image segment on the black-box prediction of the explained instance. The feature weights of the surrogate model are directly comparable because all of them are within the same  $[0, 1]$  range, either 0 or 1 to be more precise. Usually, a separate linear regression is fitted for each of the top two or three classes predicted by the black-box model  $f$  for the original image  $\hat{x}$  as each surrogate can only explain a single class. In practice, this step is achieved with a *ridge* regression algorithm<sup>3</sup> implemented in the `scikit-learn` Python package [120].

A detailed description of LIME for other data domains can be found in Section 3.2.2 and the LIME paper [129].

This 4-step process optimises the *fidelity* of the surrogate model and the *complexity* of the resulting explanation. The first task translates into a small loss  $\mathcal{L}$  calculated between the output of the black-box model  $f$  and the surrogate model  $g$  – it measures how well the surrogate mimics the black box. Complexity  $\Omega$ , in the case of linear models, is computed as the number of non-zero (or significantly larger than zero) coefficients of the surrogate  $g$ . The mathematical formulation of this objective  $\mathcal{O}$  is given in Equation 5.1, where  $\mathcal{G}$  is the set of all the possible (sparse linear) surrogate models.

$$\mathcal{O}(\mathcal{G}; f, \hat{x}, X') = \argmin_{g \in \mathcal{G}} \underbrace{\Omega(g)}_{\text{complexity}} + \underbrace{\mathcal{L}(f, g; \hat{x}, X')}_{\text{fidelity}} \quad (5.1)$$

The *fidelity* of the surrogate model is measured empirically in the vicinity of the explained data point  $\hat{x}$  by evaluating the loss function  $\mathcal{L}$  given in Equation 5.2 for all the data points sampled from the interpretable representation  $X' \subseteq \mathcal{X}'$ . The locality of the metric is enforced by the sampling strategy in  $\mathcal{X}'$ , which only covers a small region around  $\hat{x}$  (namely, variations of the explained image), and weighting individual squared differences in predictions (probabilities of the explained class) by the similarity scores, i.e., kernelised distances. This particular loss function is inspired by the *Weighted Least Squares*, where the weights are distances  $L$  passed through the exponential kernel  $k$  and computed in the interpretable domain  $\mathcal{X}'$  between  $IR(\hat{x})$ , i.e., the explained data point transformed into the interpretable representation  $\mathcal{X}'$ , and the sampled data points  $x' \in X'$ . In Equation 5.2, the  $\hat{c}$  subscript in  $f_{\hat{c}}$  indicates the probability of the class  $\hat{c} \in \{1, \dots, c\}$  computed with the black-box model  $f$ .

$$\mathcal{L}(f, g; \hat{x}, X') = \sum_{x' \in X'} \underbrace{k(L(IR(\hat{x}), x'))}_{\text{weighting factor}} \times \underbrace{(f_{\hat{c}}(IR^{-1}(x')) - g(x'))^2}_{\text{individual loss}} \quad (5.2)$$

<sup>3</sup>The `sklearn.linear_model.Ridge` class.

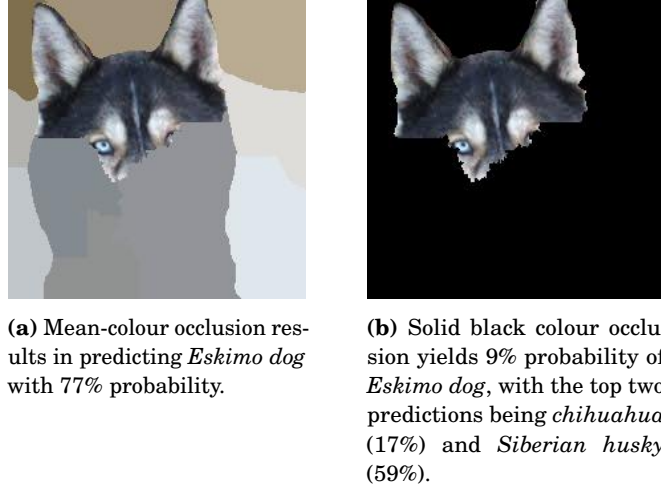
Figure 5.2 shows the various stages of the LIME image explainer. Panel 5.2a depicts the picture to be explained, which has been classified by the black-box *Inception v3* neural network [164] as *Eskimo dog*. Panel 5.2b shows the interpretable representation of this image – a (super-pixel) segmentation with  $d = 11$  interpretable features. The last two panels of Figure 5.2 present a LIME explanation of the *Eskimo dog* prediction in different formats: Panel 5.2c shows the influence of interpretable features (regression coefficients) as a bar plot and Panel 5.2d displays these segments superimposed on top of the original image.

### 5.2.2 LIME Trade-offs

Given the versatility and complexity of the LIME algorithm, it is subjected to various trade-offs. In particular, we look into the independence and linearity assumptions imposed on the interpretable features by the use of a linear model as the surrogate. We also examine the consequences of the explanations being limited to a single class. Next, we inspect various properties of the interpretable domain, i.e., image segmentation, and show how choices such as using the mean colour of a segment for its occlusion, granularity of the partition and object edges affect the explanations. Furthermore, we touch upon the impossibility of removing information from tabular and image data in the operationalisation of their respective interpretable representations, which is doable for text. We also analyse fidelity issues exhibited by surrogates, which can be attributed to inherent randomness and high level of parameterisation of the LIME algorithm. A more extensive discussion of these topics for all data domains – tabular, image and text – was given in Section 3.3 of the bLIMEy chapter.

**One Class Limitation** LIME explanations are confined to a single class, which makes the process of discovering the dependencies between different classes a challenge. For example, the same super-pixels may be important – in varying degrees – for two separate classes, potentially leading to confusion. Explaining multiple classes requires training a separate linear model for each one, therefore the explanations have to be interpreted independently, forcing the user to relate them and draw conclusions that may lack theoretical grounding and validation. Furthermore, when the underlying black-box model is not calibrated and the estimated class probabilities are pushed to the extrema (model over-confidence), the linear surrogate trained for any other but the top class may be very sensitive to variations in the sampled data.

**Linear Model Assumptions** Using the family of linear models as surrogates propagates their assumptions and restrictions to the resulting explanations. Linear classifiers are unable to model target variables that are *non-linear* with respect to the data features, which property does not necessarily hold for high-level meta-features such as image segments. Correlations and interactions among the data features may also have an adverse effect on the quality of such explanations. The latter observation is particularly important for interpretable domains



**Figure 5.3:** Black-box predictions for a single segment (#3) using different occlusion techniques. The chosen super-pixel is the most important part of the image according to its LIME explanation shown in Panel 5.2c. (This figure is an altered reproduction of Figure 3.3.)

with features that are highly structured or inter-dependent, e.g., adjacent image segments. This phenomenon can be observed by occluding all of the segments but #3 – visualised in Panel 5.3a – which is the most important meta-feature according to LIME. In this case, the probability assigned to *Eskimo dog* is 77% according to the black box, compared to 83% without **any** occlusions (Panel 5.2a), i.e., one segment “carries most of the probability mass”. However, with both segments #3 and #7 preserved – the two most important, and adjacent, segments with respective LIME scores of 0.4 and 0.3 – the probability of the same class increases by just 4 percentage points to 81%, i.e., due to their high correlation the surrogate model overestimates their individual importance. The observed behaviour is not uncommon given the nature of the interpretable representation and the intrinsic characteristics of *linear* models. Without replacing either of these two components, fixing this issue is simply impractical.

**Mean-colour Occlusion** LIME uses the mean colour of a segment for its occlusion, see Panel 5.3a for an example. This approach may have undesired effects for certain segmentation patterns and colour distribution of an image, in some cases undermining the utility of the occlusion procedure altogether. We already touched upon these issues in Section 3.3.2, however we revisit them here for completeness.

**Colour Uniformity** Segments that have a relatively uniform colour gamut may, effectively, be impossible to occlude. This is especially common for segments that are in the background or out of focus, e.g., bokeh and depth-of-field effects.

**Segmentation Granularity** The smaller the segments become, the more likely it is that their colour composition is uniform given the “continuity” of images, i.e., high correlation

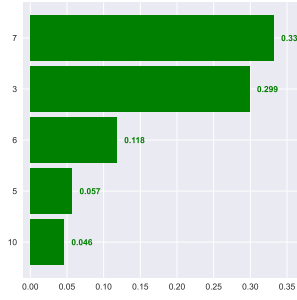
of adjacent pixels, resulting in a similar effect as above.

**Edges Preservation** Whenever the segmentation coincides with objects’ edges or regions of images where colour continuity is not preserved, which is common for edge-based segmenters, occluding super-pixels with their mean colour causes (slight) colour variations of adjacent segments, thus preserving the edges in the (partially) occluded images. Such patterns may convey enough information for the black-box model to recognise the image class correctly. This phenomenon can be observed in Panel 5.3a where despite occluding all of the segments but #3 with their mean colour, the black-box model still recognises the picture as *Eskimo dog* with a slight decrease in probability: 77% down from 83%.

Since these issues are artefacts of using the mean colour of each segment for its occlusion, it may seem that fixing a single occlusion colour for all of the super-pixels would eradicate some of these issues. However, from the discussion in Section 3.3.2, we already know that such an approach is not a silver bullet. It hides the edges between occluded segments and removes their content instead of just blurring the image, but the edges between occluded and preserved super-pixels are still present. Furthermore, the choice of the occlusion colour significantly impacts the explanations regardless of the colouring strategy. This type of interpretable representation implicitly assumes that the black-box model is indifferent to the occlusion colour, i.e., none of the modelled classes is biased towards it. Adjusting the granularity of the segmentation also plays an important role given high correlation of adjacent super-pixels.

To better understand the effect of a single colour occlusion on black-box image classification and LIME explanations we tweak the algorithm to occlude segments with a solid black colour, see Panel 5.3b for an example. In this case, when all of the segments but #3 are occluded, the top three classes predicted by the black-box model are *Siberian husky* with 59% probability, *chihuahua* with 17% and *Eskimo dog* with 9%. This is a drastic change from predicting 77% probability of *Eskimo dog* when using the mean-colour occlusion as illustrated in Figure 5.3. With a similar effect on other images partially occluded with a solid black colour instead, the corresponding LIME explanations are different despite using the same data sampled from the interpretable domain in each case – see Figure 5.4 for an example. Notably, the implicit assumptions of linear models transferred onto the surrogate explanations are also pertinent with this occlusion technique. The two most important segments are still #3 and #7, but in reversed order and with respective influence of 0.299 and 0.332 in contrast to 0.409 and 0.326 for the mean-colour occlusion.

Observing the influence of each algorithmic component on the variability of LIME explanations has prompted us to reexamine Ribeiro et al.’s conclusions drawn from experimental results presented in the LIME paper [129]. In particular, the unintended consequences of occlusion colour sensitivity cast doubt on the importance of snow in the background of the image shown in Panel 5.2a as suggested by Ribeiro et al. [129]. Replacing the segments of this picture showing snow with their respective mean-coloured patches produces off-white mosaic that still resembles snow, for example, compare the bottom-left and the bottom-right segments in Panels 5.3a and 5.2a.



**Figure 5.4:** LIME explanation for the Husky image (Panel 5.2a) when using black occlusions. It was generated based on the same interpretable representation and (binary) data sample as the explanation for the mean-colour occlusion presented in Panel 5.2c, making them directly comparable. (Segment #5 is the one below #1.)

These almost visually indistinguishable alterations may therefore have insignificant influence on the probabilities output by a black-box model, as shown in Figure 5.3, thereby decreasing the soundness of LIME explanations.

**Impossibility of Information Removal** The influence of the occlusion colour stems from the impossibility of truly removing a super-pixel as many image classifiers cannot handle “missing” data. Occlusion is thus a proxy for hiding information from a black-box model, which is a means for testing its sensitivity to the information contained therein – step 2 of the LIME algorithm outlined in Section 5.2.1. A similar phenomenon can be observed for explanations of tabular data when using an interpretable representation based on discretisation and binarisation of continuous features [129, 158] (cf. Section 3.2.1). This type of an interpretable representation combined with a linear surrogate model yields an explanation that indicates the influence of a particular feature value being within or outside of a given numerical range on the black-box prediction of the explained instance. Selecting these bin boundaries is non-trivial and biases the explanation in a similar way to the effect of the occlusion colour choice when explaining images. However, the third data domain – text – is less prone to such issues as many black-box text classifiers do not impose length or content restrictions on their input. This means that words or tokens can be *explicitly* removed from the explained text excerpt, thereby not biasing the explanation in any way.

**Fidelity Issues** Finally, the flexibility and generality of LIME – it is post-hoc and model-agnostic – also contribute to the instability of its explanations [89, 158, 183]. Since the training data for a local surrogate is sampled randomly, there are no guarantees with respect to reproducibility and stability of the explanations unless the random seed is fixed, which only provides an illusion of stability. These problems with *local fidelity* of surrogates, i.e., their predictive coherence with respect to the underlying black-box model, are not limited to LIME and are the major factor inhibiting their uptake as reported by Rudin [133]. The multitude of parameters

and possible component choices when building surrogates further contribute to this phenomenon: number of samples, distance metric, kernel (width), interpretable representation (segmentation) and occlusion colour, to name a few [46]. All in all, surrogates only (locally) *approximate* complex behaviour of a black-box model and if their fidelity is miscommunicated to the explaine, such explanations may be misleading.

## 5.3 Surrogate Multi-output Regression Trees

In order to alleviate LIME’s implicit assumption of a linear relation between the interpretable features and the target variable as well as independence of the interpretable features, we propose a surrogate explainer based on *regression trees*. Given the rich family of decision trees and their diverse capabilities – regression and binary or multi-class classification trees, which are often referred to as CART (Classification And Regression Trees) [23] – choosing the appropriate tree type is crucial.

### 5.3.1 Advantages of Multi-output Regression Surrogates

**Regression and Non-probabilistic Classification** When the explained black box is a *regressor*, the surrogate model also has to be a regressor unless we are willing to discretise the output of the black box. Similar reasoning applies to *non-probabilistic* black-box classifiers: the surrogate must be a classifier unless we encode the class predictions as probability vectors. Furthermore, if the black-box classifier is multi-class, the surrogate can either be fitted to predict (and explain) one of the classes, i.e., binary one-vs-rest, or to model a selected subset of classes, i.e., multi-class. Naturally, these two cases are indistinguishable for *binary* black-box models. Each decision, including the choice of a model family, entails different assumptions and explanatory power of the resulting surrogate.

When the black box is a regressor and the surrogate is a regression tree, the optimisation objective  $\mathcal{O}$  as defined in Equation 5.1 and the loss function  $\mathcal{L}$  given in Equation 5.2 remain unchanged. The model complexity function  $\Omega$ , however, is adapted to trees, thereby measuring either the *depth* of the surrogate or its *width* (number of leaves) as shown later by Equation 5.5 in Section 5.3.2. The choice between the two mostly depends on the type of explanation that we want to extract from the surrogate tree, for example, depth may be preferred when visualising the tree structure or extracting rules. Nevertheless, in certain cases, e.g., unbalanced trees with the extreme case being one-sided trees, optimising for width or a combination of the two can be more helpful.

When the black-box model is a non-probabilistic classifier and the surrogate is a classification tree, the optimisation objective  $\mathcal{O}$  remains as defined in Equation 5.1, but the loss function  $\mathcal{L}$  given in Equation 5.2 is adapted from regression to classification. To this end, the squared error component of the loss function  $\mathcal{L}$  is replaced with an indicator function, resulting in a weighted

accuracy. This loss function for classification is shown in Equation 5.3, with the underlined term indicating the altered part. Any other classification evaluation metric can be used within  $\mathcal{L}$  by modifying it in this manner. Similarly to surrogate regression, the model complexity function  $\Omega$  is adapted to trees using Equation 5.5 shown later in Section 5.3.2.

$$\mathcal{L}(f, g; \hat{x}, X') = \sum_{x' \in X'} k(L(IR(\hat{x}), x')) \times \underline{\mathbb{1}(f_{\hat{c}}(IR^{-1}(x')), g(x'))} \quad (5.3)$$

**Probabilistic Classification** This chapter focuses on a more common scenario, especially for object recognition tasks built on top of neural networks, where the black box is a *probabilistic* classifier. One approach is to transform such models into non-probabilistic classifiers by applying  $\arg\max$  to the vector of predicted probabilities and proceeding as described above. Doing so, however, is sub-optimal as it leads to losing vital information about the confidence of the model’s prediction. For example, the top two classes may be almost equally likely – 49% *Labrador retriever* and 48% *golden retriever* – or one of them may be dominant – 98% *Siberian husky*. The latter disproportion is often visible when the number of modelled classes is relatively large, e.g., the popular ImageNet data set [33] has 200 classes, many of which are highly correlated, e.g., malamute, Eskimo dog, (Siberian) husky and (grey/Arctic) wolf. Such adverse behaviour is not uncommon and can be partially attributed to a model’s overconfidence and poor calibration [82], which get magnified when treating a probabilistic predictor as an  $\arg\max$  classifier.

A more natural approach in this case is fitting a surrogate regressor to the probabilities predicted by the black box. In this setting, one surrogate model is required for every explained class where it implicitly acts as a *one-vs-rest* explainer with respect to the classes predicted by the black box. Intuitively, a surrogate regression tree for a class  $A$  can only answer questions about the probability of this single class, with the complementary probability  $p(\neg A) = 1 - p(A)$  modelling the union of all the other possible classes  $\neg A = B \cup C \cup \dots \cup Z$ . The explanations, e.g., counterfactuals, extracted from surrogate regressors are thus limited to answering “Why  $A$  rather than  $\neg A$ ?” questions, which may have insufficient explanatory power for non-binary tasks. Other viable explanation types follow a similar pattern: “How important are selected features for class  $A$ ?”, “How does the tree structure tell apart class  $A$  from all the other classes?”, and “What are the logical rules used to identify class  $A$ ?”

The magnitude of the probability  $p(A)$  predicted by the surrogate when explaining class  $A$  can also be problematic in certain cases and presents us with similar challenges to treating the black box as an  $\arg\max$  classifier. If  $p(A)$  is (much) greater than 0.5, class  $A$  is clearly dominant and often we do not need to worry about the other classes. However, if  $p(A) \leq 0.5$  we cannot be certain whether there is a single event  $B$  with  $p(B) \geq p(A)$ , or alternatively the combined probability of all the complementary events  $p(\neg A)$  is greater than or equal to  $p(A)$  with no single event dominating over  $A$ . To complicate matters even further, the numerical output of some surrogate regressors is unbounded, which will be confusing to an explainee expecting a proper

probability within the  $[0, 1]$  range. This last property affects linear models but not regression trees since the latter output the mean of the target value for training data points in each leaf, which lies between their minimum and maximum value, therefore is guaranteed to be within the  $[0, 1]$  range.

The training procedure of surrogate regression trees utilises the unchanged optimisation objective  $\mathcal{O}$  and loss function  $\mathcal{L}$  as defined in Equations 5.1 and 5.2 respectively. Since we are using regression trees, the model complexity function  $\Omega$  has to be adapted appropriately, as given by Equation 5.5 in Section 5.3.2.

**Trade-offs Between Regression and Classification Surrogates** There is a clear trade-off between regression and multi-class classification surrogate trees when dealing with *probabilistic* black boxes. While the mechanism of the former is appealing, it comes with sever restrictions and caveats impeding its widespread applicability. For example, fitting a separate surrogate for each explained class, which is required for surrogate regressors, can cause the resulting trees to be structurally inconsistent. This means that juxtaposing explanations for different classes may present competing or even contradictory evidence, which risks confusing the explainees and puts their trust at stake. Surrogate (multi-class) classifiers, on the other hand, overcome this challenge and explicitly allow to answer both “Why  $A$  rather than  $\neg A$ ?” and “Why  $A$  rather than  $B$ ?” questions, thereby uncovering relations between multiple classes. Such explanations are more powerful and more natural to the explainee but come at the cost of losing important information when applying  $\text{argmax}$  to the probabilistic output of a black-box classifier.

### 5.3.2 LIMETree

To address the issues discussed in the previous section, we propose to use a **multi-output regression tree** as the surrogate model, which in many ways provides the best of both worlds. It simulates *multi-class* modelling in a *regression* setting, allowing the surrogate to *capture interactions* between multiple classes, hence explain them coherently. This is a significant improvement over training a separate one-vs-rest regression surrogate for each explained class, which may produce diverse and competing explanations because these models do not necessarily share a common tree structure or may split on different feature subsets. Since class probabilities predicted by the black box and used as target variables for training the surrogates are highly correlated, independent one-vs-rest surrogates cannot replicate this behaviour. For example, an increase in the predicted probability of class  $A$  causes the probability of another event  $B$  to decrease, which plays an important role among the top classes output by the black box. Notably, since each leaf can model probabilities of multiple classes, their sum may be greater than 1 for any given leaf, which can be addressed by rescaling them to avoid confusing the explainee.

To ensure low complexity and high fidelity of our multi-output regression trees, we employ the same optimisation objective  $\mathcal{O}$  as given in Equation 5.1 and use either of the decision tree-

specific complexity functions  $\Omega$  given in Equation 5.5, where  $d$  is the dimensionality of the binary interpretable domain  $\mathcal{X}'$ . We also adapt the loss function  $\mathcal{L}$  to account for the surrogate tree  $g$  outputting multiple values in a single prediction as shown in Equation 5.4, where  $C \subseteq \{1, \dots, c\}$  is the subset of classes chosen to be explained by  $g$ , for which the  $\hat{c}$  subscript in  $g_{\hat{c}}(x')$  indicates a prediction of the class  $\hat{c} \in C$  for a data point  $x'$ . In practice, this is achieved by training a multi-output tree regressor<sup>4</sup> implemented by the `scikit-learn` Python package [120] and iteratively increasing the depth or width bound of the tree to optimise the objective function  $\mathcal{O}$ . The optimisation procedure terminates when the loss  $\mathcal{L}$  defined in Equation 5.4 reaches a certain, user-defined level  $\epsilon \in [0, 1]$ , which corresponds to the fidelity of the local surrogate, i.e.,  $\mathcal{L}(f, g; \hat{x}, X') \leq \epsilon$ . Increasing the complexity of the surrogate model  $\Omega(g)$  improves its predictive power, which allows to further minimise the loss  $\mathcal{L}$ .

$$\mathcal{L}(f, g; \hat{x}, X') = \frac{1}{\sum_{x' \in X'} \omega(x'; \hat{x})} \sum_{x' \in X'} \left( \omega(x'; \hat{x}) \frac{1}{2} \sum_{\hat{c} \in C} (f_{\hat{c}}(IR^{-1}(x')) - g_{\hat{c}}(x'))^2 \right) \quad (5.4)$$

$$\text{where } \omega(x'; \hat{x}) = k (L_{\cos}(IR(\hat{x}), x'))$$

$$\Omega(g; d) = \frac{\text{depth}(g)}{d} \quad \text{or} \quad \Omega(g; d) = \frac{\text{width}(g)}{2^d} \quad (5.5)$$

Note that the inner sum  $\sum_{\hat{c} \in C}$  over the explained classes is scaled by a factor  $\frac{1}{2}$  since the biggest squared difference can be 2. This happens when the predictions of  $f$  and  $g$  assign a probability of 1 to two different classes, e.g.,  $[1, 0, 0]$  and  $[0, 0, 1]$ . The underlying assumption is that the sum of values predicted by each leaf of the surrogate tree is smaller or equal to 1, which may require normalisation in some cases. The outer sum  $\sum_{x' \in X'}$  is normalised by the sum of weights  $\omega(x'; \hat{x})$  to ensure that the loss  $\mathcal{L}$  is bounded between 0 and 1, facilitating a direct comparison of different surrogates and allowing for a meaningful user-defined parameter  $\epsilon$ .

Putting everything together we arrive at Algorithm 5.1, which we call LIMEtree. While the algorithm itself is relatively lightweight, manipulating images and querying black-box models may become a bottleneck. The explainees has no control over the computational and memory complexity of querying the black-box model  $f$ , which is executed  $n$  times, where  $n$  is the number of data points sampled from the interpretable domain. Given the recent advances in hardware dedicated for machine learning applications, this step should not be a burden when utilising GPUs, and manageable with just CPUs. Moreover, transforming the interpretable representation (binary vectors) into the original domain (images) requires a considerable amount of RAM. The explained image has to be duplicated for every data point sampled from the interpretable domain, and its RGB pixel values need to be altered to reflect segment occlusions. The efficiency of these two steps can be improved significantly with batch processing and parallelisation, therefore reducing the use of operational memory and decreasing the overall processing time. Other parts of the algorithm, which are executed just once, are relatively efficient: sampling a binary matrix

<sup>4</sup>The `sklearn.tree.DecisionTreeRegressor` class.

---

**Algorithm 5.1:** LIMETree.
 

---

**Data:** • black-box model  $f$  • explained data point  $\hat{x}$  • interpretable representation transformation function  $IR$  and its inverse  $IR^{-1}$  • samples number  $n$  • set of classes to be explained  $C \subseteq \{1, \dots, c\}$  • distance function  $L$  • kernel  $k$  • tree depth bound  $d$  • expected fidelity of the local surrogate  $\epsilon$

**Result:** local surrogate multi-output regression tree

- 1  $X' \leftarrow$  sample  $n$  data points from the interpretable domain  $\mathcal{X}'$ ;
  - 2 Transform the sample into the original domain  $\mathcal{X}$  with  $X = IR^{-1}(X')$ ;
  - 3 Predict the probabilities of  $X$  with the black-box model  $f$ ;
  - 4 Compute the distances between  $IR(\hat{x})$  and the sample  $X'$  using  $L$ ;
  - 5 Compute the weights by kernelising the distances with  $k$ ;
  - 6 **for**  $i \in [1, \dots, d]$  **do**
  - 7     Fit a multi-output regression tree  $g$  with a depth bound  $i$  to the weighted data set  $X'$  using the specified subset  $C$  of class probabilities from Step 3 as the target;
  - 8     Break the loop if the surrogate reaches the user-defined fidelity  $\epsilon$ , i.e.,  $\mathcal{L}(f, g) \leq \epsilon$ ;
  - 9 **end**
  - 10 Return the optimal tree;
- 

from the interpretable domain, fitting a multi-output regression tree to binary data with feature thresholds fixed at 0.5 and segmenting the explained image.

### 5.3.3 Improved Surrogate Fidelity

Our multi-output regression trees can significantly improve the local fidelity of explanations, which, as already discussed, has been identified as a major drawback of surrogate explainers [133]. To this end, we use Definition 5.1 to retrieve the *minimal* interpretable representation  $X'_T$ , which is unique for each tree. Intuitively, this set is composed of binary vectors  $x'_t$  from the interpretable representation  $\mathcal{X}'$  – one for each leaf  $t \in T$  of the decision tree – that have the least possible number of 0 components while still being assigned to the leaf  $t$ . For images, this can be understood as looking for the minimal possible occlusion of an image for each leaf of the tree – a 0 component of a vector in the interpretable representation indicates an occluded segment.

**Definition 5.1.** Assume a binary decision tree  $g$  with a set of leaves  $T$  fitted to a binary  $d$ -dimensional data set  $X' \subseteq \mathcal{X}' = \{0, 1\}^d$ . This tree assigns a data point  $x' \in \mathcal{X}'$  to a leaf  $t \in T$  with the function  $g_{\text{id}}(x') = t$ . For a selected tree leaf  $t$ , its *unique minimal* data point  $x'_t$  is given by:

$$x'_t = \underset{x' \in \mathcal{X}'}{\operatorname{argmax}} \sum_{i=1}^d x'_i \quad \text{for } g_{\text{id}}(x') = t,$$

where  $x'_i$  is the  $i^{\text{th}}$  component of the binary vector  $x'$ . We can further define a **minimal** set of data points  $X'_T \subseteq \mathcal{X}'$  that uniquely represents the tree  $g$  and the set of its leaves  $T$ . It is composed of

all the *minimal* data points for this tree:

$$X'_T = \{x'_t : t \in T\}.$$

Next, we transform this minimal representation set  $X'_T$  from the interpretable into the original domain  $\mathcal{X}$ , i.e., images, using the inverse of the interpretable representation transformation function with a fixed occlusion colour, e.g., black, resulting in  $X_T = \{IR^{-1}(x'_t) : x'_t \in X'_T\}$ . We then predict class probabilities for each image in  $X_T$  with the black box  $f$  and **replace** the values estimated by the surrogate tree with these probabilities for each leaf  $t \in T$ , i.e., modify the surrogate tree by overriding its predictions. Doing so is only feasible for the tree leaves as the *minimal data points* for some of the splitting nodes are indistinguishable; for example, all of the nodes on the root-to-leaf path that decides every interpretable feature to be 1 are non-unique and would be represented by the original (non-occluded) image. This procedure ensures **full local fidelity** of the surrogate tree with respect to the *explanations derived from the tree structure* such as counterfactuals and root-to-leaf decision rules. However, for this property to hold, the function that transforms the data points from the original domain into the interpretable representation  $IR$  has to be *deterministic* as outlined by Lemma 5.2. This assertion follows from the discussion presented in Section 3.3.2, which is summarised in the next paragraph.

**Lemma 5.2.** *A decision tree surrogate can achieve **full fidelity** with respect to the explanations derived from the structure of this tree – model-driven explanations – if the function  $IR : \mathcal{X} \rightarrow \mathcal{X}'$  transforming data from their original domain  $\mathcal{X}$  into an interpretable representation  $\mathcal{X}'$  is **deterministic**. This means that the mapping from  $\mathcal{X}$  to  $\mathcal{X}'$  is a one-to-one correspondence, with the  $IR$  function having a corresponding and uniquely defined inverse function  $IR^{-1} : \mathcal{X}' \rightarrow \mathcal{X}$ . Therefore, a data point  $x \in \mathcal{X}$  can be translated into a unique data point  $IR(x) = x' \in \mathcal{X}'$  and vice versa  $IR^{-1}(x') = x$ .*

Intuitively, the determinism of the interpretable representation transformation function imposed in Lemma 5.2 implies that each leaf in the surrogate tree is associated with only a single *minimal* data point  $x_t$  in the original representation  $\mathcal{X}$ . This data point is derived from the *minimal* interpretable data point  $x'_t$  by applying the inverse of the interpretable representation transformation function  $IR^{-1}$ , i.e.,  $x_t = IR^{-1}(x'_t)$ . Therefore,  $x_t$  represents the original image with the smallest possible number of occluded segments with  $g_{id}(x'_t) = t$ . By assigning the probabilities predicted by the black box for each data point  $x_t$  to the corresponding leaf  $t$  of the surrogate, it achieves full fidelity for the minimal representation set, which is the backbone of *faithful* model-driven explanations. The interpretable representation of images introduced in this chapter, and used by LIME [129], is *deterministic* since a single image partition is created and the underlying occlusion strategy is fixed: an identical colour for all segments in our experiments (Panel 5.3b) and a segment-specific mean colour occlusion in LIME (Panel 5.3a).

While such a setting ensures full fidelity of model-driven explanations, the same is not guaranteed for data-driven explanations such as answers to what-if questions, e.g., “What if

segments #3, #5 and #9 were absent?” Root-to-leaf paths that do not condition on all of the binary interpretable features allow for more than one data point to be assigned to that leaf, e.g., for three binary features  $(x_1, x_2, x_3) \in \{0, 1\}^3$ , a root-to-leaf path with  $x_1 \leq 0.5 \wedge x_3 \leq 0.5$  conditions assigns  $(0, 0, 0)$  and  $(0, 1, 0)$  to the corresponding leaf. This observation prompted us to specify the minimal interpretable representation  $X'_T$  (Definition 5.1) that chooses a single data point to represent each leaf, thus enabling full fidelity of model-driven explanations without additional assumptions. However, to achieve *full fidelity* for *data-driven* explanations as well, the surrogate tree must faithfully model the *entire* interpretable feature space, i.e., have one leaf for every data point in this feature space, which can be thought of as *extreme overfitting*. Since the cardinality of a binary  $d$ -dimensional space  $\mathbb{B}^d = \{0, 1\}^d$  is equal to  $|\mathbb{B}^d| = 2^d$ , and a complete and balanced binary decision tree of  $2^d$  width (number of leaves) is  $d$ -deep, relaxing the tree complexity bound  $\Omega$  accordingly guarantees full fidelity of all the explanations. This finding is expressed by the following corollary, which stems from Lemma 5.2.

**Corollary 5.3.** *If the complexity bound  $\Omega$  (width) of a surrogate tree  $g$  is relaxed to be equal to the cardinality of a binary interpretable domain  $\mathcal{X}'$ , i.e.,  $\Omega(g) = |\mathcal{X}'|$ , the surrogate is guaranteed to achieve **full fidelity**. This property applies to explanations that are both:*

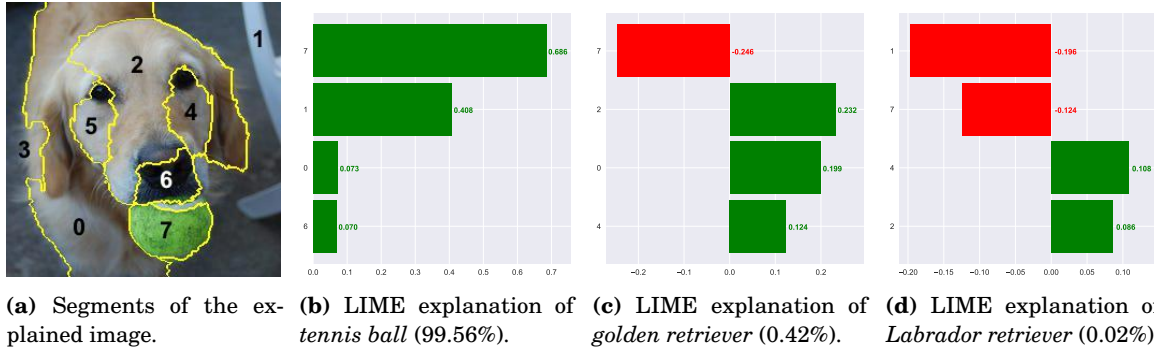
**data-driven** – derived from any data point in the interpretable representation, and

**model-driven** – derived from the structure of the surrogate tree.

Therefore, a surrogate tree that guarantees faithfulness of *model-driven* explanations (Lemma 5.2) can only deliver trustworthy counterfactuals and exemplar explanations sourced from the minimal representation set. For such surrogates, we can also generate what-if explanations with full fidelity by bypassing the surrogate tree and directly querying the black-box model. This may be an attractive alternative for more complex surrogate trees that additionally guarantee faithfulness of *data-driven* explanations (Corollary 5.3) whenever the black-box predictive function is accessible to the explaineer and querying it is not prohibitively expensive in time or compute. This latter surrogate type, which usually results in deeper trees, can deliver a broader spectrum of trustworthy explanations: tree structure-based explanations, feature importance, decision rules (root-to-leaf paths), answers to what-if questions and exemplar explanations based on *any* data point, in addition to counterfactuals.

## 5.4 Examples of LIMETree Explanations

To support the discussion and experimental results presented in the following sections, we first introduce examples of LIMETree explanations and compare them with the corresponding explanations produced by LIME [129]. After personalising the interpretable representation, as shown in Panel 5.5a, we explain the top three classes predicted by a black-box model: *tennis ball* (99.56%), *golden retriever* (0.42%) and *Labrador retriever* (0.02%). Their LIME explanations are

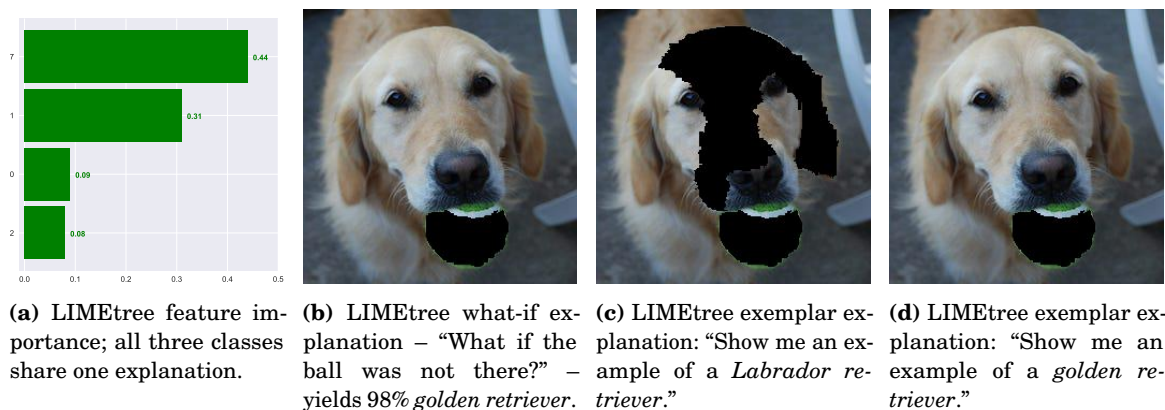


**Figure 5.5:** LIME explanations for the top three classes predicted by a black-box model for the image shown in Panel (a): *tennis ball* with 99.56% (b), *golden retriever* with 0.42% (c) and *Labrador retriever* with 0.02% (d).

given in Figure 5.5. As expected, segment #7, which depicts the ball, has an overwhelmingly positive influence on the *tennis ball* prediction – see Panel 5.5b. We can also see that this explanation is significantly affected by the correlation of the interpretable features since all of the important segments following #7 – i.e., #1, #0 and #6 – are adjacent and fully surround it. The second most important segment for this class is #1, with magnitude that is almost six times larger than the magnitude of the next two segments. Intuitively, the reason behind this configuration is the white stripe – a characteristic feature of tennis balls – appearing in this segment.

The other two LIME explanations shown in Panels 5.5c and 5.5d are for *golden retriever* and *Labrador retriever* respectively. For both predictions, segment #7 has a relatively large negative influence, which is expected, and segments #2 and #4, forming the dog’s face, have a positive effect. The difference between predicting these two dog breeds is determined by the positive effect of segment #0 on the *golden retriever* class (maybe because it reveals the long coat) and the negative influence of segment #1, which includes the white stripe of the tennis ball, strongly indicating the *Labrador retriever* class. Based on this evidence alone, it is difficult to determine the model’s heuristic for telling apart the two classes; in particular, the role played by segment #1.

Next, we explain these three classes with LIMETree, which can produce various types of explanations, thus helping us to analyse the behaviour of the underlying black box. We have already shown one type of explanation – the surrogate tree structure visualisation – in Figure 5.1. The depth of this tree was limited to two for the purpose of presentation, therefore it complies with Lemma 5.2 but not with Corollary 5.3, only achieving full fidelity with respect to model-driven explanations. Another explanation type, which closely resembles LIME explanations, gives the importance of interpretable features (calculated with the *Gini importance* [22] as described earlier in Section 4.2) and is shown in Panel 5.6a. Since LIMETree models all three classes simultaneously, the importance explanation captures the image segments that help to

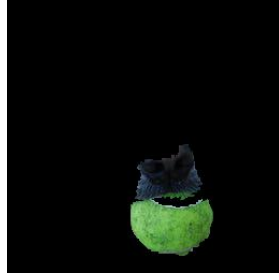
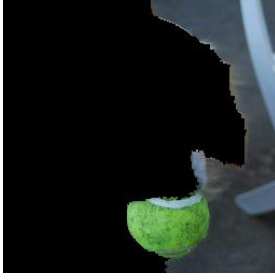


**Figure 5.6:** Three types of LIMETree explanations: (a) feature importance, (b) what-if explanation and (c–d) exemplar explanation.

differentiate between these classes. Comparing Panel 5.6a with analogous LIME explanations in Figure 5.5 shows a satisfying overlap, with each LIME explanation sharing three of its important segments with the LIMETree explanation. The tree-based feature importance clearly indicates that segments #7 and #1 (depicting the ball) are the most important, owing to the dominant prediction of *tennis ball* (99.56%), and are followed by segments #0 and #2, which together encompass most of the dog. While informative, these insights cannot be explicitly attributed to any single class and the feature importance values can only be positive adding to this issue.

Since all of the LIMETree explanations are coherent – they come from the same surrogate tree – with some help of another explanation type, e.g., the tree structure visualisation presented in Figure 5.1, we can discover the relation between each important feature and the three explained classes. Comparing the two leftmost with the two rightmost leaves – the result of the root split on segment #7 – tells us that this segment has positive influence on the *tennis ball* prediction. Additionally, when segment #1 is present, this prediction strengthens, however without it, while *tennis ball* is still the most likely prediction, *Labrador retriever* is almost equally likely and nearly twice as likely as *golden retriever*. On the other hand, when the ball is absent, i.e., segment #7 is occluded, both dog breeds are almost equally likely with the presence of segment #2 being the deciding factor: it is *Labrador retriever* if it is occluded and *golden retriever* if it is present.

Arriving at these conclusions required us to use feature importance and simultaneously inspect the tree structure, which cannot be expected of a lay explainee or when the surrogate tree is complex. In such cases, we can use other types of explanations, for example, interactive what-if questions. Since the tree presented in Figure 5.1 is not complete (see Corollary 5.3), we use the black-box model instead of the tree to evaluate hypothetical scenarios. Because segment #7, depicting the ball, is the most important factor, we are interested in *what if* this segment was not there; as expected, the new prediction is 98% *golden retriever* – see Panel 5.6b. We can also ask for exemplar explanations of the *Labrador retriever* and *golden retriever* classes, which are shown in Panels 5.6c and 5.6d respectively.



(a) When segments #7 and (b) Preserving segments #7  
#1 are preserved, the pre- and #6 yields 66% *tennis*  
diction is 90% *tennis ball. ball.*

**Figure 5.7:** The shortest LIMETree explanations of *tennis ball*.



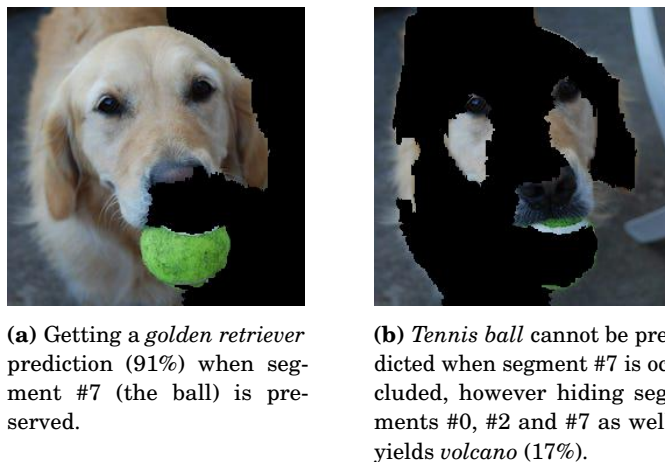
**Figure 5.8:** Visual representation of a LIMETree rule explanation that maximises the *Labrador retriever* prediction (99%).

In order to take full advantage of LIMETree explanations, we train a *complete* surrogate tree (see Corollary 5.3). We use it to ask for the *shortest* possible explanation, i.e., the highest number of occluded segments, of *tennis ball*. There are two such explanations of length 2: one with segments #7 and #1 and another with segments #7 and #6 preserved, both of which are shown in Figure 5.7. We can also ask the tree for a rule explanation (root-to-leaf path) of *Labrador retriever* that results in the maximal possible confidence of the black-box model for this class. This explanation is  $f_0 = 0 \wedge f_1 = 0 \wedge f_2 = 1 \wedge f_3 = 0 \wedge f_4 = 1 \wedge f_5 = 1 \wedge f_6 = 1 \wedge f_7 = 0$ , giving us 99% confidence. This particular representation of our logical rule is not particularly appealing, however, as discussed in Section 5.5.2, we can express it in the visual domain as well – see Figure 5.8.

The biggest advantage of LIMETree is its ability to output *personalised counterfactual* explanations. For example, we can ask the following question: “Given segment #7 (the ball), what would have to change for the image to be classified as *golden retriever*?” Therefore, we are looking for an image modification with the ball segment (#7) preserved that is classified as *golden retriever*. LIMETree tells us that by occluding segments #1 and #6 – the smallest viable occlusion shown in Panel 5.9a – the model predicts *golden retriever* (91%). Since occluding segment #7, i.e., the ball, on its own results in 98% *golden retriever* (see Panel 5.6b), another interesting question is: “Had segment #7 not been there, can we revert the prediction to *tennis ball*?” LIMETree indicates that this is impossible, however when segments #7, #2 and #0 are occluded, the image is not predicted as *golden retriever* anymore – see Panel 5.9b.

## 5.5 Advantages of LIMETree

LIMETree is highly flexible, supports different types of explanations and comes with fidelity guarantees. By tailoring the interpretable representation to a particular data set or individual instance, the explanations can be customised even further. We explore the personalisation and



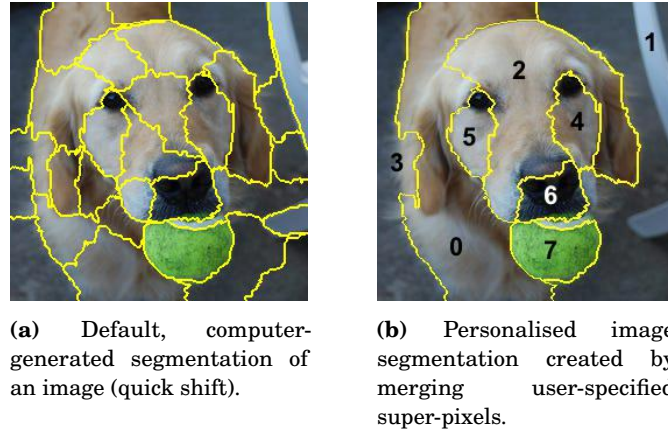
**Figure 5.9:** Customised (personalised) counterfactual explanations generated with LIMETree.

interactiveness of LIMETree explanations in Section 5.5.1, which demonstrates how to customise the interpretable representation, the explanation type and its content. Importantly, using a multi-output regression tree as the surrogate model enables accurate local mimicking of black-box probabilistic models simultaneously for multiple classes, making it appealing and compatible with modern predictors such as deep neural networks. LIMETree works equally well for data types other than images, e.g., tabular and text, and its full fidelity desideratum can be achieved in practice while preserving low complexity of explanations, which is discussed in Section 5.5.2. All of the LIMETree design choices empower the users to build an explainer that best fits a particular use case, targeting a wide range of stakeholders and purposes, for example, model debugging, robustness analysis, fairness evaluation and prediction explanation.

### 5.5.1 Personalised and Interactive Explainability

No matter how comprehensive an explanation is, it may not appeal to all explainees or exhaustively answer their diverse questions [151] as we show later in Chapter 6. Humans are accustomed to an explanatory process that entails interactive questioning, arguing and rebutting, which comes naturally in a conversation. Thus, for explanations of predictive systems to be intuitive, they should imitate this process [106]. LIMETree allows various aspects of its explanations to be interactively personalised, in particular the interpretable representation, type of an explanation and its content. This approach enables the explainees to steer the explanatory process in a selected direction, thereby achieving an explanation that satisfies their curiosity or answers specific questions.

**Interpretable Representation** The first step towards personalised surrogate explanations is tuning the interpretable representation of the data. While, in the case of images, computer generated segments (Panel 5.10a) may be good enough to produce meaningful explanations, we



**Figure 5.10:** Default (a) and custom (b) interpretable representations of an image. The top two classes predicted by a black box are 99.6% *tennis ball* and 0.4% *golden retriever*.

encourage the users to either provide custom segmentation or indicate which of the computer-generated segments should be merged (Panel 5.10b). This step aims to achieve an interpretable representation, i.e., image segmentation, that conveys meaningful *concepts*, which may be different for individuals with different levels of domain expertise and background knowledge. Similar reasoning applies to tabular and text data where the explainee can respectively customise binning of continuous features and tokenisation of sentences, e.g., bundle selected words to form a tuple considered as a single token in the interpretable representation [151]. After fixing the interpretable representation, a surrogate tree is fitted and its leaves relabelled as per Lemma 5.2, which model is then used to extract various explanations. Since personalised interpretable representations tend to be small in size (see Figure 5.10), a complete tree – according to Corollary 5.3 – can often be fitted, yielding more diverse and faithful explanations.

**Explanation Type** A surrogate based on linear regression is limited to explaining interpretable features (image segments) with their influence on a black-box prediction separately for each class. A multi-output regression tree, on the other hand, can explain the (local) behaviour of a black box with a wide range of high-fidelity artefacts discussed earlier in Section 5.1, namely:

- tree structure visualisation;
- interpretable feature importance;
- logical conditions;
- exemplar explanations;
- answers to what-if questions;
- contrastive statements; and
- supportive statements.

More importantly, beyond customising the interpretable domain, a linear surrogate is confined to static, one-off and one-size-fits-all explanations. In contrast, some of the decision tree explanations

can be framed in an interactive explanatory process, giving the explainees control over their content [151, 156] – a topic explored further in Chapter 6.

**Explanation Content** *Contrastive* and supportive statements (*counterfactuals* in particular) are a very prominent and appealing explanation type, which is arguably the most natural explanatory mechanism for humans [106] and complies with various legal regulations [173]. Such explanations can be simple “*Why?*” questions with either explicit or implicit class contrast, e.g., “Why is it a cat?” where the implicit contrast is interpreted as “Why is it a cat and not anything else?”, or “Why is it a cat and not a lion?” where the explainee explicitly provides a contrast. Additionally, the user can ask “*Why given?*” and “*Why despite?*” questions to also take control of and personalise the interpretable features appearing in the conditional part of the contrastive explanations. An explainee may prefer a counterfactual that, respectively, *must* and/or *must not* be conditioned on certain interpretable features, e.g., “Why is it a golden retriever and not a Labrador retriever, given occluded segment #3 and despite visible segments #1 and #6?”, which specifies both of these conditions and uses an explicit class contrast. Moreover, such explanations are capable of supporting interactions via an explanatory dialogue [151, 156] (Chapter 6), and are easy and efficient to derive from decision trees (Chapter 4). These observations generalise to LIMETree, which uses multi-output regression trees as its underlying surrogate model.

Another type of interactive explanations derived from a surrogate tree are answers to *what-if* questions: the explainee can formulate conditions on the features of a data point in an interpretable domain, e.g., image segments, and ask the tree for its prediction. For example, “What if segments #1 and #5 were occluded?”, which can be answered using either the black-box model or the surrogate tree depending on the desired fidelity of the explanation and completeness of the surrogate – see Section 5.3 for more details. Other, somewhat interactive, explanations are *decision rules*, i.e., root-to-leaf paths, and *exemplars*, i.e., similar data points. The first type allows the explainees to inspect the influence of each logical condition included in this path on the prediction. For example, in the image domain each root-to-leaf path could be visualised as the original image with a subset of segments occluded and the interactive interface would allow the explainee to click on each segment to switch its occlusion on or off, thereby changing the tree path, to understand its influence on the prediction. Similar interactive approaches can be developed for tabular and text data by allowing the explainee to change a value of a feature and add or remove a token from a sentence. Exemplar explanations, on the other hand, are generated by identifying all the data points in the interpretable representation that fall into the same and nearby, e.g., determined based on the Hamming distance, leaves of the surrogate tree. To better understand the local behaviour, the explainee can interactively select a leaf for which exemplars will be generated and specify whether these should be data points that are assigned the same or a different prediction to the one of the selected leaf.

Finally, the least interactive explanations are *tree structure visualisation* and interpretable *feature importance*, both of which can only be made interactive by embedding them in an

interactive user interface and are otherwise static. For example, the tree structure can be presented in an interface that allows the explaine to zoom in and out, thereby improving its comprehensibility by focusing only on one of its branches. This interface can also be a gateway to other, more interactive, explanations, e.g., selecting a leaf or a root-to-leaf path can give access to counterfactuals, exemplars and logical rules. Since all the seven types of explanations that we discussed in this section are derived from a single surrogate model, they are guaranteed to be coherent and their diverse nature should appeal to a wide range of explainees. Section 5.4 presented examples of these explanations, showcasing their power and the benefits of their interactivity.

### 5.5.2 Generalisability and Applicability

LIMEtree explanations are versatile and flexible but their fidelity guarantees require a *deterministic* interpretable representation transformation function  $IR$  – which has a unique inverse  $IR^{-1}$  (Lemma 5.2) – and a *complete* surrogate tree (Corollary 5.3), as outlined in Section 5.3.3. These conditions may seem strict and difficult to satisfy for a generic case, thereby hampering the adoption of LIMEtree, however in this section we show that these challenges can be easily overcome. We mainly focus on practical implications and requirements of our fidelity guarantees as many potential users will find this property the most appealing. We also discuss how to generalise LIMEtree to other data domains – tabular and text – while maintaining its core properties. We address concerns about the increased complexity of the surrogate tree and its adverse influence (or lack thereof) on the comprehensibility of the explanations, showing that this does not affect the most important types of explanations. All of these arguments should convince the reader that in many cases LIMEtree can be easily generalised and safely deployed while preserving its attractive explanatory properties.

#### Tabular and Text Data

This chapter largely focuses on explaining black-box probabilistic image classifiers, but in Section 5.3.1 we briefly discussed how surrogate explainers, such as LIMEtree, are also applicable to regression and binary or multi-class classification tasks. The core component in all of these use cases is the function responsible for transforming data between the original and *interpretable representations*. For images, we provided an example of this mechanism – a binary representation encoding super-pixel occlusions – analysed its properties and discussed its pros, cons and implications, showing how to design it to mitigate possible issues (Section 5.2.2). This overview led us to conclude that making the interpretable representation transformation function *deterministic* is crucial for guaranteeing full fidelity of LIMEtree – see Section 5.3.3.

A very similar line of reasoning applies to text data. Here, the most appealing interpretable domain is representing an excerpt of text as a bag of words (tokens), with the binary interpretable vector indicating their individual presence (1) or absence (0). This representation complies with all

of the properties discussed in Section 5.3.3 and required for LIMETree to achieve high fidelity. To ensure that the interpretable representation transformation function is deterministic, the order of words (tokens) in the text excerpt must be memorised, which is equivalent to remembering the adjacency of segments in an image and their occlusion colour. This interpretable domain for text has a major advantage over the one presented for images: it does not require an arbitrary protocol for removing words, akin to the occlusion colour for images, since they can be *explicitly removed* from the text (see the discussion in Sections 3.2.1 and 5.2.2). Searching for an interpretable representation for images with a similar property may be futile since for text it is an artefact of the black boxes rather than the interpretable domain itself – language models do not presuppose input of fix length or shape.

In contrast, defining an interpretable representation transformation function for tabular data with numerical features that is deterministic – hence has a unique inverse and complies with Lemma 5.2 – is challenging. The most popular approach [129] is discretisation followed by binarisation, e.g., a numerical feature  $x_3 = 7$  can be discretised into three intervals:  $(-\infty, -3]$ ,  $(-3, 8]$  and  $(8, \infty)$ , which are binarised as  $[0, 1, 0]$ , indicating that  $x_3$  falls into the middle bin (see Section 3.2.1 for more details). While this does not affect categorical features since they do not have to be discretised beforehand – their original representation can be uniquely reconstructed from the binary encoding – the same is not true for numerical attributes, making the transformation function *non-deterministic*. A number can be uniquely mapped to a bin as shown above, however the inverse procedure is ill-defined: reconstructing a number from a bin that spans a numerical range is impossible [158]. For example, LIME [129] bypasses this inverse by sampling from a truncated (at bin boundaries) Gaussian distribution fitted to each numerical bin, thereby introducing an additional source of randomness to the explanations.

While it is possible to use a surrogate explainer without an interpretable domain for tabular data, it becomes a fragile procedure and significantly alters the meaning of the explanations. For example, when the surrogate is a linear model (LIME’s approach), the explainer ceases to be a sensitivity analysis tool of interpretable features; instead the explanations convey the influence of raw attributes. In this case, dropping the interpretable representation also requires normalising all numerical features to the  $[0, 1]$  range and one-hot encoding the categorical attributes to ensure that the coefficients of the linear surrogate are comparable. Applying the interpretable representation comes with problems of its own; defining the right bin boundaries is non-trivial and requires a choice of an arbitrary algorithmic method, e.g., quartile discretisation. This can be partially addressed by allowing the user to interactively adjust the numerical bin boundaries and group categorical feature values as discussed in Section 5.5.1.

Depending on the surrogate model choice, coming up with an interpretable domain may be unnecessary altogether. Notably, decision trees learn their own discrete representation of tabular data by applying binary splits, thereby creating locally faithful and meaningful binning for continuous and grouping for categorical features [158] (see Section 3.3.2 for more details).

Furthermore, non-determinism of this interpretable representation transformation function can be overcome algorithmically by first *locally* sampling data within their original domain and then transforming them into the interpretable representation [158]. This is uniquely possible for tabular data and is the reverse of the standard procedure – steps 1–3 in Algorithm 5.1 corresponding to step 2 described in Section 5.2.1 – nonetheless, this strategy mitigates the need of using the ill-defined  $IR^{-1}$  function. Applying this “trick”, however, will not allow the surrogate to achieve full fidelity, which requires the interpretable domain transformation function to be deterministic (Lemma 5.2). Without satisfying this property it is also impossible to build a *complete* surrogate tree (Corollary 5.3), nevertheless since we are dealing with raw tabular data, we can overfit the tree to the local sample, thereby achieving high enough fidelity.

### Full Fidelity in Practice

Assuming that the interpretable representation transformation function satisfies the properties outlined in Lemma 5.2, i.e., it is deterministic and invertible, full fidelity of the surrogate is achieved in practice by adjusting the *sample size*  $n$  and relaxing the *complexity bound*  $\Omega$  of the tree (i.e., removing the depth constrain  $d$ ) in Algorithm 5.1. While the conditions underpinning Lemma 5.2 cannot be satisfied for tabular data with continuous features, reordering a few steps in the LIMETree algorithm provides a close approximation since the interpretable domain is learnt by the tree as discussed earlier. For image and text data, on the other hand, these requirements are easily met in practice, with Corollary 5.3 prescribing how to choose an appropriate sample size and tree depth bound to achieve full fidelity. For these data types, each dimension of the interpretable domain can be treated as a human-comprehensible concept, e.g., ears, eyes, muzzle and background for a dog image, which will often result in a relatively few concepts for each explained data point. Note that words (tokens) or image super-pixels do not have to be adjacent to be treated as a single entry in the interpretable domain, which, for example, allows to represent scattered background segments as one concept.

Following the logic presented in Section 5.3.3, a binary interpretable representation with 10 dimensions has  $2^{10} = 1024$  unique data points since the cardinality of a binary  $d$ -dimensional space  $\mathbb{B}^d = \{0, 1\}^d$  is equal to  $|\mathbb{B}^d| = 2^d$ . If we use all of these points (there is no benefit from oversampling) to train the local surrogate with its complexity bound  $\Omega$  relaxed to allow trees of depth 10, the model is guaranteed to achieve *full fidelity* – a complete, balanced binary tree of depth  $d$  has  $2^d$  leaves (its width), allowing one leaf per data point. The depth bound and the sample size can be adjust dynamically prior to training the local surrogate tree since the dimensionality of the interpretable domain is known beforehand. For every additional feature in the interpretable space, the number of sampled data points doubles and the tree depth is incremented by one in order to provide the interpretable domain and the surrogate tree with enough capacity to preserve the full fidelity guarantee. This exponential growth in the number of interpretable data points may seem overwhelming, however in our experience the number

of concepts is usually relatively small and training decision trees on binary data is fast. The exponential growth of the width of the surrogate tree increases its complexity and can have adverse effect on the intelligibility of some explanations, but it does not affect the most important and versatile explanation types as discussed below.

### **Preserving Low Complexity of Explanations**

Since a moderate number of interpretable features may yield a relatively large tree, one may worry about the increased complexity of the resulting explanations. After all, guaranteeing their full fidelity requires relaxing the depth bound of the surrogate  $\Omega$ , which the optimisation objective  $\mathcal{O}$  tries to minimise (Equation 5.1). While high complexity of a surrogate tree may render the explanations based on the tree structure, e.g., model visualisation, less comprehensible, these are not the most appealing explanation types and possibly require machine learning expertise to become intelligible in the first place. The interpretable feature importance, what-if explanations, exemplars, contrastive and supportive statements, on the other hand, are not affected by the tree complexity and remain highly interpretable, compact and accessible [156] with their interactive and customisable nature adding to their appeal as discussed in Section 5.5.1. The decision rules – logical conditions extracted from root-to-leaf paths – may indeed become overwhelmingly long, in fact as long as the tree depth, however this does not affect all the data types equally and the presentation medium can alleviate this issue regardless of the tree size.

Notably, rules generated for image and text data are always comprehensible regardless of their length. These rules cannot have more literals than the dimensionality of the interpretable domain, i.e., the number of segments for images and words or tokens for text. Presenting such a rule in the former case corresponds to displaying an image with various segments occluded (recall Figure 5.8) and in the latter producing a text excerpt with specific words or tokens removed. For tabular data, however, these rules may become relatively long and incomprehensible, with the exception of root-to-leaf paths that apply multiple conditions to a single feature, which allows to compress their length. In this case, visualisations are also not a viable alternative due to the inherent limitation of the human perceptual system to three dimensions, with an additional capacity enabled by considering time, e.g., when explaining a time series. Finally, a general criticism of rule-based explanations postulating that it is difficult to understand how each logical condition independently affects the prediction makes them less appealing than alternative tree explanation types.

In summary, if explanations based on the tree structure are not required for image and text data, and additionally rule-based explanations are not needed for tabular data, the complexity of the tree  $\Omega$  does not have to be controlled. In this case, the surrogate complexity measure  $\Omega(g)$  can be removed from the optimisation objective  $\mathcal{O}$  given in Equation 5.1 and the corresponding step (#7) in Algorithm 5.1 can be skipped, paving the way for full fidelity. It is worth mentioning that a *complete* surrogate tree will produce more counterfactual explanations for every data point,

thereby leaking information about the black-box model, which may be a trade secret [157].

## 5.6 Experimental Results

To demonstrate and assess the explanatory power of LIMETree we use a multi-tier evaluation approach that consists of *functionally-grounded* (Section 5.6.1) and *human-grounded* (Section 5.6.2) experiments [34]. The first involves a proxy task – numerically comparing the surrogate fidelity for different variants of LIME and LIMETree; the latter is a user study. For all of our experiments we used the pre-trained *Inception v3* neural network [164] distributed within PyTorch [116], with our surrogate explainers built on top of FAT Forensics [159, 160] (cf. Appendix B) using the bLIMEy algorithmic framework [158] (see Chapter 3).

### 5.6.1 Synthetic Validation

To understand how LIMETree behaves in various settings, we use a number of proxy metrics to experimentally evaluate its explanatory performance. First, we measure the faithfulness of the surrogate with respect to the black box, i.e., its ability to mimic the underlying predictor, which indirectly indicates the trustworthiness of its explanations. To this end, we report fidelity as measured by the LIME loss  $\mathcal{L}$  given in Equation 5.2 and the LIMETree loss  $\mathcal{L}$  defined in Equation 5.4. We do so when modelling the top three classes predicted by the black box for four different surrogate approaches: LIME and three variants of LIMETree. To complement the discussion presented in the previous section, we also analyse the complexity  $\Omega$  of LIMETree surrogates as defined in Equation 5.5, i.e., the depth of the tree normalised by the dimensionality of the interpretable domain, in relation to its fidelity.

**Surrogate Fidelity** We compare the fidelity of our method with a modified version of the LIME algorithm [159], which uses black as the occlusion colour and does not use feature selection, making it the most powerful variant of LIME since it has access to all of the interpretable features. The results presented in Tables 5.1 and 5.2 capture fidelity of three distinct LIMETree variants:

**LIMeT** a tree optimised for complexity, i.e., the shallowest tree that offers a certain level of performance;

**LIMET** a tree optimised for complexity, whose predictions are post-processed to guarantee full fidelity of model-driven explanations (see Section 5.3.2 for more details); and

**LIMET<sup>\*</sup>** a surrogate tree without complexity constraints, allowing the algorithm to learn complete trees with full fidelity.

$n^{\text{th}}$ top	<b>LIME</b>	<b>LIMEt</b>	<u><b>LIMEt</b></u>	<b>LIMEt*</b>
1 <sup>st</sup> class	$0.0172 \pm 0.0001$	<b><math>0.0070 \pm 0.0001</math></b>	$0.0144 \pm 0.0003$	<b><math>0 \pm 0</math></b>
2 <sup>nd</sup> class	$0.0056 \pm 0.0001$	<b><math>0.0027 \pm 0.0000</math></b>	$0.0045 \pm 0.0001$	<b><math>0 \pm 0</math></b>
3 <sup>rd</sup> class	$0.0029 \pm 0.0001$	<b><math>0.0012 \pm 0.0000</math></b>	$0.0029 \pm 0.0001$	<b><math>0 \pm 0</math></b>

**Table 5.1:** *Per-class* fidelity computed with the LIME loss (Equation 5.2) for different surrogate approaches (cf. Section 5.6.1). The results are based on explanations of 100 images for their top three predicted classes. (Lower is better.)

top $n$	<b>LIME</b>	<b>LIMEt</b>	<u><b>LIMEt</b></u>	<b>LIMEt*</b>
1 class	$0.0343 \pm 0.0004$	<b><math>0.0069 \pm 0.0001</math></b>	$0.0144 \pm 0.0003$	<b><math>0 \pm 0</math></b>
2 classes	$0.0227 \pm 0.0002$	<b><math>0.0026 \pm 0.0000</math></b>	$0.0045 \pm 0.0000$	<b><math>0 \pm 0</math></b>
3 classes	$0.0255 \pm 0.0002$	<b><math>0.0012 \pm 0.0000</math></b>	$0.0029 \pm 0.0001$	<b><math>0 \pm 0</math></b>

**Table 5.2:** Fidelity of the *top  $n$*  classes computed with the LIMETree loss (Equation 5.4) for different surrogate approaches (cf. Section 5.6.1). The results are based on explanations of 100 images for their top three predicted classes. When computing the LIMETree loss for one class, the factor of  $\frac{1}{2}$  is removed. (Lower is better.)

Table 5.1 reports the fidelity of the surrogates computed with the LIME loss (given in Equation 5.2) separately for each of the top three classes predicted by the black box. The LIME algorithm produces three independent linear surrogates – one for each class – while the variants of LIMETree give a single surrogate that models all of the classes simultaneously. Measuring the fidelity of each class separately helps us to capture the disparity of the probabilities predicted by the black box. Since the model is overconfident, the most probability mass is assigned to the top prediction, with the probabilities of the other two classes being much smaller. Similarly, Table 5.2 lists the fidelity of the surrogates computed with the LIMETree loss (given in Equation 5.4) for the top one, two and three classes predicted by the black box. Again, the LIME algorithm produces three independent linear surrogates – one for each class – whereas the variants of LIMETree give a separate model for the 1-class, 2-class and 3-class task. These results capture the mean fidelity of surrogates built to explain 100 random pictures from the ImageNet [33] validation set, computed over all the possible data points in the binary interpretable domain.

Both Tables 5.1 and 5.2 show that our base method – **LIMEt** – outperforms LIME. The LIMETree variant that achieves full fidelity for model-driven explanations (via prediction post-processing) – **LIMEt** – performs comparably to LIME when measuring fidelity using the LIME loss and outperforms it when the LIMETree loss is computed. The performance drop suffered by the latter approach is due to sub-optimal predictions made by the tree leaves for the majority of the interpretable space since this method is tuned to be faithful to the minimal interpretable data points representing the tree. The surrogate complexity  $\Omega$  of both LIMETree variants expressed as the proportion of interpretable features used by the tree is  $56 \pm 3\%$  on average, meaning that the surrogate requires only half of the interpretable features (i.e., half of the maximum depth) to achieve this level of performance. Finally, surrogate trees with unconstrained depth – **LIMEt\*** –

achieve full fidelity across the board, which is expected as these trees are expressive enough to cover the entire interpretable data space, creating one leaf for each data point if needed.

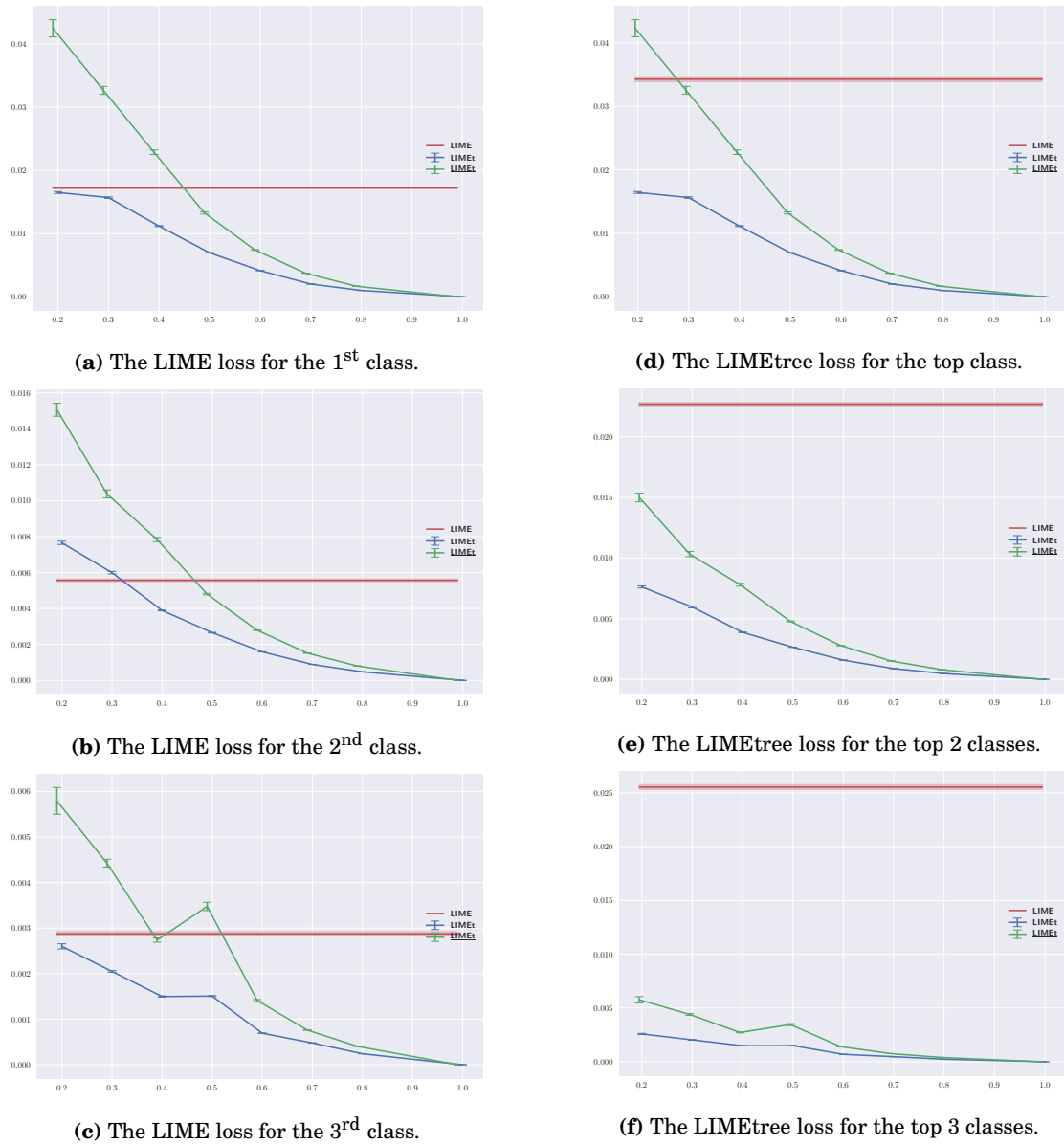
**Surrogate Complexity** Next, we investigate the relation between the depth-based complexity  $\Omega$  of a surrogate tree and its fidelity. Since various images may have different number of segments, i.e., interpretable features, our formulation of the tree complexity in Equation 5.5 accounts for that by scaling the tree depth according to the number of super-pixels, which can be interpreted as the tree completeness level. We compare this change in fidelity against a baseline achieved with the aforementioned configuration of LIME, which uses all of the interpretable features and occludes segments with a solid black colour. The empirical evidence provided by this study – visualised in Figure 5.11 – supports our discussion presented earlier in Section 5.5.2.

When using the LIME loss as our fidelity metric, **LIMEt** requires at most 33% and **LIMEt** needs at most 55% of all the interpretable features to perform on a par with LIME regardless of the number or configuration of the explained classes. For the LIMEtrees loss, **LIMEt** performs better than LIME with just 20% of interpretable features and **LIMEt** needs at most 30% of them. **LIMEt** requires deeper trees to achieve the same level of performance as **LIMEt** since the post-processing step applied to ensure full fidelity of model-driven explanations causes the surrogate to be a sub-optimal predictor for the majority of the interpretable data space. By allowing deeper trees we reduce the variance of their leaves, which improves the overall performance of the surrogates – a clear connection between the complexity  $\Omega$  of a tree and its fidelity. A more ambiguous dependency is between the surrogate complexity and the number of modelled classes, which affects the leaves impurity – visible in Panels 5.11d, e & f as different rates of convergence.

### 5.6.2 User Study

To assess the usefulness of LIMEtrees explanations in practice, we carried out a pilot user study. Our goal was to evaluate the potential impact of our method by comparing it to LIME [129], which is an established black-box surrogate explainer. Since in the pilot phase the study only allowed to serve non-interactive explanations, the participants were shown a surrogate tree, similar to the one in Figure 5.1, accompanied by a brief tutorial explaining how to obtain different kinds of explanations and their purpose. We recruited 8 participants (6 males and 2 females), evenly distributed across the 18–45 age group, 6 of whom had a background in machine learning, with 3 participants being familiar with ML explainability. The participants were not compensated for their involvement in the experiment.

The study consisted of two main sections – one devoted to LIME and one concerning LIMEtrees – displaying an image divided into three segments, with each partition enclosing a unique object, e.g., a cat, a dog and a ball. The two most applicable predictions of the black-box model for each object were explained with both methods and presented to the participants. For example, *tabby* and *tiger cat* for the cat object; *golden retriever* and *Labrador retriever* for the dog object; and



**Figure 5.11:** Fidelity of the surrogate (y-axis) plotted against the depth-based complexity of the tree (x-axis), i.e., the ratio between the tree depth and the number of interpretable features. The results are computed for the top three classes predicted by the black box. Panels (a), (b) & (c) depict the LIME loss (Equation 5.2); and Panels (d), (e) & (f) depict the LIMETree loss (Equation 5.4). Note the different scales on the y-axes.

*tennis ball* and *croquet ball* for the ball object. Thus in this case, the explaine was exposed to six LIME explanations, each one showing the influence of three image segments (one per object), and a single tree of depth three modelling all the six predictions. For each explainability method, the participants were asked about the expected behaviour of the black-box model in relation to any two out of the three displayed objects, totalling in six questions since the relations are

assumed to be non-reflective. For example, “How does the presence of *the cat object* affect the model’s confidence of the presence of *the dog object*?”, with three possible answers: confidence decreases, confidence not affected and confidence increases.

This particular question was chosen to avoid a bias towards either explainability method – we could neither ask for the importance of each object for a particular prediction (LIME) nor the influence of an object on a prediction, e.g., a counterfactual question (LIMEtree). Moreover, the participants were randomly assigned to one of two variants of the study, where they would either be exposed to LIME explanations first, followed by LIMEtree, or vice versa. We used this approach in conjunction with obfuscating the explainability method name to assess and account for any ordering and priming effects. Before viewing the explanations, the participants were asked to answer a similar set of questions using only their intuition. We used these answers to judge whether they still relied on their intuition when explicitly asked to work with explanations later.

Our findings show that regardless of the exposure order LIMEtree helped the participants to answer correctly 25% more questions as compared to LIME. The negligible overlap between the responses based on the participants’ intuition and for either of the two explainability methods shows that the participants based their answers on the explanatory evidence when instructed to do so. Despite the majority of the participants having a machine learning background, and some of them being familiar with XAI concepts, all of them found the process of manually extracting LIMEtree explanations challenging or daunting and rated the experience as either *difficult* or *very difficult*. This result was somewhat expected as LIMEtree explanations are meant to be interactive and a suite of suitable explanation presentation methods is needed to this end; however, despite poorly rated experience, LIMEtree explanations were still very insightful showing a great potential when presented to explainees within an intuitive interface. On the other hand, all of the participants indicated that using LIME explanations was either *easy* or *very easy*, which in conjunction with poor performance when compared to LIMEtree indicates that the participants were overconfident in their judgement of the quality and usefulness of LIME insights. Given all of these results, we conclude that LIMEtree explanations are promising and delivering them interactively instead of leaving this task up to the user will further improve our method’s success rate and overall user satisfaction.

## 5.7 Interactive and Surrogate Explainability Research

Our research on LIMEtree shows how to connect two important concepts from explainable AI and interpretable ML: *interactive* (dialogue-like) *explainability* and *surrogate explainers*. The former is often based on *contrastive* explanations (*counterfactuals*, in particular) since they occur naturally in human interactions [106]. Miller’s [106] foundational work in this area has summarised importance of such explanations, grounding them in social sciences, highlighting

the essential role they play in human explainability and showing the lack of consideration for human aspects in the current literature [107]. Following this observation, explainee-centred XAI and IML research has proliferated in the past years [169, 173], however another aspect of human-oriented explainability pointed out by Miller has largely gone unnoticed: their interactive, bi-directional and dialogue-like nature, which allows the explainee to guide the explainer, hence receive tailored explanations. Schneider and Handali [138] have recently reviewed an array of explainability approaches taking into consideration their interactivity, which led them to conclude that personalised explanations are generally unavailable.

While this is true for practical explainability approaches, extensive research has been undertaken to analyse theoretical properties of and frameworks designed to model explanatory interactions between two intelligent agents, be them humans, machines or one of each [10, 101, 174]. Weld and Bansal [178], on the other hand, discussed various properties of explanatory systems supporting user input and hypothesised how such interactions could look like in the real life, albeit focusing more on multiple explanatory modalities instead of explanation personalisation per se. A mixture of explainability and interactivity has also been used to refine (e.g., personalise) and improve some data modelling techniques. Kulesza et al. [81] adopted explanations of a naïve Bayes classifier to help the user *debug* and *personalise* classification of electronic mail and Kim et al. [69] showed how the users can personalise clustering results when they are given an explanation based on cluster centroids. Alternatively, otherwise static explainability approaches, such as partial dependence plots [44], were fitted into interactive user interfaces [76, 77] to provide the explainee with the freedom to explore these explanations. The following chapter focuses entirely on user-centred explainability, discussing the importance of interactive personalisation in AI and ML interpretability and showing how class-contrastive counterfactual explanations can be dynamically customised based on explainees' preferences [151, 156].

The second concept that our work builds upon is surrogate explainability [28, 129]: a model-agnostic and post-hoc technique that is compatible with any type of data (tabular, image and text). Surrogate explainers can either be used to explain an individual prediction by building a *local* surrogate, e.g., LIME [129], which makes use of a sparse linear regression; or to approximate the inner workings of an entire black-box model by building a *global* surrogate, e.g., TREEPAN [28], which is based on a decision tree. High modularity and flexibility of these explainers [158] (cf. Chapter 3) encouraged the research community to compose their different variant, some of which use decision trees as their local surrogate models [142, 158, 169]. For example, van der Waa et al. [169] showed how a local one-vs-rest classification tree can be used to produce contrastive explanations, and Shi et al. [142] fitted a local shallow regression tree and used its structure as an explanation. Both of these methods use local tree surrogates, however none of them utilises the full explainability (and interactivity) potential that they enable. Explainability of decision trees [156] and their ensembles [166] have also been investigated outside of the surrogate context, e.g., Tolomei et al. [166] proposed a method to explain predictions made by ensembles of decision

tree classifiers with class-contrastive counterfactuals. Similarly, in Chapter 4 we introduced CtreeX: a tree-specific algorithm for generating contrastive and supportive explanations of predictions made by classification and regression trees. The following chapter shows how to use this explainer to compose personalised counterfactual statements by interacting with a decision tree via a voice interface [156].

## 5.8 Forging a Human–Machine Link

This chapter introduced LIMETree: a *local* surrogate explainer of black-box *predictions* based on *multi-output regression trees*. We analysed properties of interpretable domains – which are required to make such explainers work with any type of data (image, text and tabular) – and showed how they can be designed and used to achieve the best explanatory performance, focusing on images but discussing text and tabular data as well. We then demonstrated how LIMETree improves upon LIME [129] by simultaneously modelling multiple classes and examined all the benefits of using surrogate trees with respect to the explanations that they produce. Next, we reviewed this diverse range of explanations and showed how some of them can be utilised in an interactive setting, thereby enabling their personalisation. We also provided various guarantees with respect to the local fidelity of surrogate trees, which we supported with a critical discussion and a guideline for operationalising these concepts. We showed examples of LIMETree explanations as well as evaluated our approach with quantitative experiments and a qualitative user study, all in the image classification domain.

With all of these properties, surrogate multi-output regression trees can be used to enhance transparency of black-box machine learning models in a way that feels natural to humans. Notably, our user study has highlighted the critical role of the delivery mechanism of explanatory artefacts. The participants of our experiment had to manually extract explanations from the structure of a surrogate tree rating this process as difficult, which prompted us to examine more natural (explanatory) interactions – a topic we explore in the next chapter. In particular, Chapter 6 investigates this interactive human aspect in a simplified setting, where predictions of a classification tree trained on tabular data are explained with class-contrastive counterfactual statements extracted with CtreeX. We deploy it in a (voice-driven) conversational agent, allowing the explainees to freely explore and personalise the explanations. We gauge the reception of such a system among domain experts and a lay audience, and identify user expectations and desiderata that should be considered to ensure explainee satisfaction.



## GLASS-BOX: ONE EXPLANATION DOES NOT FIT ALL

The need for transparency of predictive systems based on machine learning algorithms arises as a consequence of their ever-increasing proliferation in the industry. Whenever black-box predictions affect human affairs, the inner workings of these models should be scrutinised and their decisions explained to the relevant stakeholders, including their engineers, operators and the individuals whose case is being decided. While a variety of interpretability and explainability methods is available, none of them is a panacea that can satisfy all of the diverse expectations and competing objectives that may be required by the parties involved. We address this challenge by discussing the promises of *interactive* machine learning for improved transparency of black-box systems using the example of contrastive explanations – a state-of-the-art approach to *interpretable* machine learning.

Specifically, we show how to personalise counterfactual explanations by interactively adjusting their conditional statements and extract additional insights by asking follow-up “What if?” questions. Our experience in building, deploying and presenting this type of a system allowed us to list its desired properties as well as potential limitations, which can guide the development of interactive explainers. While dynamically customising the medium of interaction, i.e., the user interface comprised of various communication channels, may give an impression of personalisation, we argue that adjusting the explanation itself and its content is more important. To this end, properties such as breadth, scope, context, purpose and target of the explanation have to be considered, in addition to explicitly informing the explainee about its limitations and caveats – a subset of properties borrowed from our XAI taxonomy introduced in Chapter 2. Furthermore, we discuss the challenges of mirroring the explainee’s mental model, which is the foundation of intelligible human–machine interactions. We also deliberate on the risks of allowing the explainee to freely manipulate the explanations as it facilitates extracting information about

the underlying predictive model, which may be leveraged by malicious actors to steal or game it. Finally, building an end-to-end interactive explainability system is an engineering challenge; unless the main goal is its deployment, we recommend “Wizard of Oz” studies as a proxy for testing and evaluating standalone *interactive* explainability algorithms.

## 6.1 Interactive Explainability

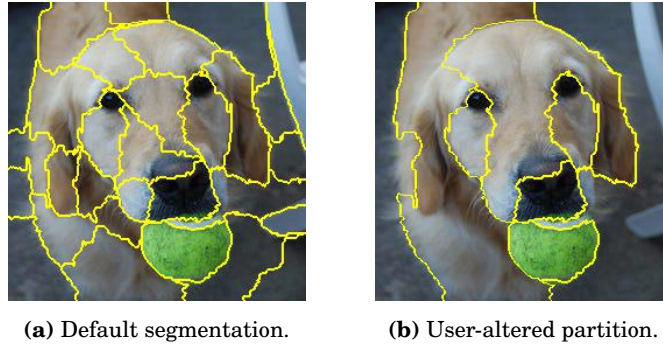
Given the opaque, black-box nature of complex machine learning systems, their deployment in high-stakes domains is limited by the extent to which they can be interpreted or validated. In particular, predictions, (trained) models and (training) data should be accounted for. One way to achieve this is with “transparency by design”, so that all components of a predictive system are “glass boxes”, i.e., ante-hoc explainability [133]. Alternatively, transparency may be obtained with post-hoc tools, which have the advantage of not limiting the choice of a predictive model in advance [129]. The latter approaches can either be model-specific or model-agnostic [131].<sup>1</sup> Despite this wide range of available tools and techniques, many of them are non-interactive, providing the explainee (explanation recipient) with a single insight that has been optimised according to some predefined metric. While a number of these methods simply cannot be customised by the end user without an in-depth understanding of their inner workings, others can take direct input from explainees with a varying level of domain expertise: from a lay audience – e.g., selecting regions of an image in order to query their influence on a classification outcome – to domain experts – e.g., tuning explanation parameters such as the underlying distance function. A particular risk of a lack of interaction and personalisation mechanisms is that the explanation may not always align with users’ expectations, thus reducing its overall value and usefulness.

Allowing the user to guide and customise an explanation, e.g., by adjusting its content and complexity, can benefit the transparency of a predictive system by making it more suitable and appealing to the explainee. Therefore, personalisation can be understood as influencing an explanation or an explanatory process to answer user-specific questions. For counterfactual explanations of the form: “had feature  $x_i$  been different, the prediction of the model would have been different too”, these can be user-defined constraints on the number and type of features that are allowed and prevented from appearing in the conditional statement. Delegating the task of customising and personalising explanations to the end user via interaction mitigates the need for the difficult process of capturing the user’s mental model beforehand, rendering the task feasible and making the whole process feel more natural, engaging and less frustrating.

In human interactions, understanding is naturally achieved via an *explanatory dialogue* [106], possibly supported with visual aids. Mirroring this explanatory process for ML transparency would make it attractive and accessible to a wide audience. Furthermore, allowing the user to customise explanations extends their utility beyond ML transparency. The explainee can steer

---

<sup>1</sup>Ante-hoc explainability, on the other hand, is predominantly model-specific.



**Figure 6.1:** Surrogate explainers of image classifiers require an interpretable representation, such as super-pixel segmentation, to effectively communicate the explanation to the user. These explainers try to identify portions of an image that influence its classification the most, i.e., segments of high positive or negative importance. Since the default outcome of image segmentation (a) may be unintuitive, we encourage the explainee to personalise the segmentation (b), e.g., by merging its elements, such that it represents (semantically) meaningful concepts.

the explanatory process to inspect fairness (e.g., identify biases towards protected groups<sup>2</sup>) [83], assess accountability (e.g., identify model errors such as non-monotonic predictions with respect to monotonic features) [95] or debug predictive models [81, 157]. In contrast to ML tasks [66] – where any interaction may be impeded by human-incomprehensible internal representations utilised by a predictive model – interacting with explainability systems is feasible as the employed representation has to be human-understandable in the first place, thereby enabling a bi-directional *communication*. Interaction with explanatory systems also allows incorporating new knowledge into the underlying ML algorithm and building a proxy of the explainee’s mental model, which will help to customise the resulting explanations down the line.

Consider the example of explaining an image with a local surrogate method that relies upon segmentation, e.g., the LIME algorithm introduced by Ribeiro et al. [129]. While super-pixel discovery may be good at separating colour patches based on their edges, these segments often do not correspond to semantically meaningful *concepts* such as ears or a tail in a dog picture – see Figure 6.1 for an example. The resulting explanation, nonetheless, can be personalised by allowing the explainee to *merge* and *split* segments before analysing their influence on the output of a black-box model, thereby implicitly addressing the doubt that prompted the explainee to alter the segmentation. User input is a welcome addition given the complexity of images; a similar approach is possible for tabular and text data, although user input carries less value in these two cases. For tabular data, the explainee may select certain feature values that are of interest or create meaningful binning for some of the continuous attributes; for text data (treated as a bag of words), the user may group some words into a token that conveys the correct meaning in that particular sentence. This exchange of knowledge between the explainee and the explainability

<sup>2</sup>A *protected group* is a sub-population in a data set created by fixing a value of a protected attribute such as *age*, *gender* or *ethnicity*, which discriminating upon is illegal.

system can considerably increase the quality of explanations, but also poses significant safety, security and privacy risks. A malicious actor may abuse such a system to uncover sensitive data used to train the underlying predictive (or explanatory) model, extract proprietary model components, or learn its behaviour in an attempt to game it (see Section 6.2.2 for a discussion).

After Miller’s [106] seminal work (inspired by explanation research in the social sciences) drew attention to the lack of human-aspect considerations in the explainable artificial intelligence literature – with many such systems being designed by the technical community for the technical community [107] – researchers started acknowledging the end user when designing XAI solutions. While this has advanced human-centred design and validation of explanations produced by XAI tools, another of Miller’s insights received relatively little attention: the interactive, dialogue-like nature of explanations. Many of the state-of-the-art explainability approaches are static, one-off systems that do not take user input or preferences into consideration beyond the initial configuration and parameterisation [44, 48, 100, 129, 130, 173].<sup>3</sup> While sometimes the underlying explanatory algorithms are simply incapable of a meaningful interaction, others do apply a technique or utilise an explanatory artefact that can support it in principle. Part of this trend can be attributed to the lack of a well-defined protocol for appraising interactive explanations and the challenging process of assessing their quality and effectiveness, which – in contrast to one-shot evaluation – is a software system engineering challenge<sup>4</sup> and requires time- and resource-consuming user studies.

Schneider and Handali [138] noted that bespoke explanations in AI – achieved through interaction or otherwise – are largely absent within the existing literature. Research in this space usually touches upon three aspects of “personalised” explanations. First, there are interactive machine learning systems where the user input is harnessed to improve performance of a predictive model or align the data processing with its operator’s prior beliefs. While the classic active learning paradigm dominates this space, Kulesza et al. [81] designed a system that presents its users with classification explanations to help them refine and personalise the predictive task, hence focusing the interaction on the underlying ML model and not the explanations. Similarly, Kim et al. [69] introduced an interactive ML system with an explainability component, allowing its users to alter the data clustering based on their preferences. Second, the work of Krause et al. [77] and Weld and Bansal [178] focused on interactive (multi-modal) explainability systems. Here, the interaction allows the explainees to elicit more information about an ML system by receiving a range of diverse explanations derived from a collection of XAI approaches such as Partial Dependence [44] and Individual Conditional Expectation [48] plots. While this body of research illustrates what such an interaction (with multiple explanatory modalities) may look like and

---

<sup>3</sup>To clarify, the notion of interaction is with respect to the explanation, e.g., the ability of the explainees to personalise it, and not the overall interactiveness of the explainability system.

<sup>4</sup>Building such systems requires a range of diverse components: user interface, natural language processing unit, natural language generation module, conversation management system and a suitable and well-designed XAI algorithm. Furthermore, most of these components are domain-specific and cannot be generalised beyond the selected data set and use case.

persuasively argues its benefits [178], the advocated interaction is mostly with respect to the presentation medium itself – e.g., an interactive PD plot – and cannot be used to customise and personalise the explanation per se. Third, Madumal et al. [101] and Schneider and Handali [138] developed theoretical frameworks for interactive, personalised explainability that prescribe the interaction protocol and design of such systems. However, these theoretical foundations have not yet been employed to conceptualise and implement an interactive explainability system coherent with the XAI desiderata outlined by Miller [106], which could offer customisable explanations. A more detailed overview and discussion of the relevant literature is presented in Section 6.4.

In contrast, this chapter proposes an architecture of a truly interactive explainability system, demonstrates how to build it, analyses its desiderata and examines how a diverse range of explanations can be personalised (Section 6.2). Additionally, we discuss lessons learnt from presenting our prototype to both a technical and a lay audience, and outline a plan for future research in this direction (Section 6.3). For our first attempt to build an XAI system that allows the explainee to customise and personalise the explanations we decided to use a *classification tree* as the underlying predictive model. This choice simplifies many steps of our initial study, allowing us to validate (and guarantee correctness of) the explanations and reduce the overall complexity of the explanation generation and tuning process by simply inspecting the structure of our decision tree, all of which is facilitated by CtreeX (cf. Chapter 4). Using *ante-hoc* explanations derived from a single predictive model also allows us to mitigate engineering challenges that come with combining multiple independent XAI algorithms as proposed by Weld and Bansal [178]. Furthermore, a decision tree can provide a wide range of diverse explanation types, many of which can be customised and personalised. Specifically, for global model explanations we provide

- *model visualisation*, and
- *feature importance*;

whereas for prediction explanations we rely on

- *decision rule* – extracted from a root-to-leaf path,
- *exemplar* – a similar (training) data point extracted from the tree leaves,
- *what-if* – an answer to a “What if?” question,
- *supportive statement* – achieved by generalising logical conditions imposed on tree leaves, and
- *contrastive explanation* – retrieved by comparing decision rules for different tree leaves.

When presented to the user, all of these explanations span a wide range of explanatory artefacts in visual (image) and textual (natural language) domains, thereby allowing us to test the extent to which they can be interactively personalised. For our prototype, we focus on contrastive

explanations, in particular class-contrastive *counterfactual* statements, which are the foundation of our system. These take the form of: “had *one of the attributes been different in a particular way*, the classification outcome would have changed as follows...” Arguably, they are the most suitable, natural and appealing explanations targeted at humans [106, 173]. In addition to all of their desired properties grounded in the social sciences [106] and legal considerations [173], they can be easily adapted to an interactive dialogue aimed at their personalisation, which, notably, is not widely utilised. In our system, they are delivered in this exact format – a natural language conversation, which is the most intuitive explanatory mechanism [106]. In summary, our prototype encompasses a holistic and diverse interactive XAI system where the interaction is focused on **personalising** explanations (in accordance with Miller’s [106] notion of XAI interactivity) as opposed to simply building an XAI system that delivers explanations interactively (to explain different aspects of a black-box system using a range of XAI algorithms) – a subtle but significant distinction.

## 6.2 Conversational Explanations

As a first step towards interactive XAI systems capable of outputting personalised explanations we developed **Glass-Box** [156]: a class-contrastive counterfactual explainer that can be queried within a natural language dialogue (described in Section 6.2.1). It supports a range of “Why?” questions that can be posed either through a voice- or chat-based interface. Building this prototype and testing it in the wild provided us with invaluable experience and insights, which we report here to aid similar development and deployment efforts – Sections 6.2.2 and 6.2.3 respectively discuss *desiderata* and *properties* of interactive explainers. The feedback that helped us to refine our idea of responsive XAI systems delivering personalised explanations (presented in Section 6.2.4) was collected while demonstrating Glass-Box to a diverse audience consisting of both domain experts, approached during the 27<sup>th</sup> International Joint Conference on Artificial Intelligence (IJCAI 2018), and a lay audience, who visited a local “Research without Borders” festival<sup>5</sup> that is open to the public and attended by pupils from local schools. While at the time of presentation our system was limited to class-contrastive counterfactual explanations personalised by choosing data features upon which the counterfactual statements should and should not be conditioned, and provided to the user in natural language, we believe that our observations remain valid beyond this particular XAI technique. We hope to test this assumption in our future work – see Section 6.3 for more details – by employing the remaining decision tree explainability modalities listed in the introduction, albeit in an XAI system refined based on our experience to date.

---

<sup>5</sup>The festival spans a wide range of research projects both in social sciences and engineering.

### 6.2.1 Glass-Box Design

Glass-Box has been designed as a piece of hardware built upon the Google AIY (Artificial Intelligence Yourself) Voice Kit<sup>6</sup> – a customisable hardware and software platform for development of voice interface-enabled interactive agents. The first prototype of Glass-Box utilised the Amazon Alexa skill API, however the limitations of this platform at the time (the processing of data had to be deployed to an on-line server and invoked via an API call) hampered our progress and prompted us to switch to the aforementioned Google AIY Voice Kit. These recent technological advancements in automated speech-to-text transcription and speech synthesis (which are offered as a service) allowed us to utilise an off-the-shelf, voice-enabled, virtual digital assistant to process explainees’ speech and read out answers to their questions – something that would not have been feasible had we decided to build this component ourselves. We extended the voice-driven user interface with a (textual) chat-based web application that displays the transcription of each conversation and its history – to improve accessibility of the system, among other things – in addition to allowing the explainee to type in the queries instead of saying them out loud.

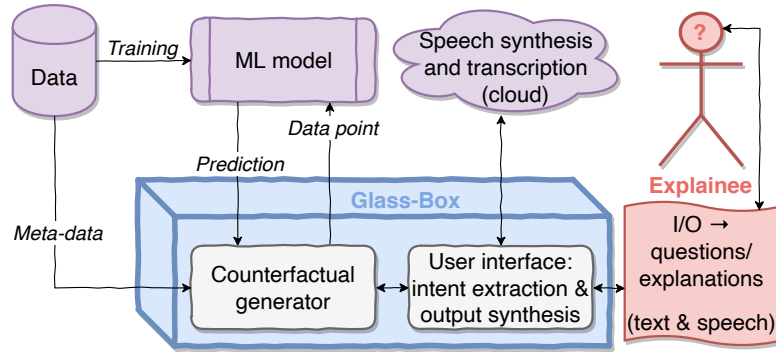
To avoid a lengthy and possibly off-putting process of submitting (mock) personal details, i.e., a data point, to be predicted by the underlying machine learning model and explained by Glass-Box, we opted for a predefined set of ten instances. Any of them could be selected and submitted to Glass-Box by scanning a QR code placed on a printed card that also listed details of this fictitious individual. Once a data point is chosen, the explainee can alter personal details of the selected individual by interacting with Glass-Box, e.g., “I am 27 years old, not 45.” Any input to the system is passed to a natural language processing and understanding module built using *rasa*<sup>7</sup>. Our prototype of the Glass-Box system was based on the *UCI German credit* data set<sup>8</sup> (using a subset of its features) for which a decision tree classifier was trained with *scikit-learn* [120]. Since the German credit data set has a binary target variable (“good” or “bad” credit score), the class contrast in the resulting counterfactual explanations is implicit. Nonetheless, this restriction can be easily overcome and the system generalised to a multi-class setting by requiring the explainee to explicitly specify the contrast class, taking the second-most likely one or providing one explanation per class. A conceptual design of Glass-Box is shown in Figure 6.2.

To facilitate some of the user interactions, the data set had to be manually annotated. This process allowed the generation of engaging natural language responses and enabled answering questions pertaining to *individual fairness* of black-box predictions. The latter functionality was achieved by indicating which features (and combinations thereof) should be treated as protected attributes – had a counterfactual data point conditioned on one of these features been found, Glass-Box would indicate unfair treatment of this individual. This functionality could be invoked by asking an “Is the decision fair?” question and further interrogating the resulting counterfactual

<sup>6</sup><https://aiyprojects.withgoogle.com/voice>

<sup>7</sup><https://github.com/RasaHQ/rasa>

<sup>8</sup>[https://archive.ics.uci.edu/ml/datasets/statlog+\(german+credit+data\)](https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data))



**Figure 6.2:** Glass-Box design and information flow.

explanation if unfair treatment was identified. Depending on the explainability and interaction capabilities expected of the system, other data set annotations may be necessary. Since this augmentation process is a predominantly manual task, it can be time- and resource-consuming.

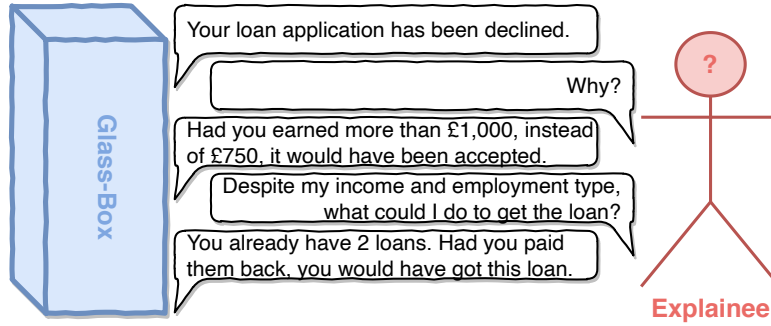
As noted before, the main objective of Glass-Box is to provide the users with personalised explanations whenever they decide to challenge the decision of the underlying machine learning model. The explainee can request and interactively customise the resulting counterfactual explanations through a natural language interface with appropriate dialogue cues. This can be done in three different ways by asking the following questions:

**“Why?”** – a plain counterfactual explanation – the system returns the shortest possible class-contrastive counterfactual;

**“Why despite?”** – a counterfactual explanation not conditioned on the indicated feature(s) – the system returns a class-contrastive counterfactual that does not use a specified (set of) feature(s) as its condition; and

**“Why given?”** – a (partially-)fixed counterfactual explanation – the system returns a class-contrastive counterfactual that is conditioned on the specified (set of) feature(s).

By repetitively asking any of the above “Why?” questions, the system will enumerate all the possible explanations with the condition set (the feature values that need to change) increasing in size until no more counterfactuals can be found. It is also possible to mix the latter two questions into “Why given ... and despite ...?”, thereby introducing even stronger restrictions on the resulting explanations. In addition to “Why?” questions, the explainee can also ask **“What if?”** In this case it is the user who provides the contrast and wants to learn the classification outcome of this hypothetical instance. Such a question can be either applied to the selected data point (which is currently being explained) or any of the counterfactual data points offered by the system as an explanation. All of these requirements imposed by the user are processed by a simple logical unit that translates them into constraints applied to the set of features that the counterfactual



**Figure 6.3:** Example explanatory conversation between Glass-Box and an explainee who personalises the explanations by asking counterfactual questions.

is allowed and/or required to be conditioned upon. All of this is facilitated through a natural language dialogue, an example of which is presented in Figure 6.3.

The method employed to generate counterfactual explanations from the underlying decision tree classifier relies upon the CtreeX algorithm (cf. Chapter 4) configured with a bespoke leaf-to-leaf distance metric. It allows to find leaves of classes that are different to the one assigned to the selected instance while requiring the fewest possible tweaks to its feature values. One self-evident solution to this problem is the neighbouring leaf, which must be of a different class and requires just one feature to be altered. However, there may as well exist a counterfactual leaf that is relatively distant in the decision tree structure but also requires just one feature value change, for example when such two decision tree paths share relatively few features. To identify both of these distinct cases, our distance metric is computed on a *meta-feature* space (an alternative representation of the tree structure) that is determined by all the unique feature partitions extracted from the splits of the decision tree. Finally, an  $L_1$ -like metric (when a particular feature is present on one branch and absent on the other, this distance component is assumed to be 0) is calculated and minimised to derive a list of counterfactual explanations ordered by their length – see Section 4.3 for an in-depth description of this procedure.

### 6.2.2 Explanation Desiderata

During the development stage and early trials of Glass-Box we identified a collection of *desiderata* and *properties* that should be considered when building such systems. Some of these characteristics are inspired by relevant literature [81, 106, 138, 178], while others come from our experience gained in the process of building the system, presenting it to various audiences, discussing its properties at different events and collecting feedback about interacting with it. While this and the following sections focus exclusively on desiderata for interactive and customisable explanations, these features are only a subset of a more comprehensive XAI taxonomy that we introduced earlier in Chapter 2. The relevant selection of these desiderata are summarised in Table 6.1 as well as collected and discussed below. Section 6.2.3, on the other hand, examines the properties of

Functional	Operational	Usability
<b>F3</b> Explanation Target	<b>O7</b> Function of the	<b>U3</b> Contextfullness
<b>F4</b> Explanation	Explanation	<b>U6</b> Chronology
Breadth/Scope	<b>O8</b> Causality vs.	<b>U7</b> Coherence
<b>F7</b> Relation to the Predictive	Actionability	<b>U8</b> Novelty
System		<b>U9</b> Complexity
		<b>U10</b> Personalisation
		<b>U11</b> Parsimony

**Table 6.1:** Summary of a subset of the XAI taxonomy (cf. Chapter 2) applicable to interactive explainers that support personalisation. (See Section 6.2.2 for a comprehensive discussion of these properties.)

interactive explainability systems.

Given the complex nature of such systems, it is to be expected that some of these objectives might be at odds with each other, their definitions may be “fuzzy”, they might be difficult to operationalise, their “correct” application might depend on the use case, etc. Similarly, striking the right balance between these desiderata can be challenging. Nonetheless, we argue that simply considering them while designing interactive explainers will improve the overall quality of the system, help the creators and users understand their strengths and limitations, and make the interaction feel more natural to humans. Furthermore, some of these desired properties can be achieved (and “optimised” towards the explainees) by simply allowing user interaction, thereby alleviating the need for baking them explicitly into the system. For example, interactive *personalisation* of the explanations – on-line, via user input – is an attractive alternative to solving this challenge off-line, which would require a dedicated algorithm.

The main advantage of Glass-Box interactivity is the explainee’s ability to transfer knowledge onto the system, specifically various preferences with respect to the desired explanations, which are used to *personalise* them (**U10**, see Table 6.1). In our experience, customisation can come in many different shapes and forms, some of which are discussed below. For one, by interacting with the system the explainee should be able to adjust the *breadth and scope* of an explanation (**F4**). Given the complexity of the underlying predictive model, the explainee may start by asking for an explanation of a *single data point* (black-box prediction) and continue the interrogation by generalising it to an explanation of a *data subspace* (cohort) with the final stage entailing an explanation of the entire black-box model. Such a shift in the explainee’s focus may require the explainability method to adapt and respond by changing the *target* of the explanation (**F3**). The user may request an explanation of a single data point or a summary of an entire data set (training, test, validation, etc.), but also an explanation of a predictive model (or its subspace) or any number of its predictions. Furthermore, interactive personalisation of an explanation can increase the overall versatility of such systems as bespoke insights may serve a variety purposes

and have different *functions* (**O7**). An appropriately phrased explanation may be used as evidence that the system is *fair* – either with respect to a group or an individual depending on the scope and breadth of the explanation – or that it is *accountable*, which again can be investigated with varied scope, for example, a “What if?” question uncovering that two perceptually indistinguishable data points yield significantly different class predictions, such as adversarial examples [49]. Notably, if the explainer is flexible enough and the interaction allows such customisation but the explanations were designed to serve only one purpose, e.g., transparency, the explainee should be explicitly warned of such limitations to avoid any unintended consequences. For example, the explanations may be counterfactually actionable but they are not causal since they were not derived from a proper causal model (**O8**).

Some of the aforementioned principles can be observed in how Glass-Box operates. The contrastive statements about the underlying black-box model can be used to assess its transparency (their main purpose), fairness (disparate treatment via contrastive statements conditioned on protected attributes) and accountability (e.g., answers to “What if?” questions that capture unexpected non-monotonic behaviour). Glass-Box explanations are personalised via user-specified constraints imposed on the conditional part (foil) of the counterfactual statements, and by default they are specific to a single prediction. However, cohort-based insights can be retrieved by asking “What if?” questions pertaining to counterfactual explanations generated by Glass-Box; Section 6.3 discusses how the scope and the target of our system can be interactively broadened to include global explanations of the black-box model. Given the wide range of possible explanations and their diverse purposes, some systems may produce contradictory or competing insights. Glass-Box is less prone to such issues as the employed explainer is ante-hoc (**F7**), i.e., predictions and explanations are derived from the same ML model, hence they are always truthful with respect to it. This means that contradictory explanations are indicative of flaws in the underlying predictive model, in which case they can be very helpful in improving its accountability.

In day-to-day human interactions we are able to communicate effectively and efficiently because we share common background knowledge about the world that surrounds us – a mental model of how to interact with the world and each other [79]. Often, human-machine interactions lack this implicit link, causing the entire process to feel unnatural and frustrating. Creators of interactive explainability techniques should therefore strive to make their systems *coherent* with the explainee’s mental model to mitigate this phenomenon as much as possible (**U7**). While this objective may not be feasible in general, modelling a part of the user’s mental model, however small, can make a significant difference. The two main approaches to extracting the explainee’s mental model are: interactive querying of the user in an iterative dialogue (on-line), or embedding the explainee’s characteristics and preferences in the data or in the parameters of the explainer (off-line), both of which are discussed in Section 6.4.

For explainability systems, capturing the explainee’s mental model is possible to some extent as the purpose and behaviour of such tools are limited in scope; especially in comparison to more

challenges tasks within this domain, e.g., building a *generic* virtual personal assistant. Designers of explainability algorithms should also be aware that many interactions rely upon implicit assumptions that are embedded in the explainees’ mental model and perceived as mundane, hence not voiced, for example, the context of a (Glass-Box) *follow-up* question. Importantly, within human–machine interactions the *context* and its dynamic changes can be much more subtle, which may cause the coherence between the internal state of an explainer and the explainees’ mental model to diverge abruptly (U3). This issue can be partially mitigated by explicitly grounding explanations in a context at certain stages, for example, whenever the context shifts, which will help the users to adapt by updating their mental model and any assumptions. Contextfulness will also help the explainees better understand the limitations of the system, e.g., whether an explanation produced for a single prediction can (or must not) be generalised to other (similar) instances: “Please note that this explanation can be generalised to other data points that – in relation to the explained instance – have all of their feature values the same with the exception of attribute  $x_5$ , which can span the  $0.4 \leq x_5 < 1.7$  range.”

Regardless of how interactive the system is, the explanations should strive to be *parsimonious* – as short as possible but no shorter than necessary – to convey the required information without overwhelming the explainees (U11). Maintaining a mental model of the user can help to achieve this objective as the system can provide the explainees only with *novel* explanations – accounting for factors that are unfamiliar to the user – therefore reducing the amount of information carried by the explanation (U8). Further user-centred aspects of an explanation are its *complexity* and *granularity* (U9). The former should be adjusted according to the depth of the technical knowledge expected of the envisaged audience, and the latter chosen appropriately to the intended use of the explanation. This can either be achieved by design (i.e., incorporated into the explainability technique), be part of the system configuration and parameterisation step (off-line), or adjusted interactively by the user as part of the explanatory dialogue (on-line). Another aspect of an explanation, which is often expected by humans [106], is the *chronology* of factors presented therein as the explainees anticipate to hear more recent events first (U6). When the underlying data set supports it, the explainees should be given the opportunity to trace the explanation back in time, which can be easily facilitated through interaction.

Glass-Box attempts to approximate its users’ mental models by mapping their interests and interaction context (inferred from regularly asked questions) to data features that are used to compose counterfactual explanations. Memorising previous interactions, their sequence and the frequency of features invoked by the user help to achieve this goal and avoid repeating the same answers – once all of the explanations satisfying given constraints are presented, the system explicitly states this fact. Contextfulness of our explanations is also based on user interactions; it is implicitly preserved for follow-up queries (within an interrogative dialogue) that are initiated by the user and do not affect the context. Whenever the assumptions shift – e.g., a new personalised explanation is requested by the user or an interaction is triggered

Operational	Usability	Safety
<b>O1</b> Explanation Family	<b>U4</b> Interactiveness	<b>S3</b> Explanation Invariance
<b>O2</b> Explanatory Medium		
<b>O3</b> System Interaction		
<b>O4</b> Explanation Domain		
<b>O5</b> Data and Model Transparency		
<b>O6</b> Explanation Audience		
<b>O10</b> Provenance		

**Table 6.2:** Summary of a subset of the XAI taxonomy (cf. Chapter 2) specifically applicable to Glass-Box. (See Section 6.2.3 for a comprehensive discussion of these properties.)

by Glass-Box – the new context is explicitly communicated to the explainee. While contrastive statements are inherently succinct, their lack of parsimony could be observed for some Glass-Box explanations, which took the form of a long “monologue” delivered by the device. In most of the cases, this was caused by the system “deciding” to repeat the personalisation conditions provided by the user to ensure their coherence with the explainee’s mental model.

Glass-Box is capable of producing novel contrastive explanations by conditioning them on features that have not yet been acknowledged by the user during the interaction. Notably, there is a trade-off between novelty of explanations and their coherence with the user’s mental model, which we have not explored when presenting our system but which should be navigated carefully to avoid jeopardising explainee’s trust. Glass-Box was built to explain predictions of the underlying ML model and did not account for possible generalisation of its explanations to other data points (the users were informed about this limitation prior to interacting with the device). However, the explainees can ask “What if?” questions with respect to the counterfactual explanations, for example using slight variations of the explained data point, to explicitly check whether their intuition about broader scope of an explanation holds up. Finally, chronology was not required of Glass-Box explanations as the data set used to train the underlying predictive model does not have any time-ordered features.

### 6.2.3 Glass-Box Properties

In addition to a set of desiderata for interactive explainability systems, we review a number of their general properties and requirements that should be considered prior to their development. These are summarised in Table 6.2 and discussed below.

Assuming that the system is interactive, the *communication protocol* between the explainee and the explainer should be carefully chosen to support the expected input and deliver the explanations in the most natural way possible. For example, clearly indicating which parts of the explanation can be personalised and the limitations of this process should be disclosed to

the user (**O3**, see Table 6.2). The choice of *explanatory medium* used to convey the explanation is also crucial. Plots, interactive or not, can be very informative but may not communicate the entire story due to the curse of dimensionality and the inherent limitations of the human visual system (**O2**). Supporting visualisations with a textual description, and vice versa, can greatly improve their intelligibility, nonetheless relying entirely on natural language explanations may be sub-optimal in certain cases, for example, explaining images exclusively via a chat interface. The *intended audience* should be considered in conjunction with the communication protocol to choose a suitable explanation type (**O6**). Domain experts may expect explanations expressed in terms of the internal parameters of the underlying predictive model, but a lay audience may rather prefer exemplar explanations that rely on relevant data points – choosing the appropriate *explanation domain* (**O4**). The audience also determines the purpose of the explanation. For example, inspecting a predictive model for debugging purposes will need a different system than guiding the explainee with an actionable advice towards a certain goal such as receiving a loan. Interactive explainers can support a wide spectrum of these desirable properties simply by allowing the explainee to dynamically personalise the output of the explainer as discussed in Section 6.2.2.

Achieving some of these objectives may require the features of the underlying data set or the predictive model itself to be *transparent* (**O5**). For example, consider explaining a model trained on a data set whose features are measurements of an object given in metres as opposed to magnitudes of some embedding vectors. When the raw features (original domain) are not human-intelligible, the system designer may decide to use an interpretable representation (transformed domain) instead to aid the explainee. Additionally, providing the users with the *provenance* of an explanation may help them to better understanding its origin, e.g., an explanation purely based on data, model parameters or both (**O10**). Choosing the right *explanation family* is also important, for example: relation between data features and the prediction, relevant examples such as similar data points, or causal mechanisms (**O1**). Again, interactive explainers have the advantage of giving the user the opportunity to switch between multiple different explanation types. Furthermore, the design of the user interface should be grounded in Interactive Machine Learning, Human–Computer Interaction, User Experience and eXplainable Artificial Intelligence research to seamlessly deliver the explanations. For example, the explainee should be given the opportunity to reverse the effect of any actions that may influence the internal state of the explainer, and the system should always respect preferences and feedback provided by the user (**U4**). Finally, if an explanation of the same event can change over time or is influenced by a random factor, user’s trust is at stake. The explainee should always be informed about the degree of *explanation invariance* and its manifestation in the explainer’s output (**S3**). We discuss this property in more detail in Section 6.3 as it is vital to Glass-Box’s success.

### 6.2.4 Glass-Box Reception and Feedback

We presented Glass-Box to domain experts (general AI background knowledge) and a lay audience with the intention to gauge their reception of our prototype and collect feedback that would help us revise and improve our explainability system. To this end, we opted for informal and unstructured free-form feedback, which was primarily user-driven and, whenever helpful, guided by reference questions based on our list of desiderata. We decided to take this approach given the nature of the events at which we presented our prototype – a scientific conference and a research festival.

Glass-Box is composed of multiple independent components, all of which play a role in the user’s reception of the system:

- natural language understanding and generation;
- speech transcription and synthesis;
- voice and text user interfaces; and
- domain of the modelling task determined by the data set.

Therefore, collecting free-form feedback at this early stage helped us to pinpoint components of the system that required more attention and identify possible avenues for formal testing and design of subsequent user studies.

While presenting the device we only approached members of the audience who expressed interest in interacting with the device and who afterwards were willing to describe their experience. In total, we collected feedback from 6 domain experts and 11 participants of the research festival of varying demographics. When introducing the system and its modes of operation to the participants, we assessed their level of AI and ML expertise by asking background questions, which allowed us to appropriately structure the feedback session.

While discussing the system with the participants, we were mainly interested in their perception of its individual components and suggestions about how these can be improved. Most of the participants enjoyed asking questions and interacting with the device via the voice interface, however some of them found the speech synthesis module that answered their questions “slow”, “unnatural” and “clunky”. These observations have prompted some of the participants to disable voice-based responses and use the text-based chat interface to read the answers instead of listening to them. When asked about the quality of explanations, their comprehensibility and content, many participants were satisfied with received answers. They claimed that personalised explanations provided them with information that they were seeking for, in contrast to the default explanation given at first. However, some of them expressed concerns regarding the deployment of such systems in everyday life and taking the human out of the loop. The most common worry was the impossibility to “argue” with and “convince” the explainer that the decision is incorrect and the explanation does not capture the complexity of one’s case. Some participants were also

sceptical of the general idea of interacting with an AI agent and the fail-safe mode of the device, which produced “I cannot help you with this query.” response whenever the explainer could not answer the user’s question.

## 6.3 Real-life Deployment

Developing Glass-Box and demonstrating it to a diverse audience provided us with a unique experience of building, deploying and refining interactive explainers. To help researchers and engineers who have a similar agenda, we summarise the lessons learnt in Section 6.3.1. We also discuss possible extensions of Glass-Box, focusing on interactive and personalised explainability, in Section 6.3.2 to draw attention to interesting open questions.

### 6.3.1 Lessons Learnt

The major challenge of building Glass-Box was the development overhead associated with setting up the hardware and software needed to make it capable of processing the natural language, and voice interaction. While ready-made components were adapted for these purposes, the effort required to build such a system is still significant and may not always be justified. We encourage the community to build such a device if the research value lies in the system itself, or it is used as a means to an end, for example, investigation of interactive explainability systems. In this case, one should be aware of generalisability issues as each new data set used within such a framework must be adapted by preparing appropriate annotations and (possibly) training a new natural language processing model. In many cases, based on our observations, it seems that all this effort is only justified when the creator of the system is committed to deploying it in the real life. For research purposes, however, the engineering overhead can be overwhelming, in which case we suggest using the *Wizard of Oz* studies [30] as an accessible alternative.

Once Glass-Box was operational, a major barrier to its usability was the time-consuming process of inputting personal data when role-playing the loan application process. At first, we implemented this step as a voice-driven question-answering task, but even with just 13 attributes (most of which were categorical) this proved to be a challenge for the explainee. We overcame this issue by predefining ten individuals whom the explainee could impersonate. We then allowed the explainee to further customise the attributes of the selected individual by asking Glass-Box to edit them (with voice- and text-based commands). In hindsight, we believe that this kind of a task should be completed with a dedicated input form (e.g., a questionnaire delivered as a web page), thereby giving the explainee the full control of the data input stage and mitigating the lengthy “interrogation” process.

The interactive aspect of Glass-Box (discussed in length in Section 6.2.1) provides many advantages from the explainability point of view. For example, it enables the explainee to assess individual fairness of the underlying predictive model and personalise the explanations (see

Sections 6.2.2 and 6.2.3 for more details). However, not all types of explainability algorithms allow for the resulting explanation to be interactively customised and personalised, restricting the set of tools that can be deployed in such a setting. If incorporating the user feedback (captured as part of the interaction, e.g., via argumentation [101]) into the underlying predictive model is desired, this model has to support refinements beyond the training phase, further reducing the number of applicable machine learning and explainability techniques. As noted in Section 6.2.2, some of the interactivity and personalisation desiderata cannot be achieved without “simulating” the explainee’s mental model. While we believe that solving this problem will be a cornerstone of delivering explanations that feel natural to humans, we do not expect it to be solved across the board in the near future.

In case of Glass-Box, where the explanations are presented to the user as counterfactual statements, we observed a tendency amongst the explainees to apply an explanation of a single data point to other, subjectively similar, instances. However tempting, Glass-Box explanations cannot be generalised as they are derived from a predictive model (internals of a decision tree) that neither encodes nor accounts for the causal structure of the underlying phenomenon. This practice may sometimes lead to contradictory explanations, which can be detrimental to the explainee’s trust. Since Glass-Box uses an ante-hoc explainability algorithm (i.e., predictions and explanations are derived from the same ML model), contradictory, incorrect or incoherent explanations are indicative of issues embedded in the underlying predictive model, which should be reported to and addressed by its creators. However, if a post-hoc explainability tool is employed (i.e., explanations are not derived directly from the predictive model, e.g., with surrogate explainers), contradictory explanations manifest an undetermined problem with the system. This issue cannot be uniquely pinpointed and can either be attributed to low fidelity of the explainer or to an underperforming predictive model, in both cases putting the explainee’s trust at risk. Clearly communicating the limitations of such explanations can help to partially mitigate this problem; grounding the explanations in a context (see Section 6.2.2) is another viable approach.

While truthful to the underlying black box, an ante-hoc explainability approach may not be available for a chosen predictive model. For example, deep neural networks are intrinsically complex, which encumbers explaining them without resorting to proxies. This observation highlights the importance of choosing an appropriate predictive model when explainability is a priority or a requirement [55]. Simpler models such as decision trees tend to be less expressive but more interpretable. Complex models such as deep neural networks, on the other hand, are arguably more powerful at the expense of transparency. It is still possible to explain the latter model family with proxies and post-hoc approaches, but issues with the fidelity and truthfulness of these explanations may be unacceptable, e.g., in high-stakes situations such as criminal justice or financial matters [133]. These conclusions have led some researchers [133] to deem low-fidelity, post-hoc explainers as outright harmful. Instead, they argue, developers behind predictive systems for high-stake applications should invest more time in feature engineering and restrict their

toolkit to inherently transparent ML models – see Section 1.1.1 for an in-depth discussion of this dilemma.

As one might expect, the power and flexibility of Glass-Box explanations come at a cost. The interactiveness of the process enables malicious users to ask for explanations of arbitrary data points, which in large quantities may expose internals of the underlying predictive model (**S1** and **S2**). Adversaries can misuse the information leaked by the system in an attempt to reverse-engineer the black box (which may be proprietary) or use such knowledge to game it. This situation is particularly prominent for Glass-Box where the conditional part of the counterfactual explanations is derived from one of the splits in the underlying decision tree, thereby revealing the exact threshold applied to an individual feature, e.g., “had you been older than 25, . . .” implies the  $\text{age} > 25$  internal splitting node. Since every explanation reveals a part of the tree structure (at least one split), with a certain budget of queries the adversary can reconstruct the entire tree.

This issue is intrinsic to ante-hoc explainers but may also affect high-fidelity post-hoc approaches, albeit to a lesser extent since in the latter case the explanations are not generated directly from the black-box model. This undesired side effect can be somewhat controlled by limiting the explanation query budget for untrustworthy users or obfuscating the precise (numerical) thresholds. The latter can be achieved either by injecting random noise (possibly at the expense of explainees’ trust) or replacing the numerical values with *quantitative adjectives*, e.g., “slightly older” (which is also shown to enhance user satisfaction [19]). The trade-off between transparency and security of interactive explainers should be explicitly considered during their design stage, with appropriate mitigation technique implemented and documented.

### 6.3.2 Improving Glass-Box

One of the main contributions of Glass-Box lies in the composition of its software stack and hardware architecture. While investigating the challenge of readying such a system for a deployment is one possible avenue for future research, we believe that a more interesting direction is to design explainability tools and techniques that facilitate (interactive) personalisation of their explanations. Since the latter research aspect is conditioned upon the availability of the former, we suggest using the Wizard of Oz approach [30] to mitigate the need for building an interactive user interface that is responsible for processing the natural language. In this scenario, the input handling and the output generation are done by a human disguised as an intelligent interface, who can access all the components of the tested explainability approach and is only allowed to take predefined actions. Therefore, bypassing an *algorithmic* natural language interface by using the Wizard of Oz approach allows the research agenda to focus exclusively on designing and evaluating the properties of personalised explanations. Such an approach also ensures that the findings are not adversely affected by poor performance of the natural language interface.

To facilitate interactive explainability of an arbitrary black box, the underlying explainer can be based on LIMETree (cf. Chapter 5). It has well-understood properties, high (or full) fidelity

and provides a wide variety of explanations, including contrastive and supportive statements. Furthermore, LIMETree can be applied to three different data domains: tabular, text and image, allowing to test its capacity of interactively generating personalised explanations for a range of diverse tasks. We expect object recognition for images and sentiment analysis for text to be the most fruitful evaluation studies as they do not presuppose any (technical or domain-specific) background knowledge. In particular, the explainees can be asked to interactively personalise two important aspects of these explanations: their content and interpretable representation of the data features.

The objective of the second task depends on the data domain. For text – in contrast to the default bag-of-words interpretable representation – it allows the explainees to introduce bespoke concepts captured by groups of words that are not necessarily adjacent. For images, the users can modify their machine-generated super-pixel segmentation to separate semantically meaningful regions – see the example shown in Figure 6.1 and discussed in Section 6.1. For tabular data, the interpretable representation is constructed by discretising continuous features. Since the local surrogate model is a decision tree, this representation is learnt automatically and cannot be explicitly modified by the explainees. Nonetheless, we could give the users indirect control over the feature splits by allowing them to adjust the tree structure in terms of its maximum depth, the number of data points required for a split and the minimum number of data points residing in a leaf.

Personalising the explanation content, on the other hand, could allow the explainee to choose the explanation type and customise it accordingly. The visualisation of the surrogate tree structure can either depict the whole tree or zoom in on its selected part. The explainee could also inspect tree-based feature importance either by viewing it as a list spanning all of them or by querying the importance of selected features. These two explanation types allow the user to grasp the overall behaviour of the black-box model in the vicinity of the explained data point. For text and images these are the interactions between the words and super-pixels in that region, i.e., within a sentence and an image respectively, and for tabular data the influence of raw features and ranges of their values.

Furthermore, the explainee could get personalised explanations of individual predictions. A counterfactual retrieved from the local tree – e.g., “had these two super-pixels/words not been there, the image/sentence would be classified differently” – can be customised by specifying constraints pertaining to its condition. The explainee could also request a logical rule – e.g., “these three super-pixels/words must be present and these two must be removed to classify this instance as ...” – for any leaf in the tree, which is extracted from the corresponding root-to-leaf path. Both of these explanations allow the user to understand how parts of an image or a sentence (super-pixels and words respectively) come together to predict a data point. Finally, the user could view exemplar explanations of any prediction; these are given by instances drawn from the surrogate model training set (generated by perturbing the explained data point) that are assigned

to the relevant tree leaf. The exemplars will therefore be images with occluded super-pixels, sentences with missing words and, for tabular data, slight variations of the explained data point in its original feature space. We believe that this diverse set of personalised explanations will encourage the user to investigate different aspects of the black-box model, leading to a much better understanding of its behaviour.

Interacting with an explainer that is capable of delivering an array of different explanation types gives rise to another, *investigative* aspect of explainability. An explainee who has learnt which features are important may want to know whether one of the counterfactual explanations is conditioned upon them. Understanding whether the user would discount counterfactuals based on unimportant features and focus on the ones that include important factors instead could provide invaluable insights for designing better explainers. A similar experiment could gauge how the user’s confidence is affected upon discovering that most of the (counterfactual) explanations are conditioned on features labelled as unimportant by a different explanation type. Such meta-explainability studies could additionally uncover limitations of the interaction and personalisation aspects of currently available systems, for example by taking note of the requests that failed according to the user.

Focusing on counterfactuals, the possibility of retrieving multiple explanations of the same length (equal number of conditions) brings up the question of their ordering. One approach is to use a predefined, feature-specific “cost” of including a condition on that attribute into the explanation. This heuristic can be based on the purity (accuracy) of the corresponding counterfactual leaf, the cumulative importance of features that appear on the relevant root-to-leaf path, the collective importance of features listed in its conditional statement or, simply, the number of training data points falling into that leaf [169]. However, a more user-centred approach is to allow the explainee to supply this information either implicitly or explicitly during the interaction.

To improve the quality of explanatory interactions, one may choose to partially replicate the mental model of the explainee using a formal argumentative [35] dialogue introduced by Madumal et al. [101]. Certain statements provided by the user can be parsed into logical requirements, allowing for further personalisation and more convincing explanations. The roles in this dialogue can also be reversed to assess and validate the explainee’s understanding of the black-box model – the machine questioning the human [175, 176]. In this interrogative dialogue, if an insight about the black box voiced by the user is incorrect, the system can provide a personalised explanation in an attempt to correct the relevant beliefs of the explainee. Asking the user “What if?” questions can further assist in this task by directing the explainee’s attention towards evidence relevant to the identified misconceptions. When an interaction is finished, a succinct excerpt summarising the whole explanatory process (similar to a court transcript) can be provided to the user as a reference material. This document should only contain explanations that the user has challenged or investigated in detail, avoiding the ones that agree with the explainee’s beliefs.

The mental model approximation can also be utilised to adjust the granularity and complexity of explanations. For example, a disease can be explained in medical terms – e.g., on a bacterial level – or with easily observable external symptoms – e.g., cough and abnormal body temperature – depending on the audience. While solving this task across the board is currently an open challenge, it may be somewhat possible for individual cases that are highly structured, such as a data set that exhibits a hierarchy of low-level features, which can be hand-crafted and incorporated into the explainer.

## 6.4 Interactive Explainability in the Literature

Throughout our work on Glass-Box we have identified three distinct research strands relevant to interactive explanations and prominent in the literature:

- Interactive Artificial Intelligence and Machine Learning (mostly from the perspective of Human–Computer Interaction);
- interactive explainability tools, which are interactive with respect to the user interface that delivers the explanations; and
- theory of explanatory interactions – for example through a natural language dialogue – between two intelligent agents (be them humans, machines or one of each).

### 6.4.1 Technical Points of View

**HCI Approach** The Human–Computer Interaction community has identified numerous benefits of human input for tools powered by AI and ML algorithms, many of which extend beyond the active learning paradigm where people act as data labelling oracles [8]. For example, consider a movie recommendation system where the user provides both explicit feedback, such as movie ratings, and implicit cues, e.g., movies that the person did not finish watching. In order to utilise the full potential of any feedback and ensure good experience, the users have to understand how their input and actions affect the system (in particular, its underlying predictive model). Among others, the users should be informed whether their feedback is incorporated into recommendations immediately or with a delay, and how “liking” a movie influences future recommendations (e.g., similar genre and shared cast members). Here, this understanding is mostly achieved (in the case of user studies) by inviting people to onboarding sessions or (progressively) disclosing relevant information via the user interface, hence the explanation is provided outside of the autonomous system. These actions help the users to build a correct mental model of the “intelligent agent”, thus allowing them to seamlessly interact with it. Ideally, the users would develop a *structural* mental model that gives them a deep and in-detailed understanding of how the ML or AI operates, however a *functional* mental model (a shallow understanding) often suffices (cf. Section 1.1).

While explanations tend to be provided outside of the system here, several researchers demonstrated how to integrate them directly into the interaction via the underlying user interface [69, 78, 80, 81]. This is especially useful when the system is dynamic – e.g., its underlying predictive model evolves over time – in which case the explanations support and inform users’ interaction with the system and guide them towards achieving the desired objective. There are two prominent examples of such systems in the literature. Kulesza et al. [81] developed an interactive, topic-based naïve Bayes classifier for electronic mail to help the users “debug” and “personalise” email labelling. The users are presented with explanations pertaining to every classified email – words in the email that contribute towards and signal against a given label – and are encouraged to adjust the weights of these factors if they do not agree with their premise, thereby refining and personalising the model in a process which the authors call *explanatory debugging* [78, 80, 81]. Kim et al. [69] designed a similar system where the users can interactively personalise clustering results – which are explained with cluster centroids and prominent exemplars – by promoting and demoting data points within each cluster. In this literature, explanations of predictive models are used to improve users’ understanding (mental model) of an autonomous system to empower them to better utilise its capabilities (e.g., via improved personalisation) by interactively providing beneficial input. Therefore, AI and ML explainability is not the main research objective in this setting and the explanations are not interactive themselves.

**Interactive Explainers** The second relevant research strand that we identified in the literature covers interactive, multi-modal explainability tools in AI and ML. These systems help to investigate a black-box model and its predictions by providing the user with a variety of explanations produced with a range of diverse explainability techniques delivered via (an interactive) user interface. For example, Krause et al. [77] built an interactive system that allows its users to inspect Partial Dependence [44] of selected features (model explanation) and investigate how changing attribute values for an individual data point would affect its classification (prediction explanation) [76, 77]. While combining multiple explainability techniques within a single system with a unified user interface is feasible, ensuring coherence of these diverse explanations poses significant challenges as some of them may be at odds with each other and provide contradictory evidence for the same outcome. Weld and Bansal [178] showed an idealised example of such a system and persuasively argued its benefits, however they have not discussed how to mitigate the issue with contradictory and competing explanations. Despite both of these explainability tools being *interactive*, the interaction itself is limited to the presentation medium of the explanations and a choice of the explainability technique, which, we argue, is insufficient – the system is interactive but the explanations are not. *Truly interactive* explanations allow the user to tweak, tune and personalise them (i.e., their content) via an interaction, hence the explainee is given an opportunity to guide them in a direction that helps to answer selected questions.

**Explanatory Process** The third research strand found in the literature characterises explanatory communication as an interaction between two intelligent agents [10, 101, 138, 176]. Arioua and Croitoru [10] formalised explanatory dialogues in Dung’s argumentation framework [35] and introduced “questioning” dialogues to evaluate success of explanations. Walton [176] introduced a similar *shift model* composed of two distinct dialogue modes: an explanation dialogue and an examination dialogue, where the latter is used to evaluate the success of the former [174–176]. Madumal et al. [101] refined these two approaches and proposed an interactive communication schema that supports explanatory and questioning dialogues, which additionally allow the explainee to formally challenge and argue against some of the (automated) decisions and their explanations. The authors have also empirically evaluated their explanatory dialogue protocol on various text corpora to show its flexibility and applicability to a range of different scenarios. Schneider and Handali [138] approached this problem on a more conceptual level discussing interactions with various explainability tools and showing examples of how they could allow for personalised explanations. Most of the findings published in this body of literature are purely theoretical and have not yet been embraced by practical explainability tools.

#### 6.4.2 Interdisciplinary Perspective

These diverse research paths come together to help explainable AI and interpretable ML researchers and practitioners design appealing and useful explainability tools. While prominent in computer science literature, many of the insights and recommendations discussed in the previous section actually originate from studies of explanatory interactions between humans. This observation has prompted Miller [106] to review a diverse body of research on human explanations in social sciences and propose an agenda for human-centred explainability in artificial intelligence and machine learning. In particular, Miller et al. [107] noticed that explainability systems built for autonomous agents and predictive systems rarely ever consider the end users and their expectations, as they are mostly “built by engineers, for engineers”. Since then, XAI and IML research has taken a more human-centred direction, with many academics and engineers [58, 138, 169, 173, 178] developing and evaluating their approaches against Miller’s guidelines to help mitigate such issues.

**Insights from Social Sciences** Two of Miller’s recommendations are of particular importance: interactive (dialogue-like) nature of explanations and popularity of contrastive explanations among humans. While interactivity of explanations [138] has been investigated from various viewpoints in the literature (and discussed in the previous section), explanations delivered through a bi-directional conversation – giving the explainee the opportunity to customise and personalise them – have not seen much uptake in practice. One-off explanations are still the most popular operationalisation of explainability algorithms [138], where the explainer outputs a one-size-fits-all explanation in an attempt to make the behaviour of a predictive system transparent.

A slight improvement over this scenario is to enable the explainer to account for user preferences when generating the explanations [87, 123], but this modality is not common either. Interactively personalising an explanation allows the users to adjust its complexity to suit their background knowledge, experience and mental capacity; for example, explaining a disease to a medical student should differ from explaining it to a patient. Therefore, an interactive system can satisfy a wide range of explainees' expectations, including objectives other than improving algorithmic transparency itself, e.g., inspecting individual fairness of black-box predictions [83].

The prominence of contrastive statements in human explanations is another important insight from the social sciences, which also highlights their capacity to be interactively customised and personalised. In the recent years this type of explanations has proliferated into the XAI and IML literature in the form of *class-contrastive counterfactual* statements: “Had you earned twice as much, your loan application would have been successful.” This uptake can also be attributed to their legal compliance with the “right to explanation” originally proposed as part of (but ultimately excluded from) the European Union’s General Data Protection Regulation [172, 173]. However, their capacity to be customised and personalised by the explainees themselves is often overlooked in practice [106, 123, 169, 173].

All in all, many of Miller’s insights from the social sciences have found their way into research and real-life applications. An example of the latter is Google’s *People + AI Guidebook*<sup>9</sup> describing best practices for designing human-centred AI and ML products and acknowledging the importance of interaction and explainability in such systems. The lack of customisable explanations has also received attention in the literature [37, 58, 138]. Schneider and Handali [138] have reviewed an array of explainability approaches focusing on the personalisation capabilities of the insights they generate. The authors have observed that bespoke explanations are generally absent in the existing XAI and IML literature. To help researchers design and implement such methods, Schneider and Handali [138] proposed a generic framework for personalised explanations that identifies their three adjustable properties: complexity, content (called “decision information”, i.e., what to explain) and presentation (how to explain, e.g., figures or text). Similarly, Eiband et al. [37] discussed the latter two properties from a perspective of user interface design. Furthermore, Schneider and Handali [138] highlighted that interactive explanation personalisation can either be an *iterative*, e.g., a conversation, or a *one-off* process, e.g., specifying constraints (passed on to the explainer) before the explanation is generated. The latter approach does not, however, require the explainability system to be interactive as the same personalisation can be achieved off-line by extracting the explanation specification from the explainee and subsequently incorporating it into the data or algorithm (when it is initialised). Interaction with explainability systems has also been acknowledged by Henin and Le Métayer [58], who proposed a generic mathematical formulation of black-box explainers consisting of three distinct steps: sampling, generation and *interaction*.

---

<sup>9</sup><https://pair.withgoogle.com>

**Missing in Action** While some explainability approaches introduced in the literature are simply incapable of interactive personalisation – a number of them may still support off-line explanation customisation – others are [123], nonetheless this property is neither utilised nor acknowledged [173]. This lack of recognition may be because the explainability system designers do not see the benefits of this step, or due to the difficulties with building such systems (from the engineering perspective) as well as evaluating them. To facilitate interactive personalisation, the user interface has to be capable of delivering explanations and collecting explainees’ feedback, which may require an interdisciplinary collaboration with User Experience and Human–Computer Interaction researchers. Systematic evaluation and validation of this type of explainers is also more elaborate, possibly requiring multiple rounds of time-consuming user studies.

Despite these hurdles, a number of explainability tools and techniques allow the user to personalise explanations to some extent. Akula et al. [7] presented a dialogue-driven explainability system that uses contrastive explanations based on predictions derived from And-Or graphs and bespoke ontology, however generalising this technique may be challenging as it requires hand-crafting separate ontology and And-Or graph for each individual application. Lakkaraju et al. [87] introduced rule-based explanations that the user can personalise by specifying the features to appear in the explanation – an off-line customisation. Google published their *what-if* tool<sup>10</sup>, which provides the explainee with an interactive interface that supports generating contrastive explanations of selected data points by modifying their features, i.e., asking “What if?” questions. With Glass-Box, we strive to bring together the most important concepts from this wide spectrum of research to enable creation of truly interactive and personalised explanations.

## 6.5 It Is All about the Explainee

This chapter discussed how personalised explanations can improve the transparency of machine learning models and how they can be generated via a human–machine interaction. While other aspects of an explainability system can also be made interactive, we argued that one of the major benefits stems from personalisation. In particular, we showed the difference between interactivity of an explainability system – such as an interactive user interface – and interactiveness of an explanation – e.g., explanation content customisation. To ground our study we have reviewed relevant literature, where we identified three related research strands and showed how our work has the potential to bridge them together. We also supported our discussion and claims with experience gained from building and demonstrating Glass-Box: a class-contrastive counterfactual explainability system that communicates with its users via a natural language dialogue. To the best of our knowledge, it is the first XAI system tested in the wild that supports explanation customisation and personalisation via interaction.

---

<sup>10</sup><https://pair-code.github.io/what-if-tool/>

Our experience with building Glass-Box and experimenting with it helped us to identify a collection of desired functionality and a set of properties that such systems should exhibit (which indirectly benefited other parts of our research, for example, LIMETree and the XAI taxonomy). We discussed their selection applicable to Glass-Box and summarised a list of lessons that we have learnt. Our most important insight draws attention to the engineering overhead required to build such a system despite adapting many off-the-shelf components. We concluded that one should eschew this effort in favour of Wizard of Oz studies when the main objective is to use such a platform as a test bed for various explainability techniques, unless the intention is to deploy it afterwards. Other key observations concerned both the importance and impossibility of simulating the mental model of explainees. While doing so is desirable and highly beneficial, fully satisfying this requirement is out of reach at present. Nonetheless, we observed that by using a formal argumentation framework to model parts of the user-machine interaction, it may be possible to extract relevant fragments of the explainee's background knowledge that can be later utilised to this end. In summary, our findings allowed us to critically evaluate properties of interactive explainers and formulate their desiderata as well as development and deployment guidelines, all of which are a versatile and powerful aid to people building real-life explainers of predictive systems.

## CONCLUSIONS AND FUTURE DIRECTIONS

This thesis explored various explainability and interpretability concepts in artificial intelligence and machine learning. In particular, we investigated the theory behind IML and XAI algorithms and identified their social and technical desiderata, which we formalised as a taxonomy. Then, we examined state-of-the-art surrogate explainers and appraised them based on our list of requirements. We chose this particular explainability approach because of its advertised flexibility and apparent versatility – it is post-hoc, model-agnostic and data-universal. While superficially surrogates may seem like a silver bullet, especially once the issue with their faithfulness is resolved, behind the facade of universality hides a complex process governing their composition and influencing their quality. Therefore, to better understand their inner workings, we decomposed surrogates into independent building blocks, providing us with an opportunity to analyse their fidelity, and design a framework and a meta-algorithm, called bLIMEy, for building tailored surrogate explainers. Insights from bLIMEy uncovered that this seemingly easy-to-use technique conceals crucial complexities away from the end user. Even though composing a bespoke surrogate may be an involved process, we showed that the freedom to build an explainer with desired qualities allows to adjust the type of the resulting explanations, their complexity and delivery mechanism, to name just a few customisation possibilities.

We then tapped into the potential of our bLIMEy framework by proposing LIMETree – a surrogate explainer based on multi-output regression trees that, among others, produces human-centred explanations, accounts for interactions between the predicted classes and comes with fidelity guarantees. To this end, we adapted CtreeX, which is our (ante-hoc) decision tree explainer designed to output widely acclaimed contrastive insights for multi-output regression and multi-class classification trees. Despite all of the progress with surrogate explainers laid out by this thesis, the exceptional effort required to set them up illuminates one question: Why should

we expend so much time and effort into configuring surrogates if instead we can invest these resources in building inherently explainable predictive models? Regardless of the answer – which depends on numerous factors – we cannot overlook the explanation recipients who ultimately judge success of these technologies; simply put, there is much more to artificial intelligence explainability and machine learning interpretability than their purely technical aspects. In particular, our XAI taxonomy and experience with interactively customisable explanations gained through Glass-Box show the importance and benefit of an interdisciplinary approach, spanning multiple fields of computer and social sciences, when building explainability systems.

## 7.1 Explainability, What Is It Good For?

In this thesis, we delved into relatively recent and still dynamic fields of artificial intelligence explainability and machine learning interpretability. We introduced these topics in Chapter 1, which provides a philosophical and theoretical background needed to appreciate the depth and complexity of this research. We reviewed diverse notions of explainability, interpretability, transparency, intelligibility and many others that are often used interchangeably in the literature, and argued in favour of *explainability*. We defined this concept (depicted by Equation 7.1) as (logical) *reasoning* applied to transparent XAI and IML insights interpreted under certain *background knowledge* – a process that engenders *understanding* in explainees. We also inspected the machine learning workflow – which consists of data, models and predictions – and showed how each of these components may be in need of interpretability. In view of a variety of explainability approaches, each one operating in a unique way, we looked at the disputed trade-off between explainability and predictive power, existence of which has only been supported by anecdotal evidence thus far. Our research on surrogate explainers addresses this debate to some extent since one of the main arguments for post-hoc transparency revolves around its flexibility and universality at the expense of fidelity. Such methods are often contrasted with intrinsically explainable predictive models (i.e., ante-hoc interpretability), which provide explanations of superior quality but require extensive engineering effort to be built. However, having seen the complexity of composing bespoke surrogate explainers within the bLIMEy framework, achieving trustworthy post-hoc explainability may require just as much commitment.

$$\text{Explainability} = \underbrace{\text{Reasoning}(\text{Transparency} | \text{Background Knowledge})}_{\text{understanding}} \quad (7.1)$$

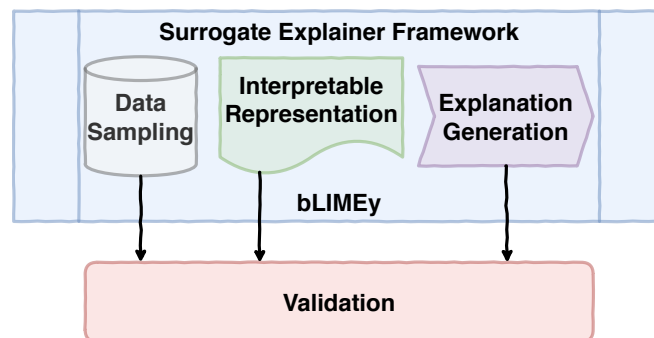
While the most visible aspect of XAI research is the technology that makes it possible, the recipients of such explanations are just as important since their *understanding* of the underlying predictive system determines the ultimate success of an explainer. We explored this topic by looking at human-centred explainability and various desiderata that this concept entails, in particular focusing on explicitly acknowledging presence of humans and projecting the explanations directly at them. To this end, we pursued important insights from the social

sciences that prescribe how to adapt machine explainability to fulfil expectations of the explainees, hence achieve seamless explanatory interaction. The two crucial observations in this space are: a preference for *contrastive* explanations and facilitating a bi-directional explanatory *process* – akin to a conversation – as opposed to delivering a one-off, one-size-fits-all explanation. In addition to enhancing explainee satisfaction, operating within this purview has other, far-reaching benefits such as enabling evaluation of algorithmic fairness, accountability assessment of predictive models and their debugging. We concluded Chapter 1 with a high-level overview of landmark literature in the field of explainable artificial intelligence and interpretable machine learning, which set the scene for the contributions of this thesis.

**Explaining Decision Trees** As part of the introduction, we highlighted two different mental models: *functional* – enough understanding to operationalise a concept; and *structural* – in-depth, theoretical appreciation of underlying processes. We further argued that the former – a shallow form of understanding – aligns with The Chinese Room Argument [139] and the notion of simulatability [96]. We used this observation in Chapter 4 to challenge a popular view that decision trees are interpretable because they are transparent. Deep and/or wide trees are transparent but lack interpretability, which can be restored by applying a suitable form of logical reasoning – a prerequisite of explainability (see Equation 7.1) – undertaken by either an algorithm or a human investigator. We addressed this challenge with CtreeX: a tree-specific explainability algorithm that generates contrastive and supportive statements for individual predictions, the former of which is considered the gold standard of XAI. Since our approach is ante-hoc, it comes with a range of desirable properties achieved at the expense of the method being model-specific.

**Intelligible and Robust Surrogate Explainers** While making trees truly explainable is a contribution in itself, we also showed how to generalise CtreeX to an arbitrary black-box model. To this end, in Chapter 3 we introduced bLIMEy – a principled meta-algorithm and framework for building bespoke surrogate explainers. Surrogates are a powerful and flexible XAI technique that builds an interpretable model to approximate and explain a selected region of a black box: a single prediction, a cohort or an entire predictive model. bLIMEy consists of three modules: interpretable data representation, data sampling and explanation generation – depicted in Figure 7.1 – that are embedded in a structured process instructing how to build effective explainers, and accompanied by a collection of practical recommendations for individual algorithmic components. This theory is complemented by an open source implementation of selected surrogate building blocks within the FAT Forensics Python package, which we introduced in Appendix B.

In particular, we discussed the sensitivity of segmentation-based interpretable representations of images to the occlusion colour and partition granularity, concluding that mean-colour occlusion should be avoided. We also uncovered the ineffectiveness of discretisation&binarisation-based interpretable representation of tabular data, which results in a significant information



**Figure 7.1:** Depiction of the bLIMEy meta-algorithm and framework (Chapter 3) for developing bespoke surrogate explainers. It consists of: interpretable data representation, data sampling and explanation generation steps.

loss, especially when paired with a *linear* surrogate model. We identified label-aware methods as a viable alternative and showed how such partitioning of a feature space can be learnt with a decision tree. For sampling of tabular data, we proposed an explicitly local approach that takes into consideration the black-box predictions of the augmented data, which ensures a diverse sample and identifies the closest decision boundary. In case of images and text, we argued for generating a complete sample instead, which allows to improve fidelity of our surrogate. When fitting a (local) surrogate model to generate an explanation, we demonstrated various benefits of using decision – classification or regression – trees. For example, they do not assume feature linearity and independence, and alleviate the need for a separate interpretable representation when dealing with tabular data. Some of these findings were shown experimentally, whereas others were proven analytically – the main conclusions are presented in Chapter 3 with additional results included in Appendix C.

The mounting evidence of wide-ranging advantages of surrogate decision trees encouraged us to examine this particular explainer configuration in more detail by joining together CtreesX and bLIMEy. In particular, we investigated explainability of black-box image classifiers given their overall popularity and ease of explanation validation by means of visual inspection. To this end, in Chapter 5 we introduced LIMETree: a surrogate explainer based on *multi-output regression trees*, the use of which enables modelling class interactions for probabilistic black boxes. In addition to its inherent ability to model probabilistic multi-class predictors without resorting to the “one-vs-rest” approach, such surrogates have a number of advantages over the ones based on linear models (e.g., LIME), including but not limited to:

- appealing built-in explanation types such as feature importance, decision rules, tree structure visualisation and exemplars, many of which remain meaningful for high-dimensional data, albeit they can become overwhelmingly large;
- native support for categorical attributes;

- out-of-the-box compatibility with unnormalised features – no need to scale the underlying (interpretable) features to the same range;
- inherent capability to model feature interactions;
- desirable modelling assumptions – not imposing linearity of the target but enforcing axis-parallel splits; and
- opportunity to overfit the tree to better represent the black-box decision boundary being approximated, thus enabling high-fidelity explanations.

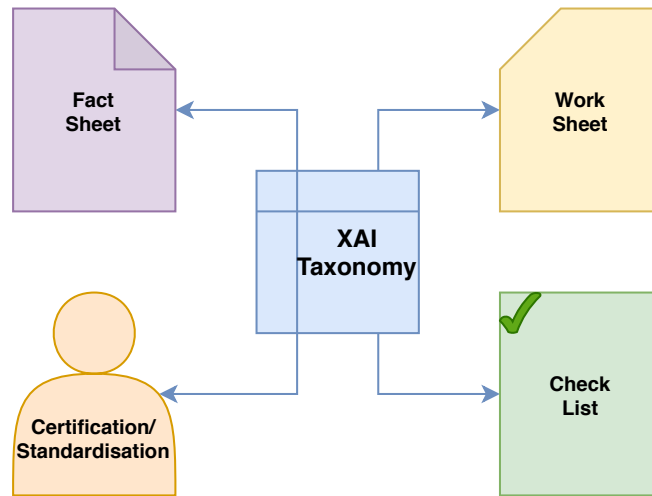
Additionally, we supported LIMETree with a range of theoretical guarantees for achieving full fidelity and showed how to operationalise these concepts in practice. All of our findings were grounded with a pilot user study and a collection of synthetic experiments, showing superiority of our approach over LIME, which is an alternative surrogate explainer.

**Human Aspects** While most of our contributions described so far revolve around technical aspects of AI and ML interpretability, explainees – explanation recipients who tend to be humans – are just as important and ought to be treated as first-class citizens in such systems. For example, the use of CtreeX in a surrogate setting enabled extraction of user-focused and meaningful *contrastive* and supportive explanations, with the former being the cornerstone of human-centred explainability inspired by research in social sciences. The second pillar of this XAI design agenda is making explainability systems interactive and dialogue-like in nature to ensure coherence with people’s expectations regardless of their background knowledge and prior experience with this type of technologies. In Chapter 6, we showed how both of these desiderata can be achieved in practice with Glass-Box: a voice-driven interactive explainer that allows its users to dynamically customise contrastive explanations to answer their unique questions. We also discussed how explainees can engage with explanations on multiple levels to personalise their various aspects such as complexity, content, scope, target and delivery mechanism, among many others.

Given these diverse technical and social requirements, in Chapter 2 we introduced an explainable artificial intelligence taxonomy that covers a broad range of topics spanning five distinct dimensions: functional, operational, usability, safety and validation. Individual properties from within these categories are summarised in Table 7.1. One possible application of the taxonomy, which we adopted throughout this thesis, is a reference guide for systematic construction and evaluation of explainers. Following a well-defined list of properties helped us to direct the development of our XAI approaches – CtreeX and LIMETree – and compare them to LIME, for which we prepared *Explainability Fact Sheets* based on the taxonomy – see Appendix A. In addition to a strong foundation for reporting characteristics of an explainer, our taxonomy can serve as a basis for various other use cases such as work sheets, check lists or a reference for certification and standardisation procedures – see Figure 7.2.

Functional	Operational	Usability	Safety	Validation
<b>F1</b> Problem Supervision Level	<b>O1</b> Explanation Family	<b>U1</b> Soundness	<b>S1</b> Information Leakage	<b>V1</b> User Studies
<b>F2</b> Problem Type	<b>O2</b> Explanatory Medium	<b>U2</b> Completeness	<b>S2</b> Explanation Misuse	<b>V2</b> Synthetic Experiments
<b>F3</b> Explanation Target	<b>O3</b> System Interaction	<b>U3</b> Contextfullness	<b>S3</b> Explanation Invariance	
<b>F4</b> Explanation Breadth/Scope	<b>O4</b> Explanation Domain	<b>U4</b> Interactiveness	<b>S4</b> Explanation Quality	
<b>F5</b> Computational Complexity	<b>O5</b> Data and Model Transparency	<b>U5</b> Actionability		
<b>F6</b> Applicable Model Class	<b>O6</b> Explanation Audience	<b>U6</b> Chronology		
<b>F7</b> Relation to Predictive System	<b>O7</b> Function of Explanation	<b>U7</b> Coherence		
<b>F8</b> Compatible Feature Types	<b>O8</b> Causality vs. Actionability	<b>U8</b> Novelty		
<b>F9</b> Caveats and Assumptions	<b>O9</b> Trust vs. Performance	<b>U9</b> Complexity		
	<b>O10</b> Provenance	<b>U10</b> Personalisation		
		<b>U11</b> Parsimony		

**Table 7.1:** Summary of the explainable artificial intelligence taxonomy introduced in Chapter 2.



**Figure 7.2:** Possible use cases of the explainable artificial intelligence taxonomy (Chapter 2) include: fact sheets, work sheets, check lists and a reference for certification or standardisation procedures.

## 7.2 Towards Human-like Explanations

This thesis investigated a collection of diverse technical and social topics in explainable artificial intelligence and interpretable machine learning. Its centrepiece were modular surrogate explainers and a rigorous engineering process allowing to develop methodologically sound explainability systems that are appealing to both lay and technical audiences. The latter desideratum prompted the creation of an interactive explainer that empowers the users to dynamically personalise contrastive explanations in a human-machine natural language conversation. With such a broad range of findings, there are many possible research directions stemming from our work, some of which more promising than others. One of the incremental improvements could be a generalisation of the CtreeX algorithm to other logical predictive models and their ensembles. Similarly, our XAI taxonomy establishes a strong foundation for a systematic review and classification of existing explainability literature. The research presented in this thesis also uncovers a need and an opportunity to create a collection of clever components of surrogate explainers and a structured bi-directional explanation delivery mechanism.

**Meaningful Interpretable Representations** Popular interpretable representations encode presence and absence of relevant high-level concepts. However, image and tabular data domains require a proxy to achieve this objective: super-pixel occlusion and placing an instance outside of the explained hyper-rectangle respectively. To prevent a surrogate from introducing unintended bias into explanations, we should identify better strategies for removing information from data, e.g., via randomisation and contextual substitution. For tabular data, we proposed an alternative approach where instances were separated and described with logical rules learnt by a decision

tree, thus avoiding the ill-defined procedure of “switching off” an element of the interpretable representation. A similar approach would be highly beneficial for images, where the explainer could perform conceptually meaningful (substitution) operations that go beyond “deleting” parts of an image by occluding them via a predefined colouring strategy.

**Efficient Sampling** Another potential improvement of surrogate explainers concerns the data sampling module. Among others, bLIMEy showed the importance of creating diverse data samples in the selected neighbourhood by considering their predictions provided by the explained black box. However, we did not investigate how to reduce the number of samples while preserving high fidelity of the explainer. Minimising the quantity of augmented data is important as these instances need to be processed by the black box to capture its predictive behaviour (i.e., approximate its decision boundary), which can be expensive in time and/or compute. One particularly promising avenue of research in this direction is active learning. By placing the samples strategically along the black-box decision boundaries within the explained region, one could minimise the number of queries submitted to the explained model and improve the quality of data used to train the (local) surrogate.

**Explanatory Process and Argumentation** The final idea for follow-on research applies to the entire field and not just surrogate explainers. With Glass-Box, we showed a basic form of explainer–explainee interaction that, simply put, was a question-answering system lacking sufficient reasoning capacity on the machine side. Since each explanation reveals just a fragment of the black box and only the right mixture of evidence can paint the full picture, the explainer needs to be responsive and adapt seamlessly to the user’s requests and expectations. Such an engaging algorithmic interlocutor should build logically consistent narratives and serve more as a guide and a teacher than a facts reporter. To this end, we need to develop an explanatory process built on top of a system that enables logical reasoning between intelligent agents: human–machine or machine–machine. The formal argumentation framework can provide such a foundation, managing the dialogue as well as tracking and storing the evolving knowledge base of the involved parties. In the end, nonetheless, the explainee needs to be a savvy interrogator, asking the right questions and firmly navigating the entire process to understand the behaviour of such *super-charged silicone oracles*. After all, in Arthur C. Clarke’s words:

*Any sufficiently advanced technology is indistinguishable from magic.*



## EXPLAINABILITY FACT SHEET EXAMPLES

Explainable AI taxonomy presented in Chapter 2 can be operationalised in a number of ways depending on the intended application. For example, it can form the basis of fact sheets, work sheets, check lists, standards, guidelines or recommendations – see Section 2.4.2 for more details. Nonetheless, the most comprehensive application for preëxisting explainers is an explainability *Fact Sheet* [149], which can serve as a reference guide or a manual for each popular explainer. To aid our future research, we composed such explainability *Fact Sheets* (based on our explainable AI taxonomy) for the three explainers central to our work. We start with Local Interpretable Model-agnostic Explanations (LIME [129]) – Section A.1 on page 196 – which are the most prominent example of surrogate explainers. Their choice is the result of our search for a promising explainer – summarised in Section 2.6 – which led us towards surrogate-based explainability approaches. Next, in Section A.2 starting on page 206, we present an explainability *Fact Sheet* for the Contrastive tree eXplainer (CtreeX) introduced in Chapter 4. This model-specific method is the core building block of our final explainer – LIMETree (Chapter 5) – which is a surrogate based on multi-output regression trees. It is characterised by a *Fact Sheet* included in Section A.3 on page 215. Notably, neither bLIMEy nor Glass-Box received a dedicated *Fact Sheet* since the former is a meta-algorithm and not a specific explainer, whereas the latter is an interactive natural-language voice interface built around CtreeX. We present said *Fact Sheets* on the following pages of this appendix.

# Local Interpretable Model-agnostic Explanations

## Approach Characteristic

### Description

Local Interpretable Model-agnostic Explanations (LIME) is a surrogate explainability method (post-hoc and model-agnostic) that aims to approximate local (in the neighbourhood of a specific data point) behaviour of a black box with a sparse linear model to interpret individual predictions. It was introduced by *Marco Tulio Ribeiro, et al.*, who also published its open source implementation that is capable of explaining *tabular, image* and *text* data.

### Citation

```
@inproceedings{ribeiro2016why,
  title      = {{W}hy Should {I} Trust You?': {E}xplaining the Predictions
               of Any Classifier},
  author     = {Ribeiro, Marco Tulio and Singh, Sameer and Guestrin, Carlos},
  booktitle = {Proceedings of the 22nd ACM SIGKDD
               International Conference on Knowledge Discovery and
               Data Mining, San Francisco, CA, USA, August 13-17, 2016},
  pages      = {1135--1144},
  year      = {2016},
  url       = {https://www.kdd.org/kdd2016/papers/files/rfp0573-ribeiroA.pdf}
}
```

## Variants

### ***bLIMEy***

*build LIME yourself* (bLIMEy) – a modular meta-algorithm for building custom surrogate explainers.

```
@article{sokol2019blimey,
  title      = {b{LIME}y: {S}urrogate Prediction Explanations Beyond {LIME}},
  author     = {Sokol, Kacper and Hepburn, Alexander and Santos-Rodriguez, Raul
               and Flach, Peter},
  journal    = {2019 Workshop on Human-Centric Machine Learning (HCML 2019) at
               the 33rd Conference on Neural Information
               Processing Systems (NeurIPS 2019), Vancouver, Canada},
  year      = {2019},
  url       = {https://arxiv.org/abs/1910.13016},
  note      = {arXiv preprint arXiv:1910.13016}
}
```

### Implementations

Python	
LIME	<a href="https://github.com/marcotcr/lime">https://github.com/marcotcr/lime</a>
FAT Forensics (bLIMEy)	<a href="https://fat-forensics.org/how_to/transparency/tabular-surrogates.html">https://fat-forensics.org/how_to/transparency/tabular-surrogates.html</a>

### Related Approaches

N/A

## Functional Requirements

### F1: Problem Supervision Level

LIME works with:

- **supervised** predictive algorithms; and
- **semi-supervised** predictive algorithms.

### F2: Problem Type

LIME is designed for:

- **probabilistic classifiers**, and supports *binary* and *multi-class* classification tasks; and
- **regression** problems.

### F3: Explanation Target

LIME can only explain **predictions** of a machine learning model.

### F4: Explanation Breadth/Scope

Explanations produced by LIME are **local**.

### F5: Computational Complexity

For every explained data point, the LIME algorithm performs the following *computationally-* and/or *time-*expensive steps, with the cost of each one depending on the actual algorithmic component used:

- **Generating an interpretable data representation** may be necessary for some applications. *Tabular data* may be binned (e.g., using quartile discretisation) to form human-comprehensible features (i.e., concepts) such as  $15 < \text{age} \leq 18$ . *Images* need to be pre-processed to identify super-pixels. Similarly, *text* has to be (possibly, pre-processed and) transformed into the bag-of-words representation.
- In order to train the surrogate model to approximate the local behaviour of a black box, we need to **sample data** around the instance being explained. For *tabular data*, the data augmentation algorithm needs to sample data points with the same number and type of features as the original data set. When an interpretable representation is used, on the other hand, the number of features is still the same, but each one becomes a multinomial feature with its values indicating different bins defined on this feature. For *images* and *text*, the sampling procedure is performed on a binary vector of length equal to the number of unique words (or tokens) for text and super-pixels for images.
- Each sampled data point has to be **predicted** with the underlying black-box model.
- To enforce the locality of an explanation, sampled data are weighted based on their **distance** to the explained instance, which has to be computed for every synthetic data point. While for *text* and *images* this distance is computed on binary vectors, for *tabular data* without an interpretable data representation this procedure is likely to be computationally-heavy, e.g., the Euclidean distance computed on numerical features.
- A **feature selection** algorithm may be run on *tabular data* to introduce sparsity into the explanations.
- For every data point being explained, a local model has to be **trained** for each explained class as the local model's task is to predict one class vs. the rest.

### F6: Applicable Model Class

The LIME algorithm is **model-agnostic**, therefore it works with any predictive model.

The official LIME implementation uses linear **regression** for the local surrogate model, therefore for classification tasks, the explained black box has to be a **probabilistic** model (i.e., output class probabilities).

### F7: Relation to the Predictive System

This approach is **post-hoc**, therefore it can be retrofitted to any (preexisting) predictive system.

### F8: Compatible Feature Types

#### *Tabular Data*

Tabular LIME works with both **categorical** and **numerical** features. If an interpretable data representation is used (default behaviour in the official implementation), all of the features become categorical (bins) to improve legibility of the explanations.

#### *Images*

Images are *always* transformed into an interpretable data representation, namely super-pixels represented as a binary "on/off" vector.

#### *Text*

Text data are *always* transformed into an interpretable data representation, namely a bag of words represented as a binary "on/off" vector.

### F9: Caveats and Assumptions

By default, the LIME implementation discretises tabular data before the sampling procedure, which leads to the sampled data resembling more of a global rather than a local neighbourhood. This is counterbalanced with the data point weighting step based on the proximity of each synthetic data point to the instance being explained. Moreover, discretising first means that in order to predict the sampled data with the underlying black-box model, we need to "un-discretise" them first. In the LIME implementation, this step is performed by uniformly sampling data from each bin, therefore introducing another source of randomness.

For more details, please see "bLIMEy: Surrogate Prediction Explanations Beyond LIME" by *Kacper Sokol, et al.*

## Operational Requirements

### O1: Explanation Family

**Associations between antecedent and consequent.** (Influence of interpretable concepts generated by applying an *interpretable data representation*.)

#### Tabular Data

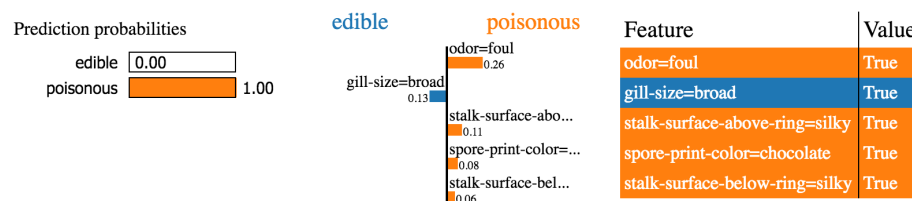
The explanations produced by tabular LIME are **associations between antecedent and consequent** – each feature, or a particular bin on that feature if data are transformed into an interpretable representation, is assigned a positive or negative influence on the local (probabilistic) prediction of a user-selected class.

#### Images and Text

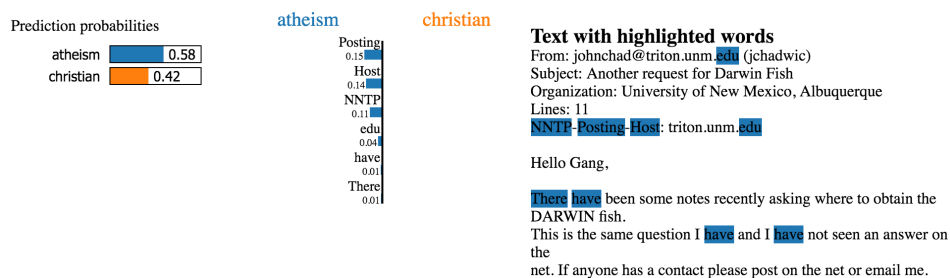
The explanations produced by image and text LIME are **associations between antecedent and consequent** – each word or super-pixel is assigned a positive or negative influence on the local (image-specific or sentence-specific) prediction of a user-selected class.

### O2: Explanatory Medium

LIME explanations are delivered as **visualisations**. For *tabular data*, this is interpretable feature influence, e.g.:



For *text*, this is word influence, e.g.:



Finally, for *images*, this is super-pixel influence, e.g.:



(The figures are taken from the LIME package documentation.)

### O3: System Interaction

LIME explanations are **static** visualisations.

### O4: Explanation Domain

LIME explanations are expressed in terms of *local* interpretable feature influence determined by the coefficients extracted from the locally fitted linear model. For tabular data, the explanation can either be represented in the original data domain (training data features) or in an interpretable domain (quartile feature binning). For images these influence factors correspond to super-pixels and for text these are unique tokens (words) within the explained sentence.

### O5: Data and Model Transparency

For *image* and *text* data there is **no need** for transparency as an interpretable data representation is used. For *tabular data*, on the other hand, regardless of whether an interpretable data representation is used or not, the features **need to** be human-comprehensible.

Since this is a *model-agnostic* interpretability approach, there is **no need** for the underlying predictive model to be inherently transparent in any way.

### O6: Explanation Audience

For *tabular data*, the audience should be familiar with the general domain of the problem to be able to interpret the meaning of the data features. For *images* and *text*, any audience is suitable.

The audience is not required to be familiar with machine learning concepts.

### O7: Function of the Explanation

The main function of LIME is to increase transparency of a prediction output by a black-box model. However, with enough background knowledge, the algorithm can also be used as a diagnostic tool when debugging a black-box predictive system.

### O8: Causality vs. Actionability

LIME explanations are **not** of a causal nature. The explanations also lack a direct actionable interpretation.

### O9: Trust vs. Performance

There is **no** performance penalty since LIME is post-hoc and model-agnostic. Trust in LIME explanations may suffer given instability and randomness of the components making up the explanation generation process (see **S3** for more details).

### O10: Provenance

LIME explanations are based on *interactions* with the black-box model and *synthetic data* sampled around the explained instance, which both affect construction of the local, interpretable, surrogate linear model. The *coefficients* of this model are used as an explanation.

## Usability Requirements

### U1: Soundness

There are two types of local soundness and one type of global soundness that should be measured to evaluate quality of a LIME explanation. First, *mean squared error* (or any other performance metric for numerical values) between the underlying (black-box) model and the local surrogate model (used to generate the explanations) should be evaluated in the **neighbourhood** of the instance being explained to understand soundness of the surrogate model around that instance. Then, *mean squared error* in the neighbourhood of the **closest black-box decision boundary** should be measured to understand how well the surrogate model approximates the black-box decision boundary in that region. Finally, *mean squared error* on the whole data set (e.g., the training data) should be calculated to understand the overall soundness of the surrogate model.

### U2: Completeness

LIME explanations are **not** complete in their nature. For *images* the explanations are image-specific and for *text* the explanations are sentence-specific. For *tabular data*, interpretable feature influence should not be generalised beyond the single data point for which it was generated.

### U3: Contextfulness

**Not applicable.** LIME explanations do not generalise beyond the data point for which they were composed.

### U4: Interactiveness

LIME explanations are **static** visualisations. Interactiveness can only be achieved (reserved for technical users) by modifying the interpretable data representation, e.g., adjusting super-pixel boundaries for images.

### U5: Actionability

LIME explanations can only provide influence of a given factor (determined by the interpretable data representation) on the black-box prediction of a selected data point. They **cannot** however formulate this dependency such as to precisely guide future actions of the explainees.

### U6: Chronology

Chronology is **not** taken into account by LIME explanations.

### U7: Coherence

Coherence is **not** modelled by LIME explanations.

### U8: Novelty

Novelty is **not** considered by LIME explanations.

### U9: Complexity

Complexity of LIME explanations cannot be directly adjusted. It can only be fine-tuned via changes to the interpretable data representation.

### U10: Personalisation

LIME explanations **cannot** be personalised.

### U11: Parsimony

Parsimony for *tabular data* is introduced by the **feature selection** step. Sparsity of *text* and *image* explanations is not necessary as these explanations are overlaid on top of the original image or sentence. For *text*, parsimony can also be achieved by presenting the top  $k$  words in favour and against a given prediction.

## Safety Requirements

### S1: Information Leakage

Since LIME explanations are expressed in terms of the local model coefficients, they do not directly leak any information. However, if the black-box decision boundary approximation is precise enough, the surrogate model can be used to partially reconstruct the black-box model (for example by extracting its local gradient) – for more details please refer to "Model Reconstruction from Model Explanations" by Smitha Milli, et al. Another leakage may occur when creating an interpretable data representation for *tabular data* as some of the discretisation (binning) techniques may reveal characteristics of the data, e.g., quartile binning.

### S2: Explanation Misuse

LIME explanations can be misused by modifying the explained data point according to the feature influence output by the local surrogate model. Nonetheless, this is not an easy task given that the explanation can be expressed in an interpretable data representation. Moreover, this influence is derived for a single data point with a local surrogate model, therefore these insights often do not generalise beyond this individual case. Discovering that the same set of factors is influential for multiple individual explanations (data points) may be taken advantage of, however given that each insight is derived from a unique local surrogate model, this is rather unlikely. Finally, given the flexibility and complexity of surrogate explainers such as LIME, this technique can be used for fairwashing – for more details please refer to "Fairwashing: The Risk of Rationalization" by Ulrich Aïvodji, et al.

### S3: Explanation Invariance

LIME explanations may be **unstable** given that the local models are trained with synthetic data. To ensure consistency, the sampling procedure needs to be controlled either by fixing the random seed or by using a deterministic sampling algorithm. Ideally, the explanations would be *imperceptibly* different regardless of the data sample. This may be true in the limit of the number of sampled data points, however there is no consideration of the minimum quantity of synthetic data required to guarantee the explanation stability.

For *images*, explanation invariance also depends on the stochasticity of the segmenter, which generates super-pixels. For *text*, on the other hand, the interpretable data representation – a bag of words – is deterministic and stable.

Another source of explanation instability for *tabular data* is the *un-discretisation* step applied to sampled data by the LIME implementation. As it stands, the LIME algorithm first discretises the data (to create an interpretable data representation) and then samples within this discretised representation. It means that in order to get predictions of the underlying (black-box) model for each sampled data point, they first have to be un-discretised. (For *images* and *text* this is a well-defined procedure as the binary interpretable representation has 1-to-1 mapping with super-pixels in an image and words in a sentence.) The LIME algorithm does that by sampling each feature value from within the bin boundaries (determined earlier by the discretisation), therefore introducing an additional source of randomness to each explanation. For more details, please refer to "bLIMEy: Surrogate Prediction Explanations Beyond LIME" by Kacper Sokol, et al.

The consistency of LIME explanations has not been studied.

### S4: Explanation Quality

The quality of LIME explanations is **neither** considered with respect to the confidence of predictions given by the underlying model **nor** the distribution of the (black-box) training data.

## Validation Requirements

### V1: User Studies

LIME has been evaluated with three different **user studies**:

1. choosing which of two classifiers generalises better given a LIME explanation;
2. performing feature engineering to improve a model given insights gathered from LIME explanations; and
3. identifying and describing classifier irregularities based on LIME explanations.

Details of these experiments are available in Section 6 of the LIME paper.

### V2: Synthetic Experiments

LIME has been evaluated with three different **simulated** user experiments:

1. validating faithfulness of explanations with respect to the underlying predictive model – the agreement between the top  $k$  most influential features given by an inherently transparent classifier and the top  $k$  features chosen by LIME as its explanation;
2. assessing trust in predictions engendered by LIME explanations – identifying redundant features; and
3. identifying a better model based on LIME explanations.

Details of these experiments are available in Section 5 of the LIME paper.

# Contrastive tree eXplainer

## Approach Characteristic

### Description

Contrastive tree eXplainer (CtreeX) is a model-specific explainability method (designated for classification and regression trees) that generates *class-contrastive* counterfactual explanations of a specific prediction (*local*). Nonetheless, CtreeX explanations generalise to other instances within the same leaf of the tree (*cohort*) without additional processing. Alternatively, CtreeX can generate *class-supportive* counterfactuals, which are explanations based on "differences that make instances similar", i.e., adjustments to feature values that do not yield prediction change. CtreeX is limited to explaining *tabular* data and its explanations are generated by processing the structure of a decision tree – the algorithm requires a direct access to the model's internals. Since it is an *ante-hoc* approach, its explanations are truthful with respect to the tree, i.e., they exhibit full fidelity.

### Citation

```
@phdthesis{sokol2021intelligible,
  title      = {{T}owards Intelligible and Robust Surrogate Explainers:
                {A} Decision Tree Perspective},
  author     = {Sokol, Kacper},
  school     = {School of Computer Science, Electrical and Electronic
                Engineering, and Engineering Maths, University of Bristol},
  year      = {2021}
}
```

### Variants

N/A

### Implementations

Python	
FAT Forensics (CtreeX)	<a href="https://github.com/fat-forensics/fat-forensics">https://github.com/fat-forensics/fat-forensics</a>

### Related Approaches

#### *Interpretable Predictions of Tree-based Ensembles via Actionable Feature Tweaking*

This technique exploits the internals of a tree-based ensemble classifier to offer recommendations for transforming true negative instances into positively predicted ones.

```
@inproceedings{tolomei2017interpretable,
  title      = {{I}nterpretable Predictions of Tree-based
                Ensembles via Actionable Feature Tweaking},
  author     = {Tolomei, Gabriele and Silvestri, Fabrizio}
```

```
and Haines, Andrew and Lalmas, Mounia},
booktitle   = {Proceedings of the 23rd
               ACM SIGKDD International Conference on
               Knowledge Discovery and Data Mining},
pages       = {465--474},
year        = {2017},
organization = {ACM}
}
```

### ***Contrastive Explanations with Local Foil Trees***

This technique utilises locally trained one-versus-rest decision trees (surrogates) to identify the disjoint set of rules that causes the tree to classify data points as the foil and not as the fact.

```
@article{waa2018contrastive,
  title   = {{C}ontrastive Explanations with Local Foil Trees},
  author  = {van der Waa, Jasper and Robeer, Marcel
            and van Diggelen, Jurriaan and Brinkhuis, Matthieu
            and Neerincx, Mark},
  journal = {Workshop on Human Interpretability in Machine Learning
            (WHI 2018) at the 35th International
            Conference on Machine Learning (ICML 2018), Stockholm,
            Sweden},
  year    = {2018},
  url     = {https://arxiv.org/abs/1806.07470},
  note    = {arXiv preprint arXiv:1806.07470}
}
```

## Functional Requirements

### F1: Problem Supervision Level

CtreeX works with:

- **supervised** classification and regression trees; and
- **semi-supervised** classification and regression trees.

### F2: Problem Type

CtreeX is designed for:

- **crisp and probabilistic classification trees**, supporting *binary* and *multi-class* classification tasks; and
- **regression trees**.

### F3: Explanation Target

CtreeX can only explain **predictions** of a decision tree.

### F4: Explanation Breadth/Scope

Explanations produced by CtreeX are **local**, however they generalise to other instances residing in the same tree leaf, therefore they can be considered **cohort** explanations as well.

### F5: Computational Complexity

Before generating individual explanations, CtreeX needs to pre-process the tree structure, which is a one-off procedure that requires visiting each tree leaf – it extracts the conjunction of logical conditions appearing on all root-to-leaf paths. This creates a partially initialised two-dimensional *binary* meta-feature table, with the number of rows corresponding to the leaves count and the number of columns not exceeding the number of unique logical conditions extracted from the tree splitting nodes (some of them can be merged). Then, CtreeX finds a row in the table that corresponds to an explained instance, initialises the missing values in the table and computes a (modified) Hamming distance between its rows to get a ranking of contrastive statements.

### F6: Applicable Model Class

The CtreeX algorithm is **model-specific**, therefore it only works with decision (classification and regression) trees.

### F7: Relation to the Predictive System

This approach is **ante-hoc**, but it can be applied to preëxisting decision trees as long as it is given access to their internal structure.

### F8: Compatible Feature Types

CtreeX only works with **tabular data**, supporting both **categorical** and **numerical** attributes.

## F9: Caveats and Assumptions

Processing the tree structure results in a partially initialised meta-feature representation – see **F5** – since splits of a (binary) tree may result in ambiguous or overlapping encodings. For example, a categorical feature with three values, A, B and C, may be split into  $\{A\}$  and  $\{B, C\}$  (i.e., not A), which compared to another split on the same feature, e.g.,  $\{A, C\}$  and  $\{B\}$ , provides *incomparable* logical conditions (unless additional assumptions are made). When a feature value is in the  $\{B, C\}$  set, which partially overlaps with the  $\{A, C\}$  set, these two logical conditions can either encode the same event (C) or a different event (B), thus introducing ambiguity. A similar phenomenon arises for partially overlapping numerical intervals, e.g.,  $7 < x \leq 10$  and  $8 < x \leq 11$ , which either encode the same or a different event depending on the value of  $x$ . Therefore, a definite explanation cannot be generated without seeing a data point, which fixes the value of each feature, thereby uniquely determining the state of the logical conditions extracted from the tree structure.

# Operational Requirements

## 01: Explanation Family

CtreeX explanations – class-contrastive and class-supportive statements – are based on **contrasts and differences**. They show variants of the explained instance (*examples*) that differ in a subset of feature values, causing respectively a change or preservation of its prediction.

## 02: Explanatory Medium

CtreeX explanations are delivered as **textualisation** highlighting the change in attribute values of the explained instance.

## 03: System Interaction

CtreeX explanations are **static** statements.

## 04: Explanation Domain

CtreeX explanations are expressed in the original data domain, indicating changes of individual feature *values* or *ranges* thereof.

## 05: Data and Model Transparency

The features of *tabular data* used to train the underlying tree **need to** be human-comprehensible. Since CtreeX decouples the size of the explanation from the size of the tree, the model complexity is irrelevant – the tree does **not need** to be intelligible with respect to its structure.

## 06: Explanation Audience

The audience should be familiar with the general domain of the problem to be able to interpret the meaning of the data features. The audience is not required to be familiar with machine learning concepts.

## 07: Function of the Explanation

The primary objective of CtreeX is to increase transparency of decision tree predictions. Additionally, the nature of contrastive and supportive explanations allows them to be used as a gauge of individual fairness (disparate treatment). With enough background knowledge, the algorithm can also be used as a diagnostic tool for debugging decision trees.

## 08: Causality vs. Actionability

CtreeX explanations are **not** of a causal nature, however they are actionable, provided that the features upon which contrastive and supportive statements are conditioned are actionable themselves.

## 09: Trust vs. Performance

There is **no** performance penalty since CtreeX can be applied to preëxisting decision trees without any modifications. CtreeX explanations are also **trustworthy** since the algorithm is *ante-hoc*, i.e., the explanations are derived directly from the predictive model.

## 010: Provenance

CtreeX explanations are based on the *structure* of the explained tree, which is determined by its training procedure and data. The differences in logical conditions extracted from the two root-to-leaf paths corresponding to *fact* and *foil* are the source of feature value contrast shown in the explanation.

## Usability Requirements

### U1: Soundness

CtreeX explanations are **sound** since they are *ante-hoc*.

### U2: Completeness

CtreeX explanations are generated for a single instance (*local*), however they generalise to other data points assigned to the same tree leaf (*cohort*). Their completeness is determined by the logical conditions extracted from the corresponding root-to-leaf paths -- this auxiliary information may be presented along the explanations to place them in a *context* (see **U3**).

### U3: Contextfulness

Context of CtreeX explanations is given by the logical conditions extracted from the corresponding root-to-leaf paths – see **U2** for more details. It is possible to broaden such a context by determining its anti-unification (least general generalisation) based on other leaves with the same prediction (using the binary meta-feature representation).

### U4: Interactiveness

CtreeX explanations are **static** statements, however they can be easily deployed in an interactive (dialogue) system, thus forming the basis of an interactive explainer. Glass-Box is an example of such a system: it uses CtreeX to generate contrastive explanations communicated to the explainee via a voice-driven interactive agent. For more details, please refer to "Glass-Box: Explaining AI Decisions With Counterfactual Statements Through Conversation With a Voice-enabled Virtual Assistant" by *Kacper Sokol, et al.*

### U5: Actionability

Contrastive explanations provided by CtreeX are **inherently actionable**, however the suggested change may not be feasible in the real world, e.g., "had you been 5 years older". Such behaviour can be controlled by annotating human actionability of features and altering the explanation retrieval heuristic to penalise impossible actions appearing in the foil of contrastive statements.

### U6: Chronology

Chronology is **not** taken into account by vanilla CtreeX explanations, however the user can annotate features subjected to the effect of time and customise the explanation retrieval heuristic to take this information into account (see **U5** for more details).

### U7: Coherence

Coherence is **not** modelled by CtreeX explanations, however it can be achieved by deploying the explainer in an interactive system (see **U4**). This allows to collect beliefs of the explainee and encode them as logical facts with respect to the feature values. Next, they can be juxtaposed against logical conditions of the explanations and incorporated into the explanation retrieval heuristic to achieve a *simple form* of coherence with the explainee's mental model.

### U8: Novelty

Novelty is **not** considered by CtreeX explanations, however it can be achieved with a strategy similar to the one outlined in **U7**.

### U9: Complexity

CtreeX retrieves explanations of varying length, measured by the number of logical conditions appearing in the foil of each contrastive statement. By default, these explanations are presented in the order of increasing length, however this can be altered by modifying the explanation retrieval heuristic. For example, certain features may be penalised to control for the contents of the foil – see **U5** for an example. If CtreeX is deployed in an interactive system, the user can provide such restrictions on the fly instead of hard-coding them within the explanation retrieval strategy.

### U10: Personalisation

CtreeX explanations **can** be personalised either by adjusting the explanation retrieval heuristic (by the deployment engineers) or through interaction (by the explainees) – see **U9** for more details.

### U11: Parsimony

Contrastive explanations delivered by CtreeX are *inherently* sparse since they explain a prediction in terms of the *smallest* possible change to its feature values.

## Safety Requirements

### S1: Information Leakage

Since CtreetX is an ante-hoc explainer, it has direct access to the internals of the explained decision tree. All of the logical conditions included in the explanations are extracted from the tree splitting nodes, therefore they reveal parts of the model. This problem can be addressed by obfuscating such thresholds when displaying the explanations. One strategy is to add random noise to the precise numerical thresholds; alternatively, they can be replaced with adjectives quantifying the degree of required change. When explanations are accompanied by a context (**U3**), the risk of leaking information is even greater.

### S2: Explanation Misuse

When adversaries extract enough information from CtreetX explanations (if they were not obfuscated – see **S1**), they can steal or game the underlying decision tree.

### S3: Explanation Invariance

CtreetX explanations are **stable** – their reproducibility is guaranteed for each explained data point given a fixed structure of the underlying decision tree and unchanged explanation retrieval heuristic. Moreover, in many cases "slight" variations to the explained instance will not affect validity of its explanations since they remain accurate as long as the explained data point stays in its original leaf.

### S4: Explanation Quality

The quality of CtreetX explanations can be measured by the number of training instances in the leaf determined by the foil of each contrastive statement. It can be understood as a proxy metric for the density of this particular feature space partition, hence confidence of the tree in this region. For example, when the explained decision tree is overfitted, the stability of CtreetX explanations may decrease since cohorts (instances assigned to a single leaf) become smaller.

### Validation Requirements

CtreeX has neither been evaluated with *user studies* (**V1**) nor with *synthetic experiments* (**V2**). This lack of explicit validation is motivated by the approach being *ante-hoc*, making it completely truthful with respect to the explained tree. Moreover, CtreeX uses contrastive statements, which are a widely acclaimed explanation type that does not require validation per se.

#### V1: User Studies

N/A

#### V2: Synthetic Experiments

N/A

# Tree-based Surrogate Explainer

## Approach Characteristic

### Description

*Tree-based Surrogate Explainer* (LIMEtree) is a post-hoc and model-agnostic explainability method of individual black-box predictions. It approximates *local* (in the neighbourhood of a specific data point) behaviour of a black box using a decision tree. The type of the surrogate tree depends on the black box:

- probabilistic models are approximated with *multi-output regression trees*;
- regressors are modelled with *regression trees*; and
- classifiers are mimicked with *(multi-class) classification trees*.

Among others, the surrogate trees are explained with the *Contrastive tree eXplainer* (CtreeX), which composes appealing counterfactual statements. Nonetheless, explanations that are inherent to decision trees are possible as well: feature importance, tree structure visualisation and logical conditions extracted from root-to-leaf paths, to name a few (see **O1** for more details). LIMEtree is capable of explaining *tabular*, *image* and *text* data – the latter two types require using an *interpretable data representation*.

### Citation

```
@article{sokol2020limetree,
  title   = {{LIME}tree: {I}nteractively Customisable Explanations Based
            on Local Surrogate Multi-output Regression Trees},
  author  = {Sokol, Kacper and Flach, Peter},
  year    = {2020},
  url     = {https://arxiv.org/abs/2005.01427},
  note    = {arXiv preprint arXiv:2005.01427}
}
```

### Variants

N/A

### Implementations

Python	
FAT Forensics (LIMEtree)	<a href="https://github.com/fat-forensics/fat-forensics">https://github.com/fat-forensics/fat-forensics</a>

### Related Approaches

#### **bLIMEy**

LIMEtree is based upon the bLIMEy (*build LIME yourself*) framework – a modular meta-algorithm for building custom surrogate explainers.

## APPENDIX A. EXPLAINABILITY FACT SHEET EXAMPLES

---

```
@article{sokol2019blimey,  
  title   = {b{LIME}y: {S}urrogate Prediction Explanations Beyond {LIME}},  
  author  = {Sokol, Kacper and Hepburn, Alexander and Santos-Rodriguez, Raul  
            and Flach, Peter},  
  journal = {2019 Workshop on Human-Centric Machine Learning (HCML 2019) at  
            the 33rd Conference on Neural Information  
            Processing Systems (NeurIPS 2019), Vancouver, Canada},  
  year    = {2019},  
  url     = {https://arxiv.org/abs/1910.13016},  
  note    = {arXiv preprint arXiv:1910.13016}  
}
```

### LIME

LIMETree is an alternative to LIME (Local Interpretable Model-agnostic Explanations). The major difference between the two is the type of the local surrogate model used to approximate the behaviour of the explained black box. The former is a tree-based surrogate explainer, whereas the later uses sparse linear models instead.

```
@inproceedings{ribeiro2016why,  
  title    = {{W}hy Should {I} Trust You?': {E}xplaining the Predictions  
            of Any Classifier},  
  author   = {Ribeiro, Marco Tulio and Singh, Sameer and Guestrin, Carlos},  
  booktitle = {Proceedings of the 22nd ACM SIGKDD  
            International Conference on Knowledge Discovery and  
            Data Mining, San Francisco, CA, USA, August 13-17, 2016},  
  pages    = {1135--1144},  
  year     = {2016},  
  url      = {https://www.kdd.org/kdd2016/papers/files/rfp0573-ribeiroA.pdf}  
}
```

## Functional Requirements

### F1: Problem Supervision Level

LIMETree works with:

- **supervised** black boxes; and
- **semi-supervised** predictive algorithms.

### F2: Problem Type

LIMETree is designed for:

- **crisp and probabilistic classifiers**, supporting *binary* and *multi-class* classification tasks; and
- **regression** problems.

### F3: Explanation Target

LIMETree can only explain **predictions** of a machine learning model.

### F4: Explanation Breadth/Scope

Explanations produced by LIMETree are **local**.

### F5: Computational Complexity

For every explained data point, the LIMETree algorithm performs the following *computationally-* and/or *time-*expensive steps, with the cost of each one depending on the actual algorithmic component used:

- **Generating an interpretable data representation** is necessary for *images* and *text*. *Images* need to be segmented to identify their super-pixels. Similarly, *text* has to be (possibly, pre-processed and) tokenised. *Tabular data*, on the other hand, do not require an interpretable representation since the surrogate tree partitions the feature space into (computationally) meaningful regions.
- In order to train the surrogate model to approximate the local behaviour of a black box, data need to be **sampled** around the explained instance. For *tabular data*, the augmentation algorithm needs to sample data points with the same number and type of features as the original data set. For *images* and *text*, the sampling procedure is performed on a binary vector of length equal to the number of unique tokens for text and super-pixels for images. In the latter case, a complete binary sample can be generated instead of a random sample depending on the cardinality of the interpretable representation.
- Each sampled data point has to be **predicted** with the underlying black-box model.
- To enforce locality of the explanations, sampled data may be weighted based on their **distance** to the explained instance, which has to be computed for every synthetic data point. While for *text* and *images* this distance is computed on *binary* vectors, for *tabular data* this procedure is likely to be computationally-heavy, e.g., calculating the Euclidean distance on numerical features.
- For every explained data point, an individual local tree has to be **trained**.

### F6: Applicable Model Class

The LIMETree algorithm is **model-agnostic**, therefore it works with any predictive model.

### F7: Relation to the Predictive System

This approach is **post-hoc**, therefore it can be retrofitted to any (preëxisting) predictive system.

## F8: Compatible Feature Types

### *Tabular Data*

LIMEtree works with both **categorical** and **numerical** features.

### *Images*

Images are transformed into an interpretable data representation, e.g., super-pixel segmentation, expressed as binary "on/off" vectors.

### *Text*

Text data are transformed into an interpretable data representation, e.g., tokenisation such as a bag of words, expressed as binary "on/off" vectors.

## F9: Caveats and Assumptions

The choice of an interpretable representation (and its parameterisation) may affect the explanations. For example, segmentation-based interpretable representations of images are sensitive to segmentation granularity and occlusion colour.

If an interpretable representation is employed, generating a complete binary sample instead of a random sample is recommended when the dimensionality of this space is relatively low (up to around 13).

## Operational Requirements

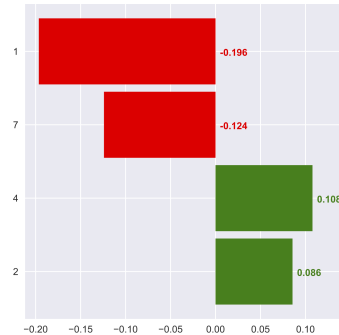
### O1: Explanation Family

(Surrogate) trees support an array of interpretability techniques:

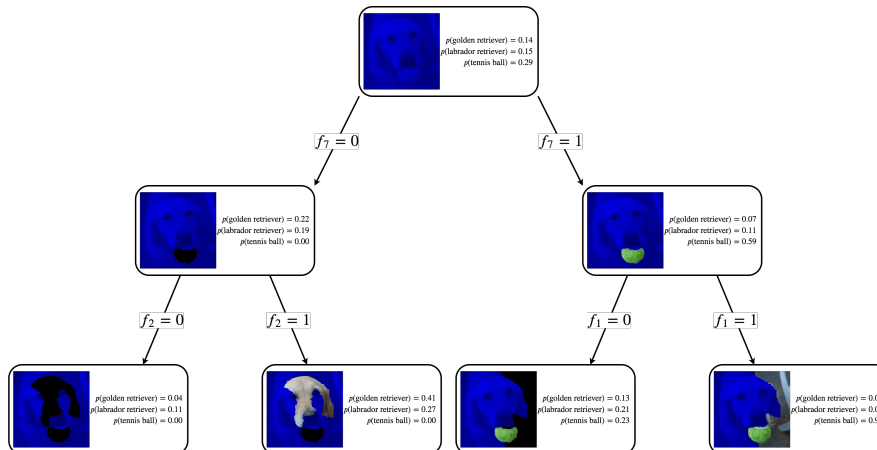
- tree-based feature importance (**associations between antecedent and consequent**);
- visualisation of the tree structure (**associations between antecedent and consequent**);
- logical conditions extracted from a root-to-leaf path (**associations between antecedent and consequent**);
- exemplar explanations taken from training data falling into a single leaf (**contrasts and differences**);
- answers to what-if questions generated based on the tree structure or by querying the model (**contrasts and differences**);
- *class-contrastive* explanations retrieved with CtreeX (**contrasts and differences**); and
- *class-supportive* explanations extracted with CtreeX (**contrasts and differences**).

### O2: Explanatory Medium

Depending on the chosen explanation type (see **O1**), the explanatory medium will vary. *Tree-based feature importance* is usually visualised as a *bar plot*, e.g.:



*Tree structure* tends to be plotted as a *diagram*, which can be adapted to the type of explained data to improve its readability, e.g.:



*Logical conditions* can be *textualised* as a conjunction of literals based on each individual feature for all data types. Exclusively for text and images, these logical conditions can also be applied to an individual data point to enable their *visualisation*. Sentences will appear with missing tokens (e.g., words) and images will see their super-pixels occluded, e.g.:



*Exemplars* are expressed in the domain of the data that are used to train the surrogate tree, however if this is an interpretable data representation, the explanations can be translated into the original data domain as well. Therefore, exemplars of tabular data can only be presented in the corresponding feature space, whereas images and text can either be shown as binary interpretable vectors or transformed into their native data representation. The latter approach would produce variants of the explained image or sentence, which are similar to the examples generated for *logical conditions* – see the figure above.

Finally, *what-if class-contrastive* and *class-supportive* explanations behave similarly to exemplars. However, instead of being derived from the surrogate tree, they are prompted by a search for instances that are similar to the explained data point and either change or preserve its prediction.

### 03: System Interaction

The interactivity of LIMETree explanations depends on the chosen explanation type – see **(O1)**. Tree-based feature importance, visualisation of the tree structure, logical conditions and exemplar explanations are mostly *static* (unless deployed in an interactive user interface). However, what-if class-contrastive and class-supportive explanations are inherently *interactive* – see CtreeX for more details.

### 04: Explanation Domain

For tabular data, the explanations are represented in the original data domain, e.g., importance of features, possible feature value changes, example data points and (logical) conditions applied to a subset of features. For images and text, the explanations are expressed with respect to the components of interpretable representations – their presence or absence – which are super-pixels and tokens respectively.

### 05: Data and Model Transparency

For image and text data there is **no need** for transparency as an interpretable data representation is used. The features of tabular data, on the other hand, **need** to be human-comprehensible.

Since this is a *model-agnostic* interpretability approach, there is **no need** for the underlying predictive model to be inherently transparent in any way.

### 06: Explanation Audience

For tabular data, the audience should be familiar with the general domain of the problem in so far as to interpret the meaning of the data features. For images and text, any audience is suitable.

The audience is not required to be familiar with machine learning concepts.

#### O7: Function of the Explanation

The main function of LIMETree is to increase transparency of a prediction output by a black-box model. However, with enough background knowledge and expertise, the algorithm can also be used as a diagnostic tool for debugging a black-box predictive system.

#### O8: Causality vs. Actionability

LIMETree explanations are **not** of a causal nature. Depending on the explanation type (see **O1**), the insights provided by LIMETree may be actionable, e.g., what-if class-contrastive and class-supportive explanations.

#### O9: Trust vs. Performance

There is **no** performance penalty since LIMETree is post-hoc and model-agnostic. However, these two properties also increase the risk of explanation instability – see **S3** – which may undermine trust of the explainees.

#### O10: Provenance

LIMETree explanations are based on *interactions* with the black-box model and *synthetic data* sampled around the explained instance, both of which affect construction of the local surrogate tree.

## Usability Requirements

### U1: Soundness

There are two types of *local* soundness and one type of *global* soundness that should be measured to evaluate quality of LIMETree explanations. First, predictive performance error, e.g., *mean squared error*, computed between the underlying black box and the local surrogate tree should be evaluated in the **neighbourhood** of the explained instance to understand soundness of the surrogate model around that data point. Then, predictive performance in the neighbourhood of the **closest black-box decision boundary** should be measured to understand how well the surrogate model approximates the black-box decision boundary in that region. Finally, predictive performance on the **whole data set** (e.g., the training data) should be calculated to understand the overall soundness of the surrogate model.

### U2: Completeness

LIMETree explanations are **not** complete per se. For images, the explanations are image-specific; for text, the explanations are sentence-specific. For tabular data, the surrogate tree is only valid for the neighbourhood in which it was trained and its explanations should not be generalised beyond data points from within this subspace.

### U3: Contextfulness

Contextfulness depends on the explanation type (see **O1**):

- tree-based feature importance *lacks* any context;
- visualisation of the tree structure *does not require* a context since the whole surrogate model is presented to the explaine;e;
- logical conditions also *do not require* a context given their completeness;
- exemplar explanations can be considered a generic *source of a context*;
- what-if explanations are phrased within a context, hence they *do not require* one;
- class-contrastive explanations *need to be contextualised* – see the CtreeX Fact Sheet for more details; and
- class-supportive explanations *need to be contextualised* – see the CtreeX Fact Sheet for more details.

### U4: Interactiveness

The degree of interactiveness depends on the explanation type (see **O1**). Tree-based feature importance, visualisation of the tree structure, logical conditions, exemplar and what-if explanations are inherently **static**. Their interactiveness can only be achieved (reserved for technical users) by placing them within an interactive user interface. Class-contrastive and class-supportive explanations, on the other hand, can become interactive – see the CtreeX Fact Sheet for more details.

### U5: Actionability

The degree of actionability depends on the explanation type (see **O1**). Actionability is easiest to achieve with what-if and class-contrastive explanations since they can guide actions of the explaine;e precisely towards the desired outcome.

### U6: Chronology

In general, chronology is **not** taken into account by LIMETree explanations. However, class-contrastive and class-supportive explanations based on CtreeX can consider such information – see the CtreeX Fact Sheet for more details.

### U7: Coherence

In general, coherence is **not** modelled by LIMETree explanations. However, class-contrastive and class-supportive explanations based on CtreeX can take such information into account – see the CtreeX Fact Sheet for more details.

### U8: Novelty

In general, novelty is **not** considered by LIMETree explanations. However, class-contrastive and class-supportive explanations based on CtreeX can take such information into account – see the CtreeX Fact Sheet for more details.

### U9: Complexity

In general, the complexity of LIMETree explanations cannot be controlled directly. However, class-contrastive and class-supportive explanations based on CtreeX can be optimised for simplicity – see the CtreeX Fact Sheet for more details. Additionally, if the surrogate is built on top of an interpretable representation, this data domain can be fine-tuned to suit the explaineer's requirements.

### U10: Personalisation

In general, LIMETree explanations **cannot** be personalised. However, class-contrastive and class-supportive explanations based on CtreeX can be customised via user interaction or modification of the explanation retrieval strategy – see the CtreeX Fact Sheet for more details.

### U11: Parsimony

The parsimony of LIMETree explanations depends upon the explanation type (see **O1**) and explanatory medium (see **O2**).

- Visualisation of the tree structure can be overwhelming when the model is deep or wide.
- Tree-based feature importance, on the other hand, tends to be comprehensible regardless of the model size.
- Exemplar explanations can be overwhelming for tabular data if the number of features is large. However, exemplars of images and text are inherently understandable.
- Similarly, a collection of logical conditions may be too large to understand for tabular data, as well as images and text when they are expressed in a binary interpretable representation. However, in the latter two cases transforming such explanations into the original data domain renders them intelligible – they are overlaid on top of the original image or sentence (see **O2**).
- What-if class-contrastive and class-supportive explanations are inherently sparse.

### Safety Requirements

#### S1: Information Leakage

High-fidelity local surrogate trees may reveal the precise behaviour of a black box in the modelled neighbourhood. See the LIME Fact Sheet for a more detailed discussion.

#### S2: Explanation Misuse

LIMEtree explanations can be misused by modifying the explained data point according to the insights gathered from the local surrogate tree. Some explanation types (see **O1**) reveal more information than others, e.g., visualising the tree structure gives out the entire surrogate model, whereas presenting the explaine with feature importance only conveys a rough behaviour of the explained black box. Moreover, the surrogate model is trained for a single data point (using instances in its neighbourhood), therefore the explanations it produces tend not to generalise beyond this individual case. Since some of the explanations are expressed in an interpretable representation, misusing them may be more difficult.

#### S3: Explanation Invariance

LIMEtree explanations may be **unstable** given that the local models are trained with synthetic data. To ensure consistency, the sampling procedure needs to be controlled either by fixing the random seed or by using a deterministic sampling algorithm. Ideally, the explanations would be *imperceptibly* different regardless of the data sample. This may be true in the limit of the number of sampled data points, however there is no consideration of the minimum quantity of synthetic data required to guarantee stability of explanations.

If a binary interpretable representation is used (images and text), the random sampling procedure may be replaced by generating a complete sample when the size of this representation is relatively small – see **F9** for more details. Doing so can significantly improve explanation stability. For images, explanation invariance also depends on the stochasticity of the segmenter, which generates super-pixels. For text, on the other hand, the interpretable data representation (based on tokenisation) tends to be deterministic and stable.

The consistency of LIMEtree explanations depends on how much the surrogate tree is locally overfitted – see the CtreeX Fact Sheet for a discussion of this phenomenon.

#### S4: Explanation Quality

The quality of LIMEtree explanations is **neither** considered with respect to the confidence of predictions given by the underlying model **nor** distribution of the (black-box) training data. However, the *post-processed* variant of LIMEtree mirrors the precise predictions of the black box to ensure full fidelity of surrogate *model-driven* explanations.

## Validation Requirements

(LIMEtree has only been validated for image data.)

### V1: User Studies

The practical (*human-grounded*) effectiveness of LIMEtree explanations was assessed with a pilot **user study**. In addition to relevant LIME explanations, the participants were shown a surrogate tree trained to explain an image. They were also presented with a brief tutorial outlining how to obtain different kinds of explanations from such a surrogate tree and their purpose (an extended version of **O1**). For each explainability method, the participants were asked about the expected behaviour of the black-box model in relation to objects visible in the image. An example of such a question is "How does the presence of the cat object affect the model's confidence of a presence of the dog object?", with three possible answers: *confidence decreases*, *confidence not affected* and *confidence increases*.

The study showed that LIMEtree helped the participants to answer correctly 25% more questions when compared to equivalent LIME explanations. Details of this experiment are available in Section 6.2 of the LIMEtree paper.

### V2: Synthetic Experiments

LIMEtree has been evaluated with two different (*functionally-grounded*) **proxy** metrics: **surrogate fidelity** and **surrogate complexity**.

1. The faithfulness of the surrogate with respect to the black box, i.e., its ability to mimic it, was measured as an indirect proxy of its trustworthiness, showing superiority of LIMEtree over LIME.
2. The complexity of LIMEtree surrogates, i.e., the tree depth, was evaluated in relation to its fidelity, showing that shallow trees can outperform LIME.

Details of these experiments are available in Section 6.1 of the LIMEtree paper.



## FAT FORENSICS AND REPRODUCIBILITY

Open source software is the backbone of accessible and reproducible research. This realisation is especially significant for artificial intelligence and machine learning, which rely heavily on community-backed toolkits such as scikit-learn, TensorFlow and PyTorch. Notably, these technologies give birth to predictive systems that can take important, and sometimes legally binding, decisions about our everyday life. In most cases, however, these models and their output are neither regulated nor certified. Given the potential harm of data-driven applications, their qualities such as *fairness*, *accountability* and *transparency* are of paramount importance, yet the landscape of publicly available software to operationalise these concepts is scarce and irregular. To ensure high-quality, fair, interpretable and reliable predictive systems, we developed an open source Python package called FAT Forensics. It can inspect important fairness, accountability and transparency aspects of artificial intelligence and machine learning algorithms to automatically and objectively report them back to engineers and users of such technologies. Our toolbox supports a wide range of use cases and can evaluate all elements of a predictive pipeline: data (and their features), models and predictions. Published under the BSD 3-Clause open source licence, FAT Forensics is suitable for personal and commercial applications alike.

### B.1 Origin of FAT Forensics

Reproducibility in AI and ML can be tricky at times since changing the seed of a random number generator may degrade a state-of-the-art predictive system into a subpar model. This phenomenon has long been plaguing both communities, with many prominent researchers promoting publication of high-quality, open source software used for scientific experiments as well as advocating such a requirement as part of the academic peer-review and publishing process [163]. Despite the

encouragement, recommendations and, more recently, requirements of releasing code and data along research papers, AI and ML are still grappling with a widespread reproducibility crisis [62]. While the importance of publicly available implementations is acknowledged almost unanimously, they are commonly treated as a research by-product and often abandoned after publishing a piece of work that benefited from their utility – after all we tend not to revise already published papers.

Numerous implementations of scientific contributions and experiments are released as standalone scripts or packages that do not follow best practices of software engineering. Lack of versioning, testing and documentation can render such code difficult to use for a wider community, thereby hampering its usability and reproducibility in general. Some of these issues can be attributed to the arduous and time-consuming process of designing, developing and maintaining open source software. In other cases, it may simply be due to poor reporting, a desire to protect trade secrets or deliberate obscurity that helps to maintain an edge over competitors while reaping the benefits of free thought embodied by the research community. Regardless of the underlying reasons, the effects are detrimental to the progress of the field. We call this phenomenon *paperware* – code intended to see a paper towards publication rather than implement any particular concept with thorough software engineering practice.

To help mitigate such undesired practices in the field of AI and ML Fairness, Accountability and Transparency (FAT), we developed an open source Python package called FAT Forensics [159, 160], which can serve as a reproducibility vehicle for work published in this area.<sup>1</sup> Its source code is hosted on GitHub<sup>2</sup> and it is accompanied by comprehensive and beginner-friendly documentation<sup>3</sup>, which includes API (Application Programming Interface) reference, usage examples, how-to guides, tutorials and a user guide. The toolbox is capable of analysing all components of a predictive pipeline – data (and their features), models and predictions – by considering their fairness, accountability (robustness, security, safety and privacy) and transparency (interpretability and explainability). It is designed as an interoperable framework for *implementing, testing and deploying* novel algorithms devised by the FAT research community in addition to facilitating their evaluation and comparison against the state-of-the-art solutions. Therefore, the package is flexible enough to support work of researchers and practitioners alike, bearing the promise of democratising access to these techniques. This versatility should encourage creators of FAT algorithms to contribute their approaches to FAT Forensics instead of releasing them as standalone software, given a sturdy foundation and visibility of our toolbox.

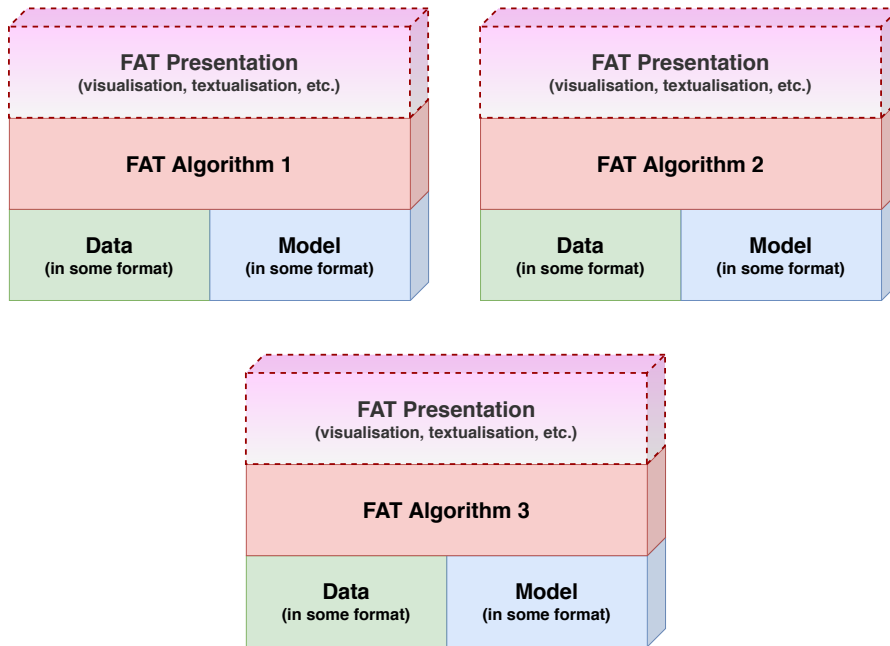
FAT Forensics, in and of itself, is a long-term contribution to the machine learning and artificial intelligence research. In the context of the work presented in this thesis, however, it enables and supports reproducibility of our findings and their easy implementation and operationalisation. Most of our experiments are executed within the scope of the package, building

---

<sup>1</sup>This project was originally funded by Thales, and has started as the result of a collaborative research agreement between Thales and the University of Bristol.

<sup>2</sup><https://github.com/fat-forensics/fat-forensics>

<sup>3</sup><https://fat-forensics.org>



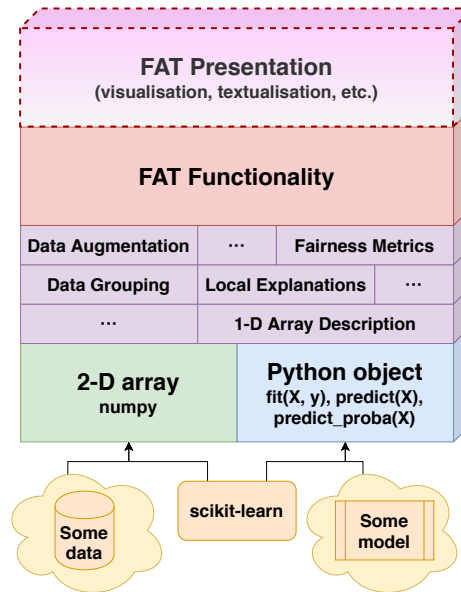
**Figure B.1:** Typical architecture exhibited by academic software landscape – standalone code-bases, distributed with unnecessary dependencies, offering incompatible and non-standard APIs.

upon its low-level API intended for ML researchers, e.g., a Jupyter Notebook<sup>4</sup> demonstrating our initial investigation of bLIMEy [158]. While some studies require custom code that is not distributed with the software, these snippets are specific to processing the selected data and models as well as visualising the results, therefore their absence does not undermine the overall reproducibility. In particular, Fat Forensics is an integral part of the bLIMEy framework and meta-algorithm (Chapter 3), whose inherent algorithmic modularity takes full advantage of the multi-layered architecture of the package. Nonetheless, others parts of our research – CtreeX (Chapter 4), LIMETree (Chapter 5) and Glass-Box (Chapter 6) – also benefit from its versatility.

## B.2 Toolbox Overview

When FAT software is developed exclusively in support of research outputs, it is often distributed as a monolithic code-base accessible via an atypical user-facing API. Given the primary objective of such programmes, they also tend to be burdened with specific data sets, predictive models and (interactive) visualisations, all of which are determined by the envisaged use case, e.g., experiments required for publishing a paper – see Figure B.1. To mitigate these issues, FAT Forensics decouples the core functionality of FAT algorithms from their possible applications and presentation artefacts (e.g., visualisation). This is achieved with a modular design that allows to share and reuse a collection of low-level building blocks, thereby freeing the package from

<sup>4</sup>[https://github.com/So-Cool/bLIMEy/blob/master/HCML\\_2019/bLIMEy.ipynb](https://github.com/So-Cool/bLIMEy/blob/master/HCML_2019/bLIMEy.ipynb)



**Figure B.2:** Modular architecture of FAT Forensics. The input requirements for data sets and predictive models are kept to a minimum and are very flexible: 2-dimensional NumPy arrays and Python objects with `fit`, `predict` and, optionally, `predict_proba` methods respectively. The FAT functionality is composed with atomic building blocks, making the process of constructing new tools, or creating variants of existing algorithms, as easy as connecting the right components.

unnecessary dependencies and creating a comprehensive and appealing API. This architecture makes the process of constructing new FAT tools, or creating variants of existing ones, as easy as connecting the right components, therefore supporting a range of diverse use cases – see Figure B.2. Since visualisations constitute a vital part of some FAT algorithms, FAT Forensics provides a basic plotting module, however its functionality is conditioned on an *optional* Matplotlib [61] software dependency, making the package suitable for plain numerical analysis fed into custom presentation interfaces such as dashboards and reporting tools.

The software architecture shown in Figure B.2 allows FAT Forensics to make minimal assumptions about the operational setting of its key FAT implementations – a generalisation achieved with a shared interface layer. Within this infrastructure, the format requirements for data sets and predictive models are kept to a minimum, lowering any barriers for adoption of the package in new and preexisting projects. In this abstraction a data set is assumed to be a 2-dimensional NumPy [113] array. Both classic and structured arrays are supported – the latter is a welcome addition given that some of the features may be categorical (string-based). A predictive model, on the other hand, is assumed to be a plain Python object that has `fit`, `predict` and, optionally, `predict_proba` methods. This flexibility ensures that our package works with scikit-learn [120] – the most popular Python machine learning toolbox – without introducing additional dependencies. Moreover, our approach makes FAT Forensics compatible with other AI and ML software packages – e.g., TensorFlow [1], PyTorch [116] and custom models, even

ones hosted on the Internet and accessible via a web API – since their predictive functions can be easily wrapped inside a Python object with all the required methods. In addition to the relaxed input formats, all of the techniques incorporated into the package are decomposed into atomic components that can later be reused to create new functionality.

FAT Forensics improves over existing solutions as it collates algorithms from across the FAT domains, taking advantage of their shared functional building blocks. This interoperability allows, for example, a counterfactual data point generator to be used as a post-hoc explainer of black-box predictions on the one hand, and as an individual fairness (disparate treatment) inspection tool on the other. The modular architecture enables the package to deliver robust and tested low-level FAT building blocks as well as a collection of user-ready FAT techniques built on top of them. Users can choose from these ready-made tools or, alternatively, combine the available building blocks to create their own bespoke algorithms without the need of modifying the code-base. We discuss this diversity and re-usability – often happening across the FAT borders – individually for fairness, accountability and transparency later in this section.

Furthermore, the common interface layer of the toolbox enables two distinct *modes of operation*. In the **research mode** (data in – visualisation out), the tool can be loaded into an interactive Python session, e.g., a Jupyter Notebook<sup>5</sup>, supporting rapid prototyping and exploratory analysis. This mode is intended for FAT researchers who can use it to propose new fairness metrics, compare them with the existing ones or use them to inspect a new predictive system or data set. The **deployment mode** (data in – data out), on the other hand, can be used to incorporate FAT functionality into a data processing pipeline to enable (numerical) analytics, providing a foundation of automated reporting and dashboarding. This mode is intended for machine learning engineers and data scientists who may use it to monitor or evaluate a predictive pipeline during its development and deployment.

FAT Forensics is published under the BSD 3-Clause open source licence, which permits personal and commercial applications. To ensure its longevity, sustainability, extensibility and streamlined maintenance, the package employs software engineering best practice such as unit testing, continuous integration, well-defined module structure and consistent code formatting. Moreover, the development workflow established around the toolbox provides an easy way for the community to report bugs, submit patches and contribute novel FAT algorithms. Notably, FAT Forensics is accompanied by a thorough and beginner-friendly documentation that is carefully crafted to cater a wide range of users and applications. It is based on four pillars, which together build up the user’s confidence in working with the package, and consists of:

- narrative-driven **tutorials** designed for new users – they provide step-by-step guidance through practical use cases of all the main aspects of the toolbox;
- **how-to guides** created for relatively new users of the package – they showcase the flex-

---

<sup>5</sup><https://jupyter.org>

	<b>Fairness</b>	<b>Accountability</b>	<b>Transparency</b>
<b>Data/ Features</b>	• Systemic bias • Sample size disparity	• Sampling bias • Data density checker	• Data description
<b>Models</b>	• Group-based fairness	• Systematic performance bias	• Global surrogates (bLIMEy) • Individual Conditional Expectation • Partial Dependence
<b>Predictions</b>	• Counterfactual fairness	• Prediction confidence	• Counterfactuals • Local surrogates (bLIMEy and LIME)

**Table B.1:** Fairness, accountability and transparency functionality implemented in the latest release (version 0.1.0) of FAT Forensics<sup>6</sup>.

ibility of the software and explain how to use it to solve common FAT challenges such as building a custom surrogate explainer with bLIMEy [158];

- **API documentation** describing functional aspects of the algorithms implemented in the toolbox and intended for a technical audience as a reference material – it is complemented by task-focused minimal *code examples* that put the objects, methods and functions in context; and
- **user guide** discussing theoretical aspects of the algorithms implemented in the package such as their properties, restrictions, caveats, computational time and memory complexity, among others.

We envisage that the versatility and flexibility of FAT Forensics will encourage the FAT community to contribute their algorithms to the package. We offer our toolbox as an attractive alternative to releasing standalone implementations, thus reaching a wider audience and keeping the package itself at the frontiers of algorithmic fairness, accountability and transparency research. At the time of writing, FAT Forensics implements a collection of basic methods summarised in Table B.1 – the current version of the software is 0.1.0, which reflects its user-readiness but also highlights the early development stage. This list includes only end-to-end tools, with a plethora of their core building blocks available to more savvy users who can create alternative, bespoke FAT algorithms. To show practicalities of how this modular design influences the functionality of the toolbox, we discuss below examples of the interoperability exhibited by these components separately for fairness, accountability and transparency.

**Fairness** All of the:

- sample-size disparity;
- sub-population fairness [57] such as group-unaware, equal opportunity, equal accuracy and demographic parity;

<sup>6</sup>[https://fat-forensics.org/getting\\_started/structure.html#structure-of-the-package](https://fat-forensics.org/getting_started/structure.html#structure-of-the-package)

- sub-population predictive performance disparity; and
- summary statistics

can be built upon a function that partitions data with respect to a chosen (sensitive) attribute. This grouping is implemented as one of the core components of the package and can be coupled with any standard predictive performance metric to evaluate group-based fairness. With the addition of a module that fits a separate threshold for probabilistic predictions of individuals belonging to different groups, a variety of fairness criteria – not limited to those implemented in the package itself – can be derived. To this end, the user simply needs to provide a function that measures some sort of performance exclusively based on true labels and predictions. Moreover, the grouping function enables evaluating quality of predictions separately for each user-defined sub-population including group-specific predictive performance, and analysis of the number of data points and their feature distribution across different (possibly underrepresented) groups, all of which can help to debug a black-box model. For example, if a data set contains a limited number of samples for some sub-population, this group will most likely face bigger predictive errors, which in turn may have fairness implications down the line.

**Accountability** Estimating density of a data set in a selected feature subspace can provide important insights about the data themselves and any model trained with them. By doing so in the neighbourhood of an individual instance, we can gather important clues about the robustness of its prediction, e.g., a density score can be treated as a proxy measure of predictive models’ confidence [121]. In addition to engendering trust in predictions, a density estimate can help to compute and validate *realistic* counterfactual explanations [123]. Such a strategy can discount counterfactuals that reside in low-density regions (with respect to the distribution of the training data) since these instances tend to be impossible to achieve in the real life. Suggesting that a person ought to be 200 years old to receive the desired outcome or presenting a case of a male who gave birth to three children are examples of unrealistic and undesired explanations that are expected to lie in sparse areas.

**Transparency** A black-box counterfactual explainer can be used to generate explicit (of a selected class) or implicit (of any class) contrastive explanations, i.e., hypothetical what-if scenarios. By restricting the set of features upon which a counterfactual statement is conditioned, e.g., to only include protected attributes, such an explanation can gauge individual fairness via *disparate treatment*. An alternative application of a contrastive explainer is identifying possible feature variations of a given data point that do not affect its prediction, i.e., explanations based on supportive instances disguised as “counterfactuals of the same class”. Surrogate models built within the bLIMEy framework [158] also exhibit a high level of modularity, albeit their building blocks have limited use outside of their primary function. Nonetheless, a selection of

algorithms enabling customisation of their individual components – interpretable representation, data augmentation and explanation generation – is at the user’s disposal<sup>7</sup>.

### B.3 Related Software

A recent attempt to create a comprehensive framework for FAT algorithms is the *What-If* tool<sup>8</sup>, which implements various fairness and explainability approaches. However, combining more than one domain in such a toolbox is rather uncommon. Instead, packages tend to focus on either fairness, accountability or transparency, collecting the most relevant state-of-the-art implementations from within a chosen realm. Among them, algorithmic *fairness* frameworks written in Python are relatively ubiquitous, for example:

- Microsoft’s Fairlearn<sup>9</sup> [5];
- IBM’s AI Fairness 360<sup>10</sup> [14];
- FairTest<sup>11</sup> [168];
- BlackBoxAuditing<sup>12</sup> [4, 39];
- FairML<sup>13</sup>; and
- fairness-comparison<sup>14</sup> [43].

*Transparency* libraries collating multiple explainability algorithms as well as inherently interpretable predictive models are also gaining in popularity. These include:

- Microsoft’s InterpretML<sup>15</sup> [111];
- IBM’s AI Explainability 360<sup>16</sup>;
- Oracle’s Skater<sup>17</sup> [75];
- ELI5<sup>18</sup>; and
- Yellowbrick<sup>19</sup> [16, 17].

Packages implementing individual algorithms – often released by the researchers who published a given method – are also common, for example:

- LIME<sup>20</sup> for Local Interpretable Model-agnostic Explanations [129];

---

<sup>7</sup>See the bLIMEy how-to guide: [https://fat-forensics.org/how\\_to/transparency/tabular-surrogates.html](https://fat-forensics.org/how_to/transparency/tabular-surrogates.html).

<sup>8</sup><https://pair-code.github.io/what-if-tool>

<sup>9</sup><https://github.com/fairlearn/fairlearn>

<sup>10</sup><https://github.com/IBM/AIF360>

<sup>11</sup><https://github.com/columbia/fairtest>

<sup>12</sup><https://github.com/algofairness/BlackBoxAuditing>

<sup>13</sup><https://github.com/adebayoj/fairml>

<sup>14</sup><https://github.com/algofairness/fairness-comparison>

<sup>15</sup><https://github.com/interpretml/interpret>

<sup>16</sup><https://github.com/IBM/AIX360>

<sup>17</sup><https://github.com/oracle/Skater>

<sup>18</sup><https://github.com/TeamHG-Memex/eli5>

<sup>19</sup><https://github.com/DistrictDataLabs/yellowbrick>

<sup>20</sup><https://github.com/marcotcr/lime>

- SHAP<sup>21</sup> for SHapley Additive exPlanations [100];
- anchor<sup>22</sup> for local high-precision rules [130]; and
- PyCEbox<sup>23</sup> for visualising Partial Dependence [44] and Individual Conditional Expectation [48].

*Accountability* software, on the other hand, is relatively under-explored. The most prominent implementations in this space deal with robustness of predictive systems against adversarial attacks, for example:

- FoolBox<sup>24</sup>;
- CleverHans<sup>25</sup>; and
- IBM’s Adversarial Robustness Toolbox<sup>26</sup>.

Even more scarce are open source packages for AI and ML *accountability* (security and privacy). The most visible software solutions in this space are:

- TensorFlow Privacy<sup>27</sup>;
- OpenMined’s PyGrid<sup>28</sup>; and
- DeepGame<sup>29</sup>(neural network verification).

With its modular design, FAT Forensics aims to bring together selected functionality from across algorithmic fairness, accountability and transparency domains. The development of the toolbox adheres to best practices of (open source) software engineering and the package is accompanied by a comprehensive documentation, both of which make it stand out amongst its peers. Our implementation abstracts away from a fixed input format for data and predictive models, creating a versatile and appealing API. Furthermore, the philosophy behind the design and development of our toolbox enables it to support two distinct modes of operation – research and deployment – thus catering to a diverse audience and supporting a range of tasks such as implementing, testing and deploying FAT solutions. All of these principles sitting at the core of FAT Forensics give it the capacity (as a piece of software) to improve reproducibility of fairness, accountability and transparency research in AI and ML.

---

<sup>21</sup><https://github.com/slundberg/shap>

<sup>22</sup><https://github.com/marcotcr/anchor>

<sup>23</sup><https://github.com/AustinRochford/PyCEbox>

<sup>24</sup><https://github.com/bethgelab/foolbox>

<sup>25</sup><https://github.com/tensorflow/cleverhans>

<sup>26</sup><https://github.com/IBM/adversarial-robustness-toolbox>

<sup>27</sup><https://github.com/tensorflow/privacy>

<sup>28</sup><https://github.com/OpenMined/PyGrid>

<sup>29</sup><https://github.com/TrustAI/DeepGame>





## ANALYSIS OF bLIMEy COMPONENTS AND THEIR INTERACTIONS

Each building block of bLIMEy – interpretable representation, data sampling and explanation generation – can be operationalised with a range of diverse algorithms, which have their own assumptions, caveats and requirements. Ensuring that their properties are well-understood and adequate for the use case at hand is the key to composing powerful and effective bespoke surrogate explainers. In this appendix we explore a collection of relevant modules that are popular in the literature, reporting on their strengths and weaknesses, in each case suggesting best practices and suitable alternatives. We inspect them individually as well as in conjunction with their accompanying components, providing recommendations in both settings. In particular, we tackle occlusion-based interpretable representations of images in Section C.1, investigating the significance of segmentation granularity and occlusion colour. We show that mean-colour occlusion should be avoided in favour of single-colour occlusion, especially when the super-pixels are small, with the findings grounded in experimental results.

Next, in Section C.2, we analyse the importance of discretisation quality when building interpretable representations of tabular data. We find that feature space segmentation generated with decision trees (which are aware of the target variable) results in improved purity of hyper-rectangles when compared to a discretisation that relies solely on the distribution of data – we show this relation experimentally on four distinct data sets. Finally, we investigate the provenance of LIME-like surrogate explanations for tabular data, where the binary interpretable representation is built upon discretisation and modelled with a linear predictor, thus outputting influence of presence and absence of human-intelligible concepts. To this end, in Section C.3, we derive a mathematical formulation of such explanations for ordinary least squares and analyse significance of surrogate training data distribution and their black-box predictions. We find that the hyper-rectangle allocation of this data sample has a large effect on the influence scores,

whereas the quality of the discretisation is somewhat unimportant and splitting numerical features into more than three bins provides no added benefit.

## C.1 Occlusion-based Interpretable Representations of Images

Occlusion-based Interpretable Representations (IRs) of images are parameterised by the segmentation *granularity* and the *colouring* strategy, which is used for “removing” the content of super-pixels to hide them from a black box. The exact relation between these two properties and the resulting explanations is discussed in Section 3.3.2, with the main conclusions summarised in this appendix. We support our findings with a collection of experimental results presented in Figure C.1, the technical setup of which is outlined in the following paragraph. Notably, while image IRs are influenced by both segmentation granularity and occlusion strategy (i.e., an information removal proxy), text data do not require the latter given that language models usually allow flexible input size – see Section 3.2.1 for more details. Studying the effect of text pre-processing and tokenisation (which correspond to image partitioning) on the quality of relevant interpretable representations, however, is a challenging task that may not provide comprehensive insights given the breadth and scope of available techniques.

**Experiment Setup** For all of the experiments, our black box was the pre-trained *Inception v3* neural network distributed with PyTorch [116]. We sampled 100 (square and no smaller than 256×256 pixels) test images at random from the ImageNet [33] validation set, which were next resized to 256×256 pixels. We segmented these images with SLIC [2] – k-means clustering in the RGB (Red, Green Blue) colour space – using the implementation provided by scikit-image (`skimage.segmentation.slic`) [170]. We executed our experiments using the bLIMEy algorithmic framework [158], which modularises surrogate explainers into: *interpretable representation*, data sampling and explanation generation, and implements these building blocks within the FAT Forensics Python package [159, 160].

Segment occlusion was done with the following selection of colours described in the RGB space:

black (0, 0, 0);  
white (255, 255, 255);  
red (255, 0, 0);  
green (0, 255, 0);  
blue (0, 0, 255);  
pink (255, 192, 203);

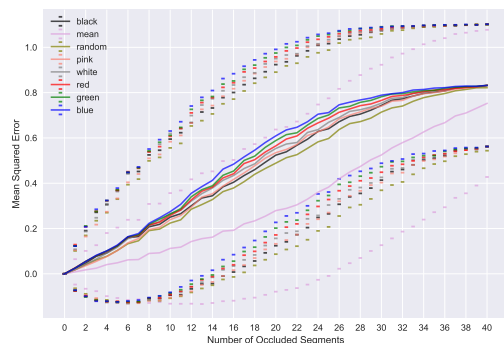
**mean** each super-pixel is replaced with a solid patch of the mean colour computed for the pixels residing within this segment; and

**random** a separate random colour, sampled uniformly from the RGB space, is used to occlude each individual super-pixel.

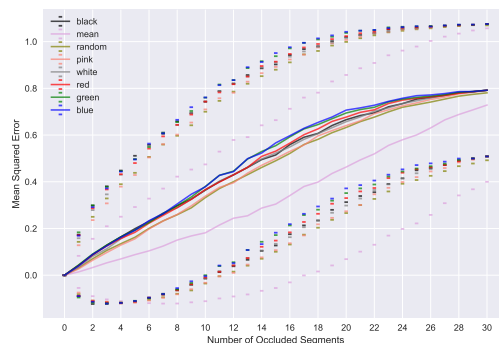
We partitioned the test images into 5, 10, 15, 20, 30 and 40 regions to capture the influence of the segmentation granularity on the IR – these tiers are visualised in separate panels of Figure C.1. For a fixed number of segments, we iterated over the quantity of occluded super-pixels from 0 to all of the partitions (x-axes in Figure C.1), randomising the occlusion pattern 20 times at each step. We applied this procedure to all of our test images, separately for every colouring strategy. Finally, we measured the influence of each occlusion strategy and segmentation granularity by calculating the Mean Squared Error (MSE) between the probability of the top class predicted for an original image and the prediction of the same class when the image was (partially) occluded (y-axes in Figure C.1).

**Occlusion Colour** Figure C.1 provides clear evidence that the *mean*-colour occlusion strategy behaves unlike any other approach, including the *random* method. The lower MSE for this particular technique indicates that it is not as effective in “removing” class-identifying information from images as any other occlusion strategy that we tested. Intuitively, the reason for this behaviour is the “blurring” effect described in the *Information Removal and Loss* part of Section 3.3.2 and exemplified in Figure 3.3. This phenomenon becomes especially pronounced when images are segmented into small super-pixels, as having more of them for a fixed image size makes each partition more uniform with respect to the colour of its individual pixels – the increasing separation of the *mean* strategy MSE line when moving from 5 (Panel C.1f) to 40 segments (Panel C.1a).

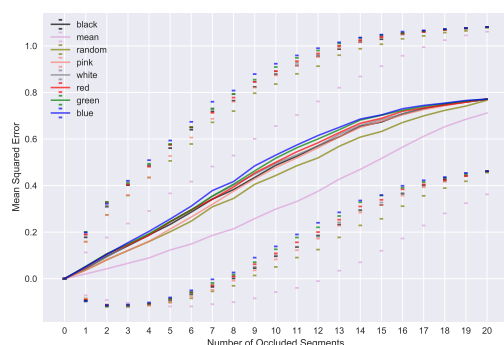
**Segmentation Granularity** By inspecting each panel of Figure C.1, we can see that the granularity of segmentation directly affects the *mean*-colour occlusion strategy – the aforementioned separation between the MSE line of the *mean* approach and every other line. The behaviour of all the fixed-colour approaches, on the other hand, is very similar for any number of segments regardless of the exact occlusion colour – these MSE lines are clustered together in Figure C.1. Notably, this observation also applies to the *random* strategy, which can be very volatile since it uses a different occlusion colour for each individual super-pixel. Both of these insights are clear evidence that using the *mean* colouring should be avoided in occlusion-based interpretable representations of images. Figure C.1 substantiates our observation that this occlusion strategy becomes less effective as the number of super-pixels increases since relatively small segments tend to have a uniform colour distribution because of the pixel “continuity”, i.e., high correlation of neighbouring pixels, making them visually similar to their respective *mean*-coloured patches. Additionally, this phenomenon may affect images that have an out-of-focus background, e.g., portraits, since their



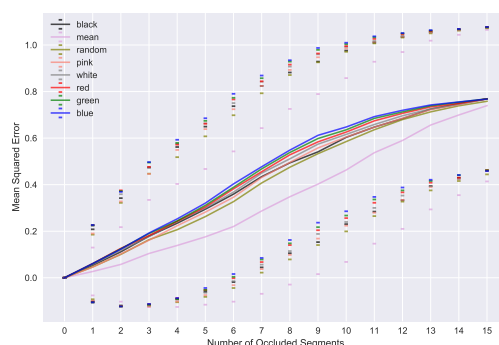
(a) Mean squared error for a 40-segment partition.



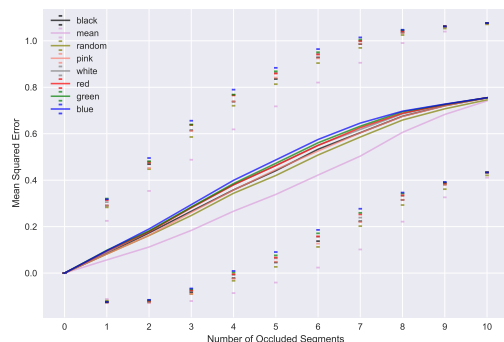
(b) Mean squared error for a 30-segment partition.



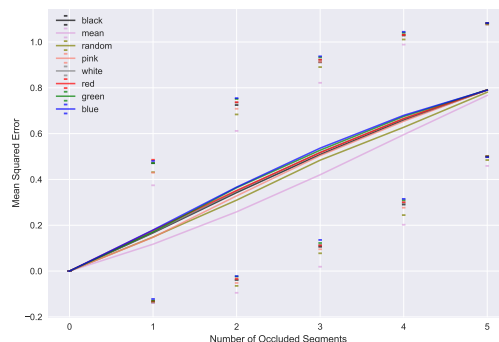
(c) Mean squared error for a 20-segment partition.



(d) Mean squared error for a 15-segment partition.



(e) Mean squared error for a 10-segment partition.



(f) Mean squared error for a 5-segment partition.

**Figure C.1:** Mean squared error calculated between the top prediction of an image (probability estimate) and predictions of the same class when progressively occluding a higher number of segments with a given colouring strategy. We use eight different approaches, the RGB (Red, Green, Blue) colour encodings of which are: white (255, 255, 255); black (0, 0, 0); red (255, 0, 0); green (0, 255, 0); blue (0, 0, 255); pink (255, 192, 203); random – drawn from a uniformly distributed colour space separately for each super-pixel of an individual image; and mean – each segment is occluded with its mean RGB colour. The panels show that the mean occlusion strategy is not as effective at hiding information from the black box as using a single colour for all of the super-pixels (regardless of the colour choice). Similarly, randomising the occlusion colour for each individual segment does not seem to have the detrimental effects observed for the mean colouring. The plots also reveal that when an image is split into more segments, the ineffectiveness of the mean-colouring approach gets magnified due to the increased colour uniformity of individual super-pixels. (See Appendix C.1 for the description of our experimental setup.)

blurry regions will be difficult to “remove” with the *mean*-colour occlusion strategy – refer to the *Information Removal and Loss* part of Section 3.3.2 for an in-detail discussion.

## C.2 Tree-based Interpretable Representations of Tabular Data

Faithfulness of interpretable representations is important for capturing the local behaviour of a black box. In Section 3.3.2, we argue that this property can be measured by investigating purity of hyper-rectangles that constitute an IR. In particular, if the underlying task is crisp classification, we can use the *Gini impurity* ( $\mathcal{L}_G$ ) defined in Equation C.1, where  $H_i$  is a set of data points and their labels  $(x, y)$  situated within the  $i^{\text{th}}$  hyper-rectangle and  $C$  is the set of all the unique labels  $c$ .

$$\begin{aligned} p_{H_i}(c) &= \frac{1}{|H_i|} \sum_{(x,y) \in H_i} \mathbb{1}_{y=c} \\ \mathcal{L}_G(H_i) &= \sum_{c \in C} p_{H_i}(c) (1 - p_{H_i}(c)) \end{aligned} \quad (\text{C.1})$$

When the task is regression or probabilistic classification (the formula applies to each individual class separately in the latter case), on the other hand, we can use the *mean squared error* ( $\mathcal{L}_{\text{MSE}}$ ) – defined in Equation C.2 – to quantify numerical uniformity of each hyper-rectangle.

$$\begin{aligned} \bar{y}_{H_i} &= \frac{1}{|H_i|} \sum_{(x,y) \in H_i} y \\ \mathcal{L}_{\text{MSE}}(H_i) &= \frac{1}{|H_i|} \sum_{(x,y) \in H_i} (y - \bar{y}_{H_i})^2 \end{aligned} \quad (\text{C.2})$$

When combining scores of multiple hyper-rectangles to assess the overall quality  $\mathcal{Q}$  of an interpretable representation, we opt for a weighted average of individual scores  $\mathcal{L}$  to account for the (possibly unbalanced) distribution of data points among these partitions – see Equation C.3.

$$\mathcal{Q} = \frac{1}{\sum_{H_i} |H_i|} \sum_{H_i} |H_i| \mathcal{L}(H_i) \quad (\text{C.3})$$

This formalisation prompted us to propose using decision trees to divide a feature space according to the separation criteria given by Equations C.1 and C.2 respectively for crisp classification and regression/probabilistic classification tasks when building interpretable representations of tabular data. In case of surrogate explainers, the target variable becomes the collection of black-box predictions generated for instances drawn from the explained neighbourhood. While this approach appears sound, we further support it with quantitative experiments on four different data sets, two of which are classification and the other two regression problems:

- wine recognition<sup>1</sup> (classification),

<sup>1</sup><https://archive.ics.uci.edu/ml/datasets/wine>

- breast cancer Wisconsin diagnostic<sup>2</sup> (classification),
- Boston house prices<sup>3</sup> (regression), and
- diabetes<sup>4</sup> (regression).

Using these data, we compare *quartile* and *tree-based* interpretable representations in two variants: *global* – based on a single discretisation created for all of the data points, and *local* – based on a collection of distinct discretisations composed separately for each individual data point. The results of our experiments are depicted in Figure C.2, which shows that tree-based IRs require a fraction of the expressiveness (unique encodings in the interpretable space) used by the quartile-based IRs to achieve a comparable level of hyper-rectangle purity, especially for the *local* interpretable representations.

**Quartile-based Interpretable Representation** This IR is based on quartile discretisation of continuous features [129]. The partition of the data space is *global*, i.e., with respect to the entire data set, however each individual instance receives a distinct IR due to the binarisation step that follows. Therefore, global evaluation is computed on the entire *discretised* data set using the formula given by Equation C.3. Local IR purity, on the other hand, is calculated individually for each instance in the data set based on its distinct binary interpretable representation. This validation is performed for a subset of data that, centred around the explained data point, is within the radius of 60 per cent of the maximum Euclidean distance computed between any two instances in the data set, which simulates locality of the IR.

**Tree-based Interpretable Representation** This IR is based on a partition of the feature space determined by the thresholds learnt with a tree model. Global evaluation is performed by computing purity of the hyper-rectangles created by a tree fitted to the entire data set and validated on this training data, which is a fair comparison given that the quartile-based IR can also access the whole data set. Local IR purity, on the other hand, is calculated independently for each instance in the data set by learning a tree model on a subset of data that, centred around the explained data point, is within the radius of 60 per cent of the maximum Euclidean distance computed between any two instances in the data set, with the same data subset used to compute purity of the resulting hyper-rectangles. Notably, the local method is somewhat disadvantaged when compared to its quartile-based counterpart since the trees are fitted to a subset of the data, whereas the quartile discretisation has access to all the data.

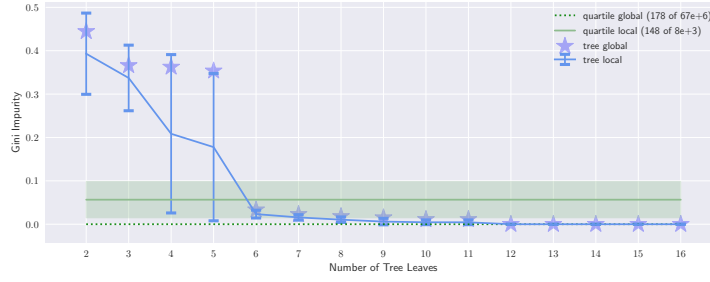
---

<sup>2</sup>[https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+\(Diagnostic\)](https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic))

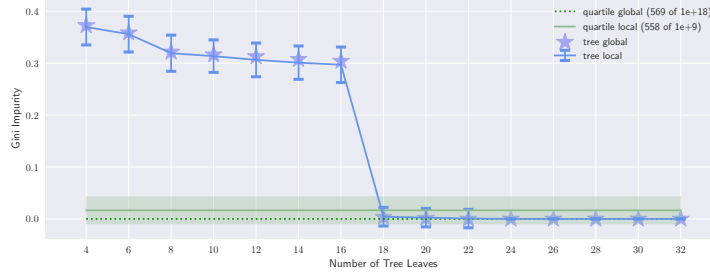
<sup>3</sup><https://archive.ics.uci.edu/ml/machine-learning-databases/housing>

<sup>4</sup><https://www4.stat.ncsu.edu/~boos/var.select/diabetes.html>

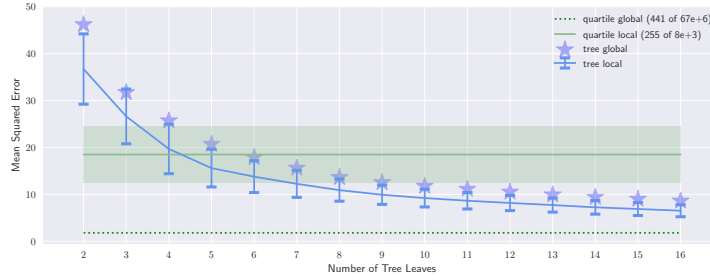
## C.2. TREE-BASED INTERPRETABLE REPRESENTATIONS OF TABULAR DATA



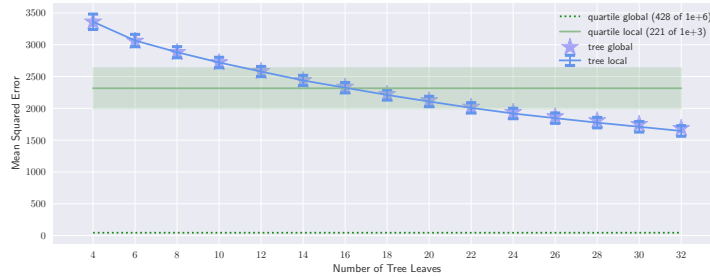
(a) Weighted average of the Gini impurity computed for IRs generated for the *wine* data set.



(b) Weighted average of the Gini impurity computed for IRs generated for the *cancer* data set.



(c) Weighted average of the mean-squared-error purity computed for IRs generated for the *housing* data set.



(d) Weighted average of the mean-squared-error purity computed for IRs generated for the *diabetes* data set.

**Figure C.2:** Interpretable representations based on decision trees achieve higher purity of hyper-rectangles (y-axes, lower is better) with fewer encodings (x-axes), i.e., they are more flexible and expressive. The number of unique encodings used by quartile-based IRs is constant for a data set and it is displayed in the legend (presented as the number of encodings used, out of the theoretical limit supported by the representation); whereas for tree-based IRs, it is equivalent to the number of leaves, which is recorded on the x-axes. Panels (c) and (d) do not capture the tree width at which this IR *globally* outperforms the quartile-based IR, which is 80 (compared to 441) and 224 (compared to 428) respectively for the *housing* and *diabetes* data sets. For more details, see the *Interpreting the Results* paragraph in Appendix C.2.

**Interpreting the Results** Figure C.2 shows the results of our experiments for the aforementioned data sets, highlighting the purity of interpretable representations achieved with a range of different tree widths (x-axes). In each case, the number of tree leaves is compared to the number of unique hyper-rectangles generated by the corresponding quartile-based IR; specifically, its discretisation step in the *global* variant and its binarisation step in the *local* variant. Each plot depicts the weighted Gini impurity or mean squared error (y-axes, Equation C.3), respectively for classification and regression tasks, computed for all the hyper-rectangles of each individual IR. The dotted green line labelled as “quartile global” is the measure of purity for the quartile *discretisation* that underlies this type of an interpretable representation – it is unique to a data set. The solid green line surrounded by the shading – marked as “quartile local” – corresponds to the mean and standard deviation of the hyper-rectangle purity computed for the quartile-based IR (*discretisation* followed by *binarisation*) of each individual instance in a data set. Equivalent calculations are done for the global and local tree-based IRs for a range of tree widths: “tree global” denoted by the blue  $\star$  symbol and “tree local” depicted by the blue line, with the error bars indicating the standard deviation. In all of the plots, a lower number on the y-axes – weighted “impurity” of an IR – is better.

The pair of numbers placed in brackets next to the “quartile global” and “quartile local” labels in the legend of each plot communicates how many distinct hyper-rectangles for the global approach, and their combinations for the local approach, are being used by the validation data, out of all the possible unique values that, respectively, the quartile discretisation and its binarisation can theoretically encode. These quantities are directly comparable to the width of trees (x-axes) used for partitioning the feature space in the tree-based IRs. Given a lack of a black-box model, whose predictions should be used to capture the distribution of the target variable within each hyper-rectangle, we utilise the ground truth provided with the aforementioned data sets instead – this proxy does not affect our experiments in any way. In summary, Figure C.2 illustrates that interpretable representations created with decision trees are more pure than their quartile-based alternatives, therefore they are superior at capturing the intricacies of the underlying labelling mechanism, whatever it may be. Furthermore, they achieve better performance with just a fraction of the encodings required by the other method, i.e., they are more expressive because of the elaborate mechanism used by decision trees to partition and merge a feature space.

### C.3 Binary Interpretable Representations of Tabular Data

When analysing the behaviour of algorithmic black boxes with surrogate explainers, linear models can be used to quantify (positive or negative) influence of interpretable concepts (extracted from the data in question) on individual predictions [45, 129]. For some binary interpretable domains, however, such an approach is inherently flawed. In this section, we show how the influence of an interpretable concept measured by the coefficients of a linear model may be deceiving.

This phenomenon is particularly prominent for tabular data transformed into the binary interpretable representation introduced earlier in the *Tabular Interpretable Representations* part of Section 3.2.1. The insights stemming from our analysis can be used to manipulate surrogate explanations, e.g., those produced with LIME [129], by using specially crafted, yet perfectly valid, IR discretisation and data sample.

Our results are based on the analytical solution to unweighted ( $\Theta$ ) and weighted ( $\Theta_{\mathbf{W}}$ ) Ordinary Least Squares (OLS) outlined in the equations below, where  $\mathbf{W}$  is the weight matrix,  $\mathbf{X}$  is the binary interpretable representation data matrix, and  $\mathbf{y}$  is a vector holding the corresponding black-box predictions. In our analysis, we assume that the black box is a probabilistic classifier, in which case  $\mathbf{y}$  captures probabilities of the explained class; nonetheless, a similar line of reasoning applies to regressors and crisp classifiers. In the latter scenario, the elements of  $\mathbf{y}$  are assumed to be 1 when the black-box predictions are the same as the explained class, and 0 for any other class. Modelling  $\mathbf{y}$  in such a way generates one-vs-rest explanations (i.e., evidence for the black box predicting the explained rather than any other class), akin to the insights produced for probabilistic black boxes, for which the linear surrogate *only* models the *explained class* probabilities. Regardless, both approaches capture the influence of interpretable concepts – measured by the coefficients of a linear model – on a selected class when tasked with telling it apart from the other classes.

$$\begin{aligned}\Theta &= (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y} \\ \Theta_{\mathbf{W}} &= (\mathbf{X}^T \mathbf{W} \mathbf{X})^{-1} \mathbf{X}^T \mathbf{W} \mathbf{y}\end{aligned}$$

In the interest of brevity and readability, we analyse tabular data with two numerical features – similar to the example shown in Figure 3.11 – nonetheless our findings generalise to an arbitrary number of attributes that are both categorical and numerical. In a generic setting, for  $n$  features there will be  $n$  binary concepts with  $2^n$  unique encodings (cardinality) in the interpretable representation. If additionally we choose to model the intercept of the linear surrogate, a phantom all-1 column vector is inserted at the front of the data matrix  $\mathbf{X}$ . Therefore, the  $\mathbf{X}^T \mathbf{X}$  and  $\mathbf{X}^T \mathbf{W} \mathbf{X}$  components of  $\Theta$  and  $\Theta_{\mathbf{W}}$  respectively are square matrices of  $n \times n$  shape sans the intercept or  $(n + 1) \times (n + 1)$  when the intercept is modelled.

Figure 3.11 depicts a simplistic view of sampling for two numerical features with just one data point in each hyper-rectangle. In reality, however, we should expect their large number since it allows to better approximate the behaviour of the underlying black box, especially when we are dealing with a lot of features. For our toy problem, the binary interpretable representation data matrix  $\mathbf{X}$  – with the first column (red) inserted to model the intercept and the remaining columns

(blue) representing the binary data – is:

$$\mathbf{X} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix},$$

which gives:

$$\mathbf{X}^T \mathbf{X} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 9 & 3 & 3 \\ 3 & 3 & 1 \\ 3 & 1 & 3 \end{bmatrix}.$$

Since some of the hyper-rectangles are merged when transitioning from the discrete into the binary interpretable representation,  $\mathbf{X}$  contains duplicated rows. The influence of this phenomenon is magnified even more when multiple data points are sampled within a single hyper-rectangle. Without loss of generality, we can use the *weighted* variant of OLS with the data set  $\mathbf{X}$  containing only one copy of each unique binary data point and the weights corresponding to their counts. In this case:

$$\mathbf{X} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \quad \text{and} \quad \mathbf{W} = \begin{bmatrix} w_{11} & 0 & 0 & 0 \\ 0 & w_{10} & 0 & 0 \\ 0 & 0 & w_{01} & 0 \\ 0 & 0 & 0 & w_{00} \end{bmatrix},$$

where  $w_{ij}$  is the count of data points residing in all of the hyper-rectangles that are assigned the  $(i, j)$  coordinates in the binary interpretable representation – see the  $(x^*, y^*)$  coordinates in Figure 3.11 for reference. Therefore, for an arbitrary number of data points with two numerical

features when modelling the intercept:

$$\begin{aligned}
 \mathbf{X}^T \mathbf{W} \mathbf{X} &= \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} w_{11} & 0 & 0 & 0 \\ 0 & w_{10} & 0 & 0 \\ 0 & 0 & w_{01} & 0 \\ 0 & 0 & 0 & w_{00} \end{bmatrix} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \\
 &= \begin{bmatrix} w_{11} & w_{10} & w_{01} & w_{00} \\ w_{11} & w_{10} & 0 & 0 \\ w_{11} & 0 & w_{01} & 0 \end{bmatrix} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \\
 &= \begin{bmatrix} \sum w_{ij} & w_{11} + w_{10} & w_{11} + w_{01} \\ w_{11} + w_{10} & w_{11} + w_{10} & w_{11} \\ w_{11} + w_{01} & w_{11} & w_{11} + w_{01} \end{bmatrix}.
 \end{aligned}$$

Notably, a derivation of this weighted formula for the toy data presented in Figure 3.11 – where  $w_{11} = 1$ ,  $w_{10} = 2$ ,  $w_{01} = 2$  and  $w_{00} = 4$  – agrees with the previous result computed directly for  $\mathbf{X}^T \mathbf{X}$ .

Next, we analyse the second component of the  $\Theta_{\mathbf{W}}$  formula:

$$\begin{aligned}
 \mathbf{X}^T \mathbf{W} \mathbf{y} &= \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} w_{11} & 0 & 0 & 0 \\ 0 & w_{10} & 0 & 0 \\ 0 & 0 & w_{01} & 0 \\ 0 & 0 & 0 & w_{00} \end{bmatrix} \times \begin{bmatrix} y_{11} \\ y_{10} \\ y_{01} \\ y_{00} \end{bmatrix} \\
 &= \begin{bmatrix} w_{11} & w_{10} & w_{01} & w_{00} \\ w_{11} & w_{10} & 0 & 0 \\ w_{11} & 0 & w_{01} & 0 \end{bmatrix} \times \begin{bmatrix} y_{11} \\ y_{10} \\ y_{01} \\ y_{00} \end{bmatrix} \\
 &= \begin{bmatrix} \sum w_{ij} y_{ij} \\ w_{11} y_{11} + w_{10} y_{10} \\ w_{11} y_{11} + w_{01} y_{01} \end{bmatrix}.
 \end{aligned}$$

This formulation, however, implies that all of the data points that share the same  $(i, j)$  coordinates in the binary interpretable representation have the same target value (black-box prediction)  $y_{ij}$ . To allow multiple copies of the same data point with distinct target values, we generalise this result by going back to  $\Theta$ , which is the solution to the classic OLS. This approach is valid since weighted OLS for which the weights represent the count of each unique data point is equivalent to classic OLS for a data set whose instances are duplicated according to the counts given by the corresponding weights.

Let us denote  $f : \mathcal{X} \rightarrow \mathcal{Y}$  as the black-box model and  $IR : \mathcal{X} \rightarrow \mathcal{X}^*$  as the transformation function from tabular data  $\mathcal{X}$  into their binary interpretable representation  $\mathcal{X}^*$ . Let us further define  $\mathcal{W}_{ij} = \{x \in \mathbf{X} : IR(x) = (i, j)\}$  as the set of all the data points that are assigned the same binary

interpretable representation  $(i, j)$ , and  $\mathcal{W} = \mathbf{X}$  as the set of all the data points. Now, recall that  $w_{ij}$  is the count of data points whose binary interpretable representation is  $(i, j)$ , therefore  $|\mathcal{W}_{ij}| = w_{ij}$  and  $|\mathcal{W}| = \sum w_{ij}$ . Without loss of generality, we can reformulate the  $\mathbf{X}^T \mathbf{W} \mathbf{y}$  part of  $\Theta_{\mathbf{W}}$  as  $\mathbf{X}^T \mathbf{y}$  by summing over the black-box predictions corresponding to the relevant hyper-rectangles:

$$\mathbf{X}^T \mathbf{y} = \begin{bmatrix} \sum w_{ij} y_{ij} \\ w_{11} y_{11} + w_{10} y_{10} \\ w_{11} y_{11} + w_{01} y_{01} \end{bmatrix} = \begin{bmatrix} \sum_{i \in \mathcal{W}} y_i \\ \sum_{i \in \mathcal{W}_{11} \cup \mathcal{W}_{10}} y_i \\ \sum_{i \in \mathcal{W}_{11} \cup \mathcal{W}_{01}} y_i \end{bmatrix}.$$

This step allows us to relax the assumption of duplicated target values  $y_{ij}$ , hence avoid imposing restrictions on the type of the black box (probabilistic, crisp or regressor) and whether the binary representation has full fidelity with respect to the black box.<sup>5</sup>

Finally, to better understand the meaning of influence-based explanations, we reformulate the sum of black-box predictions into their average:

$$\begin{aligned} \mathbf{X}^T \mathbf{y} &= \begin{bmatrix} \sum_{i \in \mathcal{W}} y_i \\ \sum_{i \in \mathcal{W}_{11} \cup \mathcal{W}_{10}} y_i \\ \sum_{i \in \mathcal{W}_{11} \cup \mathcal{W}_{01}} y_i \end{bmatrix} = \begin{bmatrix} \sum_{i \in \mathcal{W}} y_i / \sum w_{ij} * \sum w_{ij} \\ \sum_{i \in \mathcal{W}_{11} \cup \mathcal{W}_{10}} y_i / (w_{11} + w_{10}) * (w_{11} + w_{10}) \\ \sum_{i \in \mathcal{W}_{11} \cup \mathcal{W}_{01}} y_i / (w_{11} + w_{01}) * (w_{11} + w_{01}) \end{bmatrix} = \begin{bmatrix} \bar{y}_{\mathcal{W}} * \sum w_{ij} \\ \bar{y}_{\mathcal{W}_{11} \cup \mathcal{W}_{10}} * (w_{11} + w_{10}) \\ \bar{y}_{\mathcal{W}_{11} \cup \mathcal{W}_{01}} * (w_{11} + w_{01}) \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} \bar{y}_{\mathcal{W}} * \sum w_{ij} \\ \bar{y}_{\mathcal{W}_{11} \cup \mathcal{W}_{10}} * (w_{11} + w_{10}) \\ \bar{y}_{\mathcal{W}_{11} \cup \mathcal{W}_{01}} * (w_{11} + w_{01}) \end{bmatrix} \\ &= \begin{bmatrix} \sum w_{ij} & 0 & 0 \\ 0 & w_{11} + w_{10} & 0 \\ 0 & 0 & w_{11} + w_{01} \end{bmatrix} \times \begin{bmatrix} \bar{y}_{\mathcal{W}} \\ \bar{y}_{\mathcal{W}_{11} \cup \mathcal{W}_{10}} \\ \bar{y}_{\mathcal{W}_{11} \cup \mathcal{W}_{01}} \end{bmatrix}, \end{aligned}$$

<sup>5</sup>Note that in the original formulation of  $\mathbf{X}^T \mathbf{W} \mathbf{y}$ , instances within each hyper-rectangle determined by the underlying interpretable representation are assumed to share the same black-box prediction. This constraint makes our solution almost impossible to apply to black-box regressors and probabilistic classifiers; it also means that for crisp classifiers the binarised data space would have to respect the black-box decision surface (to achieve full fidelity).

and combine this result with  $\mathbf{X}^T \mathbf{W} \mathbf{X}$ :

$$\begin{aligned}
 & \begin{bmatrix} \sum w_{ij} & w_{11} + w_{10} & w_{11} + w_{01} \\ w_{11} + w_{10} & w_{11} + w_{10} & w_{11} \\ w_{11} + w_{01} & w_{11} & w_{11} + w_{01} \end{bmatrix}^{-1} \times \begin{bmatrix} \sum w_{ij} & 0 & 0 \\ 0 & w_{11} + w_{10} & 0 \\ 0 & 0 & w_{11} + w_{01} \end{bmatrix} \times \begin{bmatrix} \bar{y}_{\mathcal{W}} \\ \bar{y}_{\mathcal{W}_{11} \cup \mathcal{W}_{10}} \\ \bar{y}_{\mathcal{W}_{11} \cup \mathcal{W}_{01}} \end{bmatrix} \\
 &= \begin{bmatrix} \sum w_{ij} & w_{11} + w_{10} & w_{11} + w_{01} \\ w_{11} + w_{10} & w_{11} + w_{10} & w_{11} \\ w_{11} + w_{01} & w_{11} & w_{11} + w_{01} \end{bmatrix}^{-1} \times \begin{bmatrix} \frac{1}{\sum w_{ij}} & 0 & 0 \\ 0 & \frac{1}{w_{11} + w_{10}} & 0 \\ 0 & 0 & \frac{1}{w_{11} + w_{01}} \end{bmatrix}^{-1} \times \begin{bmatrix} \bar{y}_{\mathcal{W}} \\ \bar{y}_{\mathcal{W}_{11} \cup \mathcal{W}_{10}} \\ \bar{y}_{\mathcal{W}_{11} \cup \mathcal{W}_{01}} \end{bmatrix} \\
 &= \left( \begin{bmatrix} \frac{1}{\sum w_{ij}} & 0 & 0 \\ 0 & \frac{1}{w_{11} + w_{10}} & 0 \\ 0 & 0 & \frac{1}{w_{11} + w_{01}} \end{bmatrix} \times \begin{bmatrix} \sum w_{ij} & w_{11} + w_{10} & w_{11} + w_{01} \\ w_{11} + w_{10} & w_{11} + w_{10} & w_{11} \\ w_{11} + w_{01} & w_{11} & w_{11} + w_{01} \end{bmatrix} \right)^{-1} \times \begin{bmatrix} \bar{y}_{\mathcal{W}} \\ \bar{y}_{\mathcal{W}_{11} \cup \mathcal{W}_{10}} \\ \bar{y}_{\mathcal{W}_{11} \cup \mathcal{W}_{01}} \end{bmatrix} \\
 &= \begin{bmatrix} 1 & \frac{w_{11} + w_{10}}{\sum w_{ij}} & \frac{w_{11} + w_{01}}{\sum w_{ij}} \\ 1 & 1 & \frac{w_{11}}{w_{11} + w_{10}} \\ 1 & \frac{w_{11}}{w_{11} + w_{01}} & 1 \end{bmatrix}^{-1} \times \begin{bmatrix} \bar{y}_{\mathcal{W}} \\ \bar{y}_{\mathcal{W}_{11} \cup \mathcal{W}_{10}} \\ \bar{y}_{\mathcal{W}_{11} \cup \mathcal{W}_{01}} \end{bmatrix}.
 \end{aligned}$$

This outcome allows us to draw conclusions about the meaning of interpretable concept influence given by the coefficients of a surrogate linear model when the intercept is modelled (red & blue) and without it (blue). In particular, the influence of interpretable concepts is *solely* based on:

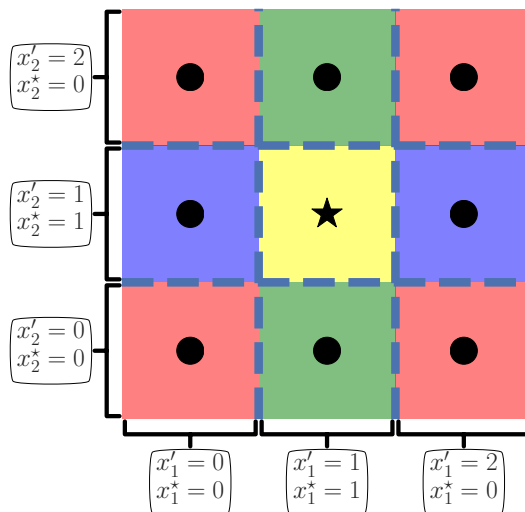
- **the proportion determined by the number of the data points** residing in the explained hyper-rectangle ( $\mathcal{W}_{11}$ ) divided by the hyper-rectangles aligned with the explained hyper-rectangle along every axis:  $\mathcal{W}_{11} \cup \mathcal{W}_{10}$  for the first feature and  $\mathcal{W}_{11} \cup \mathcal{W}_{01}$  for the second; and
- **the average value predicted** in the latter two subspaces –  $\mathcal{W}_{11} \cup \mathcal{W}_{10}$  and  $\mathcal{W}_{11} \cup \mathcal{W}_{01}$  – by the black box (appropriately scaled when the intercept is modelled).

For example, consider Figure C.3 where  $x_1^*$  denotes the first binary interpretable feature and  $x_2^*$  the second. In this case,  $\mathcal{W}_{11}$  is the yellow hyper-rectangle;  $\mathcal{W}_{11} \cup \mathcal{W}_{10}$  is the union of the yellow and green hyper-rectangles; and  $\mathcal{W}_{11} \cup \mathcal{W}_{01}$  is the combination of yellow and blue hyper-rectangles. Then,  $\bar{y}_{\mathcal{W}_{11} \cup \mathcal{W}_{10}}$  is the average prediction in the vertical green&yellow segment, and  $\bar{y}_{\mathcal{W}_{11} \cup \mathcal{W}_{01}}$  is the average prediction in the horizontal blue&yellow stripe.

When modelled, the intercept value is additionally determined by:

- *the proportion* given by the number of data points in the hyper-rectangles aligned with the explained hyper-rectangle along every axis divided by the total number of data points; and
- *the average* value predicted by the black box for all the data points.

Intuitively, the instances not aligned with the explained hyper-rectangle – red blocks in Figure C.3 – are assigned the (0,0) coordinates in the binary interpretable representation, therefore they



**Figure C.3:** Example of discrete  $(x'_1, x'_2)$  and binary  $(x_1^*, x_2^*)$  interpretable representations of tabular data. ★ represents the explained instance.

cannot contribute to the feature coefficients of a linear model, just the intercept. This can be easily seen with the  $g(\mathbf{x}^*; \Theta) = \sum_{i=0}^n \Theta_i x_i^*$  formula where  $x_0^* = 1$  is the phantom feature and the remaining data features  $x_1^*, \dots, x_n^*$  are 0; these instances can only influence the intercept  $\Theta_0$ .

An important insight uncovered by our results is **partial insignificance of the discretisation quality** given a fixed number of data points placed in the identified collections of relevant hyper-rectangles. Using this property, we can manipulate the explanation by altering the number of data points in meaningful partitions, with the discretisation faithfulness having relatively small influence. For example, consider the two discretisations depicted earlier in Figure 3.7, assuming that the explained hyper-rectangle is  $(x', y') = (1, 1)$  for both sub-plots and that the  $(x', y') = (1, 0)$  and  $(x', y') = (1, 1)$  partitions in Figure 3.7b have *three* additional data points each. In this case, when modelling the influence of interpretable components without the intercept, the only difference between these two sub-plots are the black-box predictions of the instances placed in the  $(x', y') = (1, 0)$  and  $(x', y') = (1, 1)$  hyper-rectangles since:

**Figure 3.7a**  $w_{11} = 4$ ,  $w_{01} = 4 + 4 = 8$  and  $w_{10} = 4$ , leading to  $\frac{w_{11}}{w_{11} + w_{10}} = \frac{4}{4+4} = \frac{1}{2}$  and  $\frac{w_{11}}{w_{11} + w_{01}} = \frac{4}{4+8} = \frac{1}{3}$ ; and

**Figure 3.7b**  $w_{11} = 2 + 3 = 5$ ,  $w_{01} = 4 + 4 + 2 = 10$  and  $w_{10} = 2 + 3 = 5$ , leading to  $\frac{w_{11}}{w_{11} + w_{10}} = \frac{5}{5+5} = \frac{1}{2}$  and  $\frac{w_{11}}{w_{11} + w_{01}} = \frac{5}{5+10} = \frac{1}{3}$ .

Depending on the gradient smoothness of the underlying probabilistic black box, these explanations may slightly differ. However, if the additional six instances are placed such that the average black-box predictions of  $\mathcal{W}_{11} \cup \mathcal{W}_{10}$  and  $\mathcal{W}_{11} \cup \mathcal{W}_{01}$  are identical across both discretisations, the resulting explanations will be the same. (In general, it is easier to manipulate the explanations when dealing with crisp predictions instead of probabilities as we only have to consider which

side of the black-box decision surface – if one runs across a given hyper-rectangle – to place each data point.) Notably, this observation highlights that partitioning numerical features into more than three splits is unnecessary, with the bin boundaries enclosing the explained instance being the only important ones.



## BIBLIOGRAPHY

- [1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: A system for large-scale machine learning. In *12<sup>th</sup> USENIX symposium on operating systems design and implementation (OSDI 16)*, pages 265–283, 2016.  
Cited on page(s): 230.
- [2] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2274–2282, 2012.  
Cited on page(s): xxv, 238.
- [3] Evan Ackerman. Three small stickers in intersection can cause Tesla autopilot to swerve into wrong lane. *IEEE Spectrum*, April, 1, 2019.  
Cited on page(s): 17.
- [4] Philip Adler, Casey Falk, Sorelle A Friedler, Tionney Nix, Gabriel Rybeck, Carlos Scheidegger, Brandon Smith, and Suresh Venkatasubramanian. Auditing black-box models for indirect influence. *Knowledge and Information Systems*, 54(1):95–122, 2018.  
Cited on page(s): 234.
- [5] Alekh Agarwal, Alina Beygelzimer, Miroslav Dudik, John Langford, and Hanna Wallach. A reductions approach to fair classification. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35<sup>th</sup> International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 60–69, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR. URL <http://proceedings.mlr.press/v80/agarwal18a.html>.  
Cited on page(s): 234.
- [6] Divyakant Agrawal, Philip Bernstein, Bertino Elisa, Davidson Susan, Dayal Umeshwar, Michael Franklin, and Y Papakonstantinou. Challenges and opportunities with big data: A white paper prepared for the computing community consortium. *Committee of the Computing Research Association*, 2012.  
Cited on page(s): 22, 62.

## BIBLIOGRAPHY

---

- [7] Arjun R Akula, Sinisa Todorovic, Joyce Y Chai, and Song-Chun Zhu. Natural language interaction with explainable AI models. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops on Explainable AI*, June 2019.  
Cited on page(s): 185.
- [8] Saleema Amershi, Maya Cakmak, William Bradley Knox, and Todd Kulesza. Power to the people: The role of humans in interactive machine learning. *AI Magazine*, 35(4):105–120, 2014.  
Cited on page(s): 181.
- [9] Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. Machine bias. *ProPublica, May*, 23:2016, 2016.  
Cited on page(s): 17.
- [10] Abdallah Arioua and Madalina Croitoru. Formalizing explanatory dialogues. In *International Conference on Scalable Uncertainty Management*, pages 282–297. Springer, 2015.  
Cited on page(s): 9, 127, 158, 183.
- [11] Matthew Arnold, Rachel KE Bellamy, Michael Hind, Stephanie Houde, Sameep Mehta, Aleksandra Mojsilovic, Ravi Nair, Karthikeyan Natesan Ramamurthy, Alexandra Olteanu, David Piorkowski, Darrell Reimer, John Richards, Jason Tsay, and Kush R Varshney. FactSheets: Increasing trust in AI services through supplier’s declarations of conformity. *IBM Journal of Research and Development*, 63(4/5):6:1–6:13, July 2019. ISSN 0018-8646. doi: 10.1147/JRD.2019.2942288.  
Cited on page(s): 56, 57, 59.
- [12] Roy F Baumeister and Leonard S Newman. Self-regulation of cognitive inference and decision processes. *Personality and Social Psychology Bulletin*, 20(1):3–19, 1994.  
Cited on page(s): 9.
- [13] Boris Beizer. *Black-box testing: Techniques for functional testing of software and systems*. John Wiley & Sons, Inc., 1995.  
Cited on page(s): 1.
- [14] Rachel K. E. Bellamy, Kuntal Dey, Michael Hind, Samuel C. Hoffman, Stephanie Houde, Kalapriya Kannan, Pranay Lohia, Jacquelyn Martino, Sameep Mehta, Aleksandra Mojsilović, Seema Nagar, Karthikeyan Natesan Ramamurthy, John Richards, Diptikalyan Saha, Prasanna Sattigeri, Moninder Singh, Kush R. Varshney, and Yunfeng Zhang. AI Fairness 360: An extensible toolkit for detecting and mitigating algorithmic bias. *IBM Journal of Research and Development*, 63(4/5):4:1–4:15, 2019.  
Cited on page(s): 234.

- [15] Emily M Bender and Batya Friedman. Data statements for natural language processing: Toward mitigating system bias and enabling better science. *Transactions of the Association for Computational Linguistics*, 6:587–604, 2018. doi: 10.1162/tacl\_a\_00041. URL <https://www.aclweb.org/anthology/Q18-1041>.  
Cited on page(s): 6, 37, 56, 57, 59.
- [16] Benjamin Bengfort and Rebecca Bilbro. Yellowbrick: Visualizing the scikit-learn model selection process. *Journal of Open Source Software*, 4(35):1075, 2019.  
Cited on page(s): 234.
- [17] Benjamin Bengfort, Rebecca Bilbro, and Kristen McIntyre. Yellowbrick v1.0.1, October 2019. URL <https://doi.org/10.5281/zenodo.3474252>.  
Cited on page(s): 234.
- [18] Umang Bhatt, Alice Xiang, Shubham Sharma, Adrian Weller, Ankur Taly, Yunhan Jia, Joydeep Ghosh, Ruchir Puri, José MF Moura, and Peter Eckersley. Explainable machine learning in deployment. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 648–657, 2020.  
Cited on page(s): 2.
- [19] Or Biran and Kathleen McKeown. Justification narratives for individual classifications. In *Proceedings of the AutoML workshop at ICML*, volume 2014, pages 1–7, 2014.  
Cited on page(s): 9, 46, 178.
- [20] Dimitri Bohlender and Maximilian A Köhl. Towards a characterization of explainable systems. *GI-Dagstuhl Seminar 19023 Explainable Software for Cyber-Physical Systems (ES4CPS)*, 2019. URL <https://arxiv.org/abs/1902.03096>. arXiv preprint arXiv:1902.03096.  
Cited on page(s): 42.
- [21] Hanen Borchani, Gherardo Varando, Concha Bielza, and Pedro Larrañaga. A survey on multi-output regression. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 5(5):216–233, 2015.  
Cited on page(s): 94.
- [22] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.  
Cited on page(s): xxiv, 6, 7, 19, 143.
- [23] Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. *Classification and regression trees*. CRC press, 1984.  
Cited on page(s): xxiii, 105, 136.

## BIBLIOGRAPHY

---

- [24] Mario Bunge. A general black box theory. *Philosophy of Science*, 30(4):346–358, 1963.  
Cited on page(s): 1.
- [25] Joy Buolamwini and Timnit Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In Sorelle A. Friedler and Christo Wilson, editors, *Proceedings of the 1<sup>st</sup> Conference on Fairness, Accountability and Transparency*, volume 81 of *Proceedings of Machine Learning Research*, pages 77–91, New York, NY, USA, 23–24 Feb 2018. PMLR. URL <http://proceedings.mlr.press/v81/buolamwini18a.html>.  
Cited on page(s): 16.
- [26] Pete Chapman, Julian Clinton, Randy Kerber, Thomas Khabaza, Thomas Reinartz, Colin Shearer, Rudiger Wirth, et al. CRISP-DM 1.0: Step-by-step data mining guide. *SPSS inc*, 9:13, 2000.  
Cited on page(s): xxiii, 22, 62.
- [27] Chaofan Chen, Kangcheng Lin, Cynthia Rudin, Yaron Shaposhnik, Sijia Wang, and Tong Wang. An interpretable model with globally consistent explanations for credit risk. *2018 Workshop on Challenges and Opportunities for AI in Financial Services: the Impact of Fairness, Explainability, Accuracy, and Privacy at the 32<sup>nd</sup> Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada*, 2018. URL <https://arxiv.org/abs/1811.12615>. arXiv preprint arXiv:1811.12615.  
Cited on page(s): 5.
- [28] Mark Craven and Jude W Shavlik. Extracting tree-structured representations of trained networks. In *Advances in neural information processing systems*, pages 24–30, 1996.  
Cited on page(s): xxv, 6, 20, 63, 100, 126, 129, 158.
- [29] Martin Cutts. *Oxford guide to plain English*. Oxford University Press, 2009.  
Cited on page(s): 90.
- [30] Nils Dahlbäck, Arne Jönsson, and Lars Ahrenberg. Wizard of Oz studies – why and how. *Knowledge-based systems*, 6(4):258–266, 1993.  
Cited on page(s): 176, 178.
- [31] Christian Davenport. Rashomon effect, observation, and data generation. *Media bias, perspective, and state repression*, 2010.  
Cited on page(s): 50.
- [32] Richard Dawkins. The tyranny of the discontinuous mind. *New Statesman*, 19:54–57, 2011.  
Cited on page(s): 2.

- [33] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. IEEE, 2009.  
Cited on page(s): xxiv, 137, 154, 238.
- [34] Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017. URL <https://arxiv.org/abs/1702.08608>.  
Cited on page(s): 34, 50, 51, 58, 153.
- [35] Phan Minh Dung, Robert A Kowalski, and Francesca Toni. Assumption-based argumentation. In *Argumentation in artificial intelligence*, pages 199–218. Springer, 2009.  
Cited on page(s): 8, 14, 40, 180, 183.
- [36] Bradley Efron, Trevor Hastie, Iain Johnstone, Robert Tibshirani, et al. Least angle regression. *The Annals of statistics*, 32(2):407–499, 2004.  
Cited on page(s): xxiv, 74.
- [37] Malin Eiband, Hanna Schneider, Mark Bilandzic, Julian Fazekas-Con, Mareike Haug, and Heinrich Hussmann. Bringing transparency design into practice. In *23<sup>rd</sup> International Conference on Intelligent User Interfaces*, pages 211–223. ACM, 2018.  
Cited on page(s): 40, 46, 47, 52, 53, 184.
- [38] Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. From data mining to knowledge discovery in databases. *AI magazine*, 17(3):37–37, 1996.  
Cited on page(s): xxiv, 22, 62.
- [39] Michael Feldman, Sorelle A Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. Certifying and removing disparate impact. In *Proceedings of the 21<sup>th</sup> ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 259–268. ACM, 2015.  
Cited on page(s): 234.
- [40] Aaron Fisher, Cynthia Rudin, and Francesca Dominici. All models are wrong, but many are useful: Learning a variable’s importance by studying an entire class of prediction models simultaneously. *Journal of Machine Learning Research*, 20(177):1–81, 2019.  
Cited on page(s): 6.
- [41] Ronald A Fisher. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188, 1936.  
Cited on page(s): xv, 7, 90, 109.

- [42] Peter A Flach. *Simply logical: Intelligent reasoning by example*. John Wiley, 1994.  
Cited on page(s): 119.
- [43] Sorelle A. Friedler, Carlos Scheidegger, Suresh Venkatasubramanian, Sonam Choudhary, Evan P. Hamilton, and Derek Roth. A comparative study of fairness-enhancing interventions in machine learning. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, FAT\* '19, pages 329–338, New York, NY, USA, 2019. ACM. ISBN 978-1-4503-6125-5. doi: 10.1145/3287560.3287589. URL <http://doi.acm.org/10.1145/3287560.3287589>.  
Cited on page(s): 234.
- [44] Jerome H Friedman. Greedy function approximation: A gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.  
Cited on page(s): xxiv, 6, 7, 19, 40, 158, 164, 182, 235.
- [45] Jerome H Friedman, Bogdan E Popescu, et al. Predictive learning via rule ensembles. *The Annals of Applied Statistics*, 2(3):916–954, 2008.  
Cited on page(s): xxiv, 19, 25, 66, 75, 95, 100, 244.
- [46] Damien Garreau and Ulrike Luxburg. Explaining the explainer: A first theoretical analysis of LIME. In Silvia Chiappa and Roberto Calandra, editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 1287–1296, Online, 26–28 Aug 2020. PMLR. URL <http://proceedings.mlr.press/v108/garreau20a.html>.  
Cited on page(s): 100, 136.
- [47] Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé III, and Kate Crawford. Datasheets for datasets. *5<sup>th</sup> Workshop on Fairness, Accountability, and Transparency in Machine Learning (FAT/ML 2018) at the 35<sup>th</sup> International Conference on Machine Learning (ICML 2018), Stockholm, Sweden*, 2018. URL <https://arxiv.org/abs/1803.09010>. arXiv preprint arXiv:1803.09010.  
Cited on page(s): 6, 56, 57, 59.
- [48] Alex Goldstein, Adam Kapelner, Justin Bleich, and Emil Pitkin. Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation. *Journal of Computational and Graphical Statistics*, 24(1):44–65, 2015.  
Cited on page(s): xxiv, 7, 19, 40, 164, 235.
- [49] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *Proceedings of the 3<sup>rd</sup> International Conference on Learning Representations (ICLR), San Diego, California*, 2015. URL <http://arxiv.org/abs/1412.6572>.

arXiv preprint arXiv:1412.6572.

Cited on page(s): 17, 21, 49, 171.

- [50] Bryce Goodman and Seth Flaxman. European Union regulations on algorithmic decision-making and a “right to explanation”. *AI Magazine*, 38(3):50–57, Oct. 2017. doi: 10.1609/aimag.v38i3.2741. URL <https://www.aaai.org/ojs/index.php/aimagazine/article/view/2741>.

Cited on page(s): 35, 106.

- [51] H Paul Grice, Peter Cole, Jerry Morgan, et al. Logic and conversation. *1975*, pages 41–58, 1975.

Cited on page(s): 44, 45, 47, 48.

- [52] Adam Grzywaczewski. Training AI for self-driving vehicles: The challenge of scale. *NVIDIA Developer Blog, October*, 2017.

Cited on page(s): 17.

- [53] David Gunning. Broad agency announcement, explainable artificial intelligence (XAI). Technical report, Defense Advanced Research Projects Agency (DARPA), 2016. URL <https://www.darpa.mil/attachments/DARPA-BAA-16-53.pdf>.

Cited on page(s): 3, 4.

- [54] David Gunning. Explainable artificial intelligence (XAI). *Defense Advanced Research Projects Agency (DARPA)*, 2:2, 2017.

Cited on page(s): 3, 48.

- [55] David Gunning and David W Aha. DARPA’s explainable artificial intelligence program. *AI Magazine*, 40(2):44–58, 2019.

Cited on page(s): 177.

- [56] Mark Hall, Daniel Harborne, Richard Tomsett, Vedran Galetic, Santiago Quintana-Amate, Alistair Nottle, and Alun Preece. A systematic method to understand requirements for explainable AI (XAI) systems. In *Proceedings of the IJCAI Workshop on eXplainable Artificial Intelligence (XAI 2019), Macau, China*, 2019.

Cited on page(s): 2, 58.

- [57] Moritz Hardt, Eric Price, and Nathan Srebro. Equality of opportunity in supervised learning. In *Proceedings of the 30<sup>th</sup> International Conference on Neural Information Processing Systems, NIPS’16*, pages 3323–3331, USA, 2016. Curran Associates Inc. ISBN 978-1-5108-3881-9. URL <http://dl.acm.org/citation.cfm?id=3157382.3157469>.

Cited on page(s): 232.

- [58] Clement Henin and Daniel Le Métayer. Towards a generic framework for black-box explanations of algorithmic decision systems. In *Proceedings of the IJCAI Workshop on eXplainable Artificial Intelligence (XAI 2019), Macau, China, 2019*. Cited on page(s): 101, 183, 184.
- [59] Bernease Herman. The promise and peril of human evaluation for model interpretability. *Interpretable ML Symposium at the 31<sup>st</sup> Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 2017*. URL <https://arxiv.org/abs/1711.07414>. arXiv preprint arXiv:1711.07414. Cited on page(s): 49, 51.
- [60] Sarah Holland, Ahmed Hosny, Sarah Newman, Joshua Joseph, and Kasia Chmielinski. The dataset nutrition label: A framework to drive higher data quality standards. *arXiv preprint arXiv:1805.03677*, 2018. URL <https://arxiv.org/abs/1805.03677>. Cited on page(s): 6, 56, 57, 59.
- [61] John D. Hunter. Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007. doi: 10.1109/MCSE.2007.55. Cited on page(s): 230.
- [62] Matthew Hutson. Artificial intelligence faces reproducibility crisis. *Science*, 359(6377):725–726, 2018. ISSN 0036-8075. doi: 10.1126/science.359.6377.725. URL <https://science.sciencemag.org/content/359/6377/725>. Cited on page(s): 228.
- [63] Google Inc. TensorBoard: TensorFlow’s visualization toolkit, 2015. URL <https://www.tensorflow.org/tensorboard>. Cited on page(s): 21.
- [64] Hilary Johnson and Peter Johnson. Explanation facilities and interactive systems. In *Proceedings of the 1<sup>st</sup> international conference on Intelligent user interfaces*, pages 159–166. ACM, 1993. Cited on page(s): 40, 46.
- [65] Hessie Jones. Geoff Hinton dismissed the need for explainable AI: 8 experts explain why he’s wrong, 2018. Cited on page(s): 15.
- [66] Ashish Kapoor, Bongshin Lee, Desney Tan, and Eric Horvitz. Interactive optimization for steering machine classification. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1343–1352. ACM, 2010. Cited on page(s): 163.

- [67] Patrick Gage Kelley, Joanna Bresee, Lorrie Faith Cranor, and Robert W. Reeder. A “nutrition label” for privacy. In *Proceedings of the 5<sup>th</sup> Symposium on Usable Privacy and Security*, SOUPS ’09, pages 4:1–4:12, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-736-3. doi: 10.1145/1572532.1572538. URL <http://doi.acm.org/10.1145/1572532.1572538>. Cited on page(s): 56, 59.
- [68] Been Kim, Cynthia Rudin, and Julie A Shah. The Bayesian case model: A generative approach for case-based reasoning and prototype classification. In *Advances in Neural Information Processing Systems*, pages 1952–1960, 2014. Cited on page(s): 7, 21, 40, 45.
- [69] Been Kim, Elena Glassman, Brittney Johnson, and Julie Shah. iBCM: Interactive Bayesian case model empowering humans via intuitive interaction. *MIT Libraries Technical Report: MIT-CSAIL-TR-2015-010*, 2015. Cited on page(s): 7, 127, 158, 164, 182.
- [70] Been Kim, Rajiv Khanna, and Oluwasanmi O Koyejo. Examples are not enough, learn to criticize! Criticism for interpretability. In *Advances in Neural Information Processing Systems*, pages 2280–2288, 2016. Cited on page(s): 7, 34, 40, 45, 58.
- [71] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, and Rory Sayres. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV). In *International conference on machine learning*, pages 2668–2677, 2018. Cited on page(s): xxv, 21, 121.
- [72] Alexandra Kirsch. Explain to whom? Putting the user in the center of explainable AI. In *Proceedings of the First International Workshop on Comprehensibility and Explanation in AI and ML 2017 colocated with 16<sup>th</sup> International Conference of the Italian Association for Artificial Intelligence (AI\*IA 2017), Bari, Italy*, 2017. Cited on page(s): 2.
- [73] Derek J Koehler. Explanation, imagination, and confidence in judgment. *Psychological bulletin*, 110(3):499, 1991. Cited on page(s): 9, 59.
- [74] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34<sup>th</sup> International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1885–1894, International Convention Centre, Sydney, Australia, 06–11 Aug 2017.

- PMLR. URL <http://proceedings.mlr.press/v70/koh17a.html>.  
Cited on page(s): 7, 20, 40, 43.
- [75] Aaron Kramer, Pramit Choudhary, silversurfer84, Ben van Dyke, Alvin Thai, Nitin Pasumorthy, Guillaume Lemaitre, Dave Thompson, and Ben Cook. Skater v1.1.2, September 2018. URL <https://doi.org/10.5281/zenodo.1423046>.  
Cited on page(s): 234.
- [76] Josua Krause, Adam Perer, and Enrico Bertini. Using visual analytics to interpret predictive machine learning models. *Workshop on Human Interpretability in Machine Learning (WHI 2016) at the 33<sup>rd</sup> International Conference on Machine Learning (ICML 2016), New York, New York, 2016*. URL <https://arxiv.org/abs/1606.05685>. arXiv preprint arXiv:1606.05685.  
Cited on page(s): 6, 7, 40, 42, 43, 158, 182.
- [77] Josua Krause, Adam Perer, and Kenney Ng. Interacting with predictions: Visual inspection of black-box machine learning models. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 5686–5697. ACM, 2016.  
Cited on page(s): 7, 21, 45, 58, 126, 127, 158, 164, 182.
- [78] Todd Kulesza, Simone Stumpf, Margaret Burnett, Weng-Keen Wong, Yann Riche, Travis Moore, Ian Oberst, Amber Shinsel, and Kevin McIntosh. Explanatory debugging: Supporting end-user debugging of machine-learned programs. In *2010 IEEE Symposium on Visual Languages and Human-Centric Computing*, pages 41–48. IEEE, 2010.  
Cited on page(s): 126, 182.
- [79] Todd Kulesza, Simone Stumpf, Margaret Burnett, and Irwin Kwan. Tell me more?: The effects of mental model soundness on personalizing an intelligent agent. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1–10. ACM, 2012.  
Cited on page(s): 46, 171.
- [80] Todd Kulesza, Simone Stumpf, Margaret Burnett, Sherry Yang, Irwin Kwan, and Weng-Keen Wong. Too much, too little, or just right? Ways explanations impact end users’ mental models. In *Visual Languages and Human-Centric Computing (VL/HCC), 2013 IEEE Symposium on*, pages 3–10. IEEE, 2013.  
Cited on page(s): 2, 10, 34, 35, 44, 45, 46, 53, 58, 113, 182.
- [81] Todd Kulesza, Margaret Burnett, Weng-Keen Wong, and Simone Stumpf. Principles of explanatory debugging to personalize interactive machine learning. In *Proceedings of the 20<sup>th</sup> International Conference on Intelligent User Interfaces*, pages 126–137. ACM, 2015.  
Cited on page(s): 13, 21, 34, 35, 41, 44, 45, 47, 58, 127, 158, 163, 164, 169, 182.

- [82] Meelis Kull, Miquel Perello Nieto, Markus Kängsepp, Telmo Silva Filho, Hao Song, and Peter Flach. Beyond temperature scaling: Obtaining well-calibrated multi-class probabilities with Dirichlet calibration. In *Advances in Neural Information Processing Systems*, pages 12316–12326, 2019.  
Cited on page(s): 94, 137.
- [83] Matt J Kusner, Joshua Loftus, Chris Russell, and Ricardo Silva. Counterfactual fairness. In *Advances in Neural Information Processing Systems*, pages 4066–4076, 2017.  
Cited on page(s): 17, 119, 122, 126, 163, 184.
- [84] Himabindu Lakkaraju and Osbert Bastani. “How do I fool you?” Manipulating user trust via misleading black box explanations. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 79–85, 2020.  
Cited on page(s): 63, 64, 66, 79, 101.
- [85] Himabindu Lakkaraju, Stephen H Bach, and Jure Leskovec. Interpretable decision sets: A joint framework for description and prediction. In *Proceedings of the 22<sup>nd</sup> ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1675–1684, 2016.  
Cited on page(s): xxiii, 19.
- [86] Himabindu Lakkaraju, Ece Kamar, Rich Caruana, and Jure Leskovec. Interpretable & explorable approximations of black box models. *4<sup>th</sup> Workshop on Fairness, Accountability, and Transparency in Machine Learning (FAT/ML 2017) at the 23<sup>rd</sup> SIGKDD conference on Knowledge Discovery and Data Mining (KDD 2017), Halifax, Nova Scotia, Canada*, 2017. URL <https://arxiv.org/abs/1707.01154>. arXiv preprint arXiv:1707.01154.  
Cited on page(s): xxiii, 19, 58.
- [87] Himabindu Lakkaraju, Ece Kamar, Rich Caruana, and Jure Leskovec. Faithful and customizable explanations of black box models. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*. ACM, 2019.  
Cited on page(s): 63, 66, 79, 101, 184, 185.
- [88] Thibault Laugel, Marie-Jeanne Lesot, Christophe Marsala, Xavier Renard, and Marcin Detyniecki. Comparison-based inverse classification for interpretability in machine learning. In *Information Processing and Management of Uncertainty in Knowledge-Based Systems. Theory and Foundations*, pages 100–111. Springer International Publishing, 2018. ISBN 978-3-319-91473-2.  
Cited on page(s): 90.
- [89] Thibault Laugel, Xavier Renard, Marie-Jeanne Lesot, Christophe Marsala, and Marcin Detyniecki. Defining locality for surrogates in post-hoc interpretability. *3<sup>rd</sup> Workshop on Human Interpretability in Machine Learning (WHI 2018) at the 35<sup>th</sup> International*

- Conference on Machine Learning (ICML 2018), Stockholm, Sweden*, 2018. URL <https://arxiv.org/abs/1806.07498>. arXiv preprint arXiv:1806.07498.  
Cited on page(s): 54, 63, 64, 66, 79, 92, 101, 135.
- [90] Thibault Laugel, Marie-Jeanne Lesot, Christophe Marsala, Xavier Renard, and Marcin Detyniecki. The dangers of post-hoc interpretability: Unjustified counterfactual explanations. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 2801–2807. International Joint Conferences on Artificial Intelligence Organization, 7 2019. doi: 10.24963/ijcai.2019/388. URL <https://doi.org/10.24963/ijcai.2019/388>.  
Cited on page(s): 50.
- [91] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015.  
Cited on page(s): 20, 121.
- [92] Cornelius T Leondes. *Expert systems: The technology of knowledge management and decision making for the 21<sup>st</sup> century*. Elsevier, 2001.  
Cited on page(s): 8.
- [93] David Lewis. Causation. *The journal of philosophy*, 70(17):556–567, 1973.  
Cited on page(s): 122.
- [94] Brian Y Lim, Anind K Dey, and Daniel Avrahami. Why and why not explanations improve the intelligibility of context-aware intelligent systems. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 2119–2128, 2009.  
Cited on page(s): 107.
- [95] Zachary C Lipton. The doctor just won’t accept that! *Interpretable ML Symposium, 31<sup>st</sup> Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA*, 2017. URL <http://arxiv.org/abs/1711.08037>. arXiv preprint arXiv:1711.08037.  
Cited on page(s): 163.
- [96] Zachary C. Lipton. The mythos of model interpretability. *Communications of the ACM*, 16(3):30:31–30:57, June 2018. ISSN 1542-7730. doi: 10.1145/3236386.3241340. URL <http://doi.acm.org/10.1145/3236386.3241340>.  
Cited on page(s): 6, 34, 41, 42, 52, 58, 106, 189.
- [97] Tania Lombrozo. The structure and function of explanations. *Trends in cognitive sciences*, 10(10):464–470, 2006.  
Cited on page(s): 9.

- [98] Tania Lombrozo. Simplicity and probability in causal explanation. *Cognitive psychology*, 55(3):232–257, 2007.  
Cited on page(s): 44, 46, 47, 53.
- [99] Yin Lou, Rich Caruana, Johannes Gehrke, and Giles Hooker. Accurate intelligible models with pairwise interactions. In *Proceedings of the 19<sup>th</sup> ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 623–631. ACM, 2013.  
Cited on page(s): 21.
- [100] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4765–4774. Curran Associates, Inc., 2017.  
Cited on page(s): xxv, 5, 7, 19, 22, 66, 100, 121, 164, 235.
- [101] Prashan Madumal, Tim Miller, Liz Sonenberg, and Frank Vetere. A grounded interaction protocol for explainable artificial intelligence. In *Proceedings of the 18<sup>th</sup> International Conference on Autonomous Agents and MultiAgent Systems*, pages 1033–1041. International Foundation for Autonomous Agents and Multiagent Systems, 2019.  
Cited on page(s): 9, 23, 127, 158, 165, 177, 180, 183.
- [102] Christopher Manning and Hinrich Schutze. *Foundations of statistical natural language processing*. MIT press, 1999.  
Cited on page(s): 68.
- [103] Fernando Martínez-Plumed, Lidia Contreras-Ochando, Cèsar Ferri, José Hernández Orallo, Meelis Kull, Nicolas Lachiche, Maréa José Ramírez Quintana, and Peter A Flach. CRISP-DM twenty years later: From data mining processes to data science trajectories. *IEEE Transactions on Knowledge and Data Engineering*, 2019.  
Cited on page(s): xxiii, 22, 62.
- [104] R Mead. When a teacher’s job depends on a child’s test. *The New Yorker*, 2015.  
Cited on page(s): 9.
- [105] Tim Miller. “but why?” Understanding explainable artificial intelligence. *XRDS: Crossroads, The ACM Magazine for Students*, 25(3):20–25, 2019.  
Cited on page(s): 16.
- [106] Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267:1–38, 2019.  
Cited on page(s): 3, 11, 12, 14, 23, 30, 34, 35, 40, 45, 46, 47, 53, 58, 60, 97, 106, 108, 109, 113, 114, 119, 122, 126, 127, 146, 148, 157, 158, 162, 164, 165, 166, 169, 172, 183, 184.

- [107] Tim Miller, Piers Howe, and Liz Sonenberg. Explainable AI: Beware of inmates running the asylum or: How I learnt to stop worrying and love the social and behavioural sciences. In *Proceedings of the IJCAI Workshop on eXplainable Artificial Intelligence (XAI 2017), Melbourne, Australia*, 2017. URL <https://arxiv.org/abs/1712.00547>. arXiv preprint arXiv:1712.00547.  
Cited on page(s): 11, 44, 46, 47, 58, 119, 158, 164, 183.
- [108] Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, and Timnit Gebru. Model cards for model reporting. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 220–229. ACM, 2019.  
Cited on page(s): 39, 56, 59.
- [109] Stephen Muggleton and Luc De Raedt. Inductive logic programming: Theory and methods. *The Journal of Logic Programming*, 19:629–679, 1994.  
Cited on page(s): 119.
- [110] Stephen Muggleton, José Santos, and Alireza Tamaddon-Nezhad. ProGolem: A system based on relative minimal generalisation. In *International Conference on Inductive Logic Programming*, pages 131–148. Springer, 2009.  
Cited on page(s): 119.
- [111] Harsha Nori, Samuel Jenkins, Paul Koch, and Rich Caruana. InterpretML: A unified framework for machine learning interpretability. *arXiv preprint arXiv:1909.09223*, 2019. URL <http://arxiv.org/abs/1909.09223>.  
Cited on page(s): 234.
- [112] Fabian Offert. “I know it when I see it”. Visualization and intuitive interpretability. *2017 Symposium on Interpretable Machine Learning at the 31<sup>st</sup> Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, California*, 2017. URL <http://arxiv.org/abs/1711.08042>. arXiv preprint arXiv:1711.08042.  
Cited on page(s): 14, 53.
- [113] Travis E. Oliphant. *A guide to NumPy*, volume 1. Trelgol Publishing USA, 2006.  
Cited on page(s): 230.
- [114] Alexandra Olteanu, Carlos Castillo, Fernando Diaz, and Emre Kiciman. Social data: Biases, methodological pitfalls, and ethical boundaries. *Frontiers in Big Data*, 2:13, 2019.  
Cited on page(s): 16.
- [115] The IEEE Global Initiative on Ethics of Autonomous and Intelligent Systems. Ethically aligned design: A vision for prioritizing human well-being with autonomous and intelligent

systems, 2017. URL [http://standards.ieee.org/develop/indconn/ec/autonomous\\_systems.html](http://standards.ieee.org/develop/indconn/ec/autonomous_systems.html).

Cited on page(s): 55, 58, 59.

- [116] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8026–8037. Curran Associates, Inc., 2019. URL <http://papers.nips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>.

Cited on page(s): xxiv, 153, 230, 238.

- [117] Judea Pearl. *Causality: Models, reasoning and inference*. Cambridge University Press, 2000.

Cited on page(s): 122.

- [118] Judea Pearl and Dana Mackenzie. *The book of why: The new science of cause and effect*. Basic Books, 2018.

Cited on page(s): 9, 40, 43.

- [119] Judea Pearl, Madelyn Glymour, and Nicholas P Jewell. *Causal inference in statistics: A primer*. John Wiley & Sons, 2016.

Cited on page(s): 15, 49.

- [120] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Edouard Duchesnay. scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Cited on page(s): xxv, 131, 139, 167, 230.

- [121] Miquel Perello-Nieto, E Silva Telmo De Menezes Filho, Meelis Kull, and Peter Flach. Background check: A general technique to build more reliable and versatile classifiers. In *2016 IEEE 16<sup>th</sup> International Conference on Data Mining (ICDM)*, pages 1143–1148. IEEE, 2016.

Cited on page(s): 50, 233.

- [122] François Portet, Ehud Reiter, Albert Gatt, Jim Hunter, Somayajulu Sripada, Yvonne Freer, and Cindy Sykes. Automatic generation of textual summaries from neonatal intensive care data. *Artificial Intelligence*, 173(7):789–816, 2009.  
Cited on page(s): 8.
- [123] Rafael Poyiadzi, Kacper Sokol, Raul Santos-Rodriguez, Tijl De Bie, and Peter Flach. FACE: Feasible and actionable counterfactual explanations. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 344–350, 2020.  
Cited on page(s): xxiii, 7, 15, 50, 91, 109, 119, 120, 122, 184, 185, 233.
- [124] Alun Preece, Dan Harborne, Dave Braines, Richard Tomsett, and Supriyo Chakraborty. Stakeholders in explainable AI. *Proceedings of the AAAI Fall Symposium on Artificial Intelligence in Government and Public Sector, Arlington, Virginia, USA*, 2018. arXiv preprint arXiv:1810.00184.  
Cited on page(s): 2, 42, 50, 58.
- [125] W Andrew Pruett and Robert L Hester. The creation of surrogate models for fast estimation of complex model outcomes. *PloS one*, 11(6), 2016.  
Cited on page(s): 100.
- [126] Dillon Reisman, Jason Schultz, Kate Crawford, and Meredith Whittaker. Algorithmic impact assessments: A practical framework for public agency accountability. *AI Now Institute*, 2018.  
Cited on page(s): 56, 59.
- [127] People + AI Research (PAIR) initiative at Google. People + ai guidebook: Designing human-centered ai products, 2019. URL <https://pair.withgoogle.com>.  
Cited on page(s): 58.
- [128] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Model-agnostic interpretability of machine learning. *1<sup>st</sup> Workshop on Human Interpretability in Machine Learning (WHI 2016) at the 33<sup>rd</sup> International Conference on Machine Learning (ICML 2016), New York, NY*, 2016. URL <https://arxiv.org/abs/1606.05386>. arXiv preprint arXiv:1606.05386.  
Cited on page(s): 85, 121.
- [129] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. “why should I trust you?”: Explaining the predictions of any classifier. In *Proceedings of the 22<sup>nd</sup> ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 1135–1144, 2016.  
Cited on page(s): xxiv, 6, 7, 19, 22, 25, 30, 35, 41, 54, 60, 63, 66, 67, 68, 69, 72, 84, 90, 93, 95, 100, 109, 121, 126, 128, 129, 131, 134, 135, 141, 142, 150, 155, 158, 159, 162, 163, 164, 195, 234, 242, 244, 245.

- [130] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Anchors: High-precision model-agnostic explanations. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. Cited on page(s): xxiii, 19, 22, 109, 119, 121, 164, 235.
- [131] Marko Robnik-Šikonja and Marko Bohanec. Perturbation-based explanations of prediction models. In *Human and Machine Learning*, pages 159–175. Springer, 2018. Cited on page(s): 39, 43, 162.
- [132] Leonid Rozenblit and Frank Keil. The misunderstood limits of folk science: An illusion of explanatory depth. *Cognitive science*, 26(5):521–562, 2002. Cited on page(s): 9, 15, 51, 59.
- [133] Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, 2019. Cited on page(s): 1, 3, 4, 5, 8, 9, 22, 43, 62, 63, 66, 79, 81, 98, 109, 121, 135, 140, 162, 177.
- [134] Pierangela Samarati and Latanya Sweeney. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. Technical report, Technical Report, SRI International, 1998. Cited on page(s): 48.
- [135] John G Saxe. *The blind men and the elephant*. Enrich Spot Limited, 2016. Cited on page(s): 16.
- [136] Nripsuta Ani Saxena. Perceptions of fairness. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 537–538, 2019. Cited on page(s): 16.
- [137] Nripsuta Ani Saxena, Karen Huang, Evan DeFilippis, Goran Radanovic, David C Parkes, and Yang Liu. How do fairness definitions fare? Examining public attitudes towards algorithmic definitions of fairness. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 99–106, 2019. Cited on page(s): 16.
- [138] Johanes Schneider and Joshua Handali. Personalized explanation for machine learning: A conceptualization. In *ECIS*, 2019. Cited on page(s): 42, 45, 47, 58, 127, 158, 164, 165, 169, 183, 184.
- [139] John R Searle. Minds, brains, and programs. *Behavioral and brain sciences*, 3(3):417–424, 1980. Cited on page(s): 2, 6, 10, 52, 106, 189.

- [140] Lloyd S Shapley. A value for n-person games. *Contributions to the Theory of Games*, 2(28): 307–317, 1953.  
Cited on page(s): 20.
- [141] Colin Shearer. The CRISP-DM model: The new blueprint for data mining. *Journal of data warehousing*, 5(4):13–22, 2000.  
Cited on page(s): 62.
- [142] Sheng Shi, Xinfeng Zhang, Haisheng Li, and Wei Fan. Explaining the predictions of any image classifier via decision trees. *arXiv preprint arXiv:1911.01058*, 2019. URL <http://arxiv.org/abs/1911.01058>.  
Cited on page(s): 108, 126, 158.
- [143] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017.  
Cited on page(s): 3.
- [144] Tom Simonite. Google’s ai guru wants computers to think more like brains, 2019.  
Cited on page(s): 15.
- [145] Daniel J Simons. Current approaches to change blindness. *Visual cognition*, 7(1-3):1–15, 2000.  
Cited on page(s): 51.
- [146] Daniel Smilkov, Nikhil Thorat, Charles Nicholson, Emily Reif, Fernanda B Viégas, and Martin Wattenberg. Embedding projector: Interactive visualization and interpretation of embeddings. *Workshop on Interpretable Machine Learning in Complex Systems at the 30<sup>th</sup> Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain*, 2016. URL <https://arxiv.org/abs/1611.05469>. arXiv preprint arXiv:1611.05469.  
Cited on page(s): 6.
- [147] Kacper Sokol. Fairness, accountability and transparency in artificial intelligence: A case study of logical predictive models. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 541–542, 2019.  
Cited on page(s): 15, 25, 120, 122.
- [148] Kacper Sokol and Peter Flach. Desiderata for interpretability: Explaining decision tree predictions with counterfactuals. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 10035–10036, 2019.  
Cited on page(s): 25.

- [149] Kacper Sokol and Peter Flach. Explainability fact sheets: A framework for systematic assessment of explainable approaches. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 56–67, 2020.  
Cited on page(s): 25, 80, 81, 121, 195.
- [150] Kacper Sokol and Peter Flach. LIMETree: Interactively customisable explanations based on local surrogate multi-output regression trees. *arXiv preprint arXiv:2005.01427*, 2020. URL <https://arxiv.org/abs/2005.01427>.  
Cited on page(s): 26, 68, 95, 108.
- [151] Kacper Sokol and Peter Flach. One explanation does not fit all. *KI-Künstliche Intelligenz*, pages 1–16, 2020.  
Cited on page(s): 12, 13, 22, 23, 27, 68, 82, 119, 121, 127, 146, 147, 148, 158.
- [152] Kacper Sokol and Peter Flach. Towards faithful and meaningful interpretable representations. *arXiv preprint arXiv:2008.07007*, 2020. URL <http://arxiv.org/abs/2008.07007>.  
Cited on page(s): 26, 89, 108.
- [153] Kacper Sokol and Peter A Flach. The role of textualisation and argumentation in understanding the machine learning process. In *IJCAI*, pages 5211–5212, 2017.  
Cited on page(s): 6, 8, 27.
- [154] Kacper Sokol and Peter A Flach. The role of textualisation and argumentation in understanding the machine learning process: A position paper. In *Automated Reasoning Workshop*, pages 11–12, 2017.  
Cited on page(s): 6, 8, 27.
- [155] Kacper Sokol and Peter A Flach. Conversational explanations of machine learning predictions through class-contrastive counterfactual statements. In *IJCAI*, pages 5785–5786, 2018.  
Cited on page(s): 27, 119.
- [156] Kacper Sokol and Peter A Flach. Glass-Box: Explaining AI decisions with counterfactual statements through conversation with a voice-enabled virtual assistant. In *IJCAI*, pages 5868–5870, 2018.  
Cited on page(s): 27, 119, 121, 127, 148, 152, 158, 159, 166.
- [157] Kacper Sokol and Peter A Flach. Counterfactual explanations of machine learning predictions: Opportunities and challenges for AI safety. In *Proceedings of the SafeAI Workshop at the AAAI Conference on Artificial Intelligence*, 2019.  
Cited on page(s): 25, 48, 119, 121, 153, 163.

- [158] Kacper Sokol, Alexander Hepburn, Raul Santos-Rodriguez, and Peter Flach. bLIMEy: Surrogate prediction explanations beyond LIME. *2019 Workshop on Human-Centric Machine Learning (HCML 2019) at the 33<sup>rd</sup> Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada*, 2019. URL <https://arxiv.org/abs/1910.13016>. arXiv preprint arXiv:1910.13016.  
Cited on page(s): 5, 22, 26, 54, 63, 80, 84, 89, 90, 95, 96, 108, 126, 128, 130, 135, 150, 151, 153, 158, 229, 232, 233, 238.
- [159] Kacper Sokol, Raul Santos-Rodriguez, and Peter Flach. FAT Forensics: A Python toolbox for algorithmic fairness, accountability and transparency. *arXiv preprint arXiv:1909.05167*, 2019. URL <https://arxiv.org/abs/1909.05167>.  
Cited on page(s): 26, 66, 102, 153, 228, 238.
- [160] Kacper Sokol, Alexander Hepburn, Rafael Poyiadzi, Matthew Clifford, Raul Santos-Rodriguez, and Peter Flach. FAT Forensics: A Python toolbox for implementing and deploying fairness, accountability and transparency algorithms in predictive systems. *Journal of Open Source Software*, 5(49):1904, 2020.  
Cited on page(s): xvii, 26, 66, 91, 102, 153, 228, 238.
- [161] Kacper Sokol, Alexander Hepburn, Raul Santos-Rodriguez, and Peter Flach. What and how of machine learning transparency: Building bespoke explainability tools with interoperable algorithmic components. *Hands-on Tutorial at The European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD), Ghent, Belgium*, 2020. URL [https://events.fat-forensics.org/2020\\_ecml-pkdd](https://events.fat-forensics.org/2020_ecml-pkdd).  
Cited on page(s): 26.
- [162] Kacper Sokol, Alexander Hepburn, Raul Santos-Rodriguez, and Peter Flach. What and how of machine learning transparency: Building bespoke explainability tools with interoperable algorithmic components, September 2020. URL <https://doi.org/10.5281/zenodo.4035128>.  
Cited on page(s): 66.
- [163] Sören Sonnenburg, Mikio L. Braun, Cheng Soon Ong, Samy Bengio, Leon Bottou, Geoffrey Holmes, Yann LeCun, Klaus-Robert Müller, Fernando Pereira, Carl Edward Rasmussen, Gunnar Rätsch, Bernhard Schölkopf, Alexander Smola, Pascal Vincent, Jason Weston, and Robert Williamson. The need for open source software in machine learning. *Journal of Machine Learning Research*, 8(Oct):2443–2466, 2007.  
Cited on page(s): 227.
- [164] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE*

- conference on computer vision and pattern recognition*, pages 2818–2826, 2016.  
Cited on page(s): 132, 153.
- [165] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.  
Cited on page(s): xxiv.
- [166] Gabriele Tolomei, Fabrizio Silvestri, Andrew Haines, and Mounia Lalmas. Interpretable predictions of tree-based ensembles via actionable feature tweaking. In *Proceedings of the 23<sup>rd</sup> ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 465–474. ACM, 2017.  
Cited on page(s): 45, 109, 120, 122, 158.
- [167] Richard Tomsett, Dave Braines, Dan Harborne, Alun Preece, and Supriyo Chakraborty. Interpretable to whom? A role-based model for analyzing interpretable machine learning systems. *3<sup>rd</sup> Workshop on Human Interpretability in Machine Learning (WHI 2018) at the 35<sup>th</sup> International Conference on Machine Learning (ICML 2018), Stockholm, Sweden*, 2018. URL <https://arxiv.org/abs/1806.07552>. arXiv preprint arXiv:1806.07552.  
Cited on page(s): 42, 45, 47, 50, 58.
- [168] Florian Tramer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, Jean-Pierre Hubaux, Mathias Humbert, Ari Juels, and Huang Lin. FairTest: Discovering unwarranted associations in data-driven applications. In *2017 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 401–416. IEEE, 2017.  
Cited on page(s): 234.
- [169] Jasper van der Waa, Marcel Robeer, Jurriaan van Diggelen, Matthieu Brinkhuis, and Mark Neerinx. Contrastive explanations with local foil trees. *Workshop on Human Interpretability in Machine Learning (WHI 2018) at the 35<sup>th</sup> International Conference on Machine Learning (ICML 2018), Stockholm, Sweden*, 2018. URL <https://arxiv.org/abs/1806.07470>. arXiv preprint arXiv:1806.07470.  
Cited on page(s): 21, 76, 95, 97, 109, 119, 122, 126, 127, 158, 180, 183, 184.
- [170] Stéfan van der Walt, Johannes L. Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D. Warner, Neil Yager, Emmanuelle Gouillart, Tony Yu, and the scikit-image contributors. scikit-image: Image processing in Python. *PeerJ*, 2:e453, 6 2014. ISSN 2167-8359. doi: 10.7717/peerj.453. URL <https://doi.org/10.7717/peerj.453>.  
Cited on page(s): xxiv, 130, 238.
- [171] Andrea Vedaldi and Stefano Soatto. Quick shift and kernel methods for mode seeking. In *European conference on computer vision*, pages 705–718. Springer, 2008.  
Cited on page(s): 68, 78, 130.

- [172] Sandra Wachter, Brent Mittelstadt, and Luciano Floridi. Why a right to explanation of automated decision-making does not exist in the general data protection regulation. *International Data Privacy Law*, 7(2):76–99, 2017.  
Cited on page(s): 35, 106, 184.
- [173] Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the GDPR. *Harv. JL & Tech.*, 31: 841, 2017.  
Cited on page(s): 7, 11, 17, 21, 23, 30, 34, 35, 106, 109, 119, 120, 122, 127, 148, 158, 164, 166, 183, 184, 185.
- [174] Douglas Walton. Dialogical models of explanation. *ExaCt*, 2007:1–9, 2007.  
Cited on page(s): 9, 42, 46, 47, 53, 127, 158, 183.
- [175] Douglas Walton. A dialogue system specification for explanation. *Synthese*, 182(3):349–374, 2011.  
Cited on page(s): 9, 180, 183.
- [176] Douglas Walton. A dialogue system for evaluating explanations. In *Argument Evaluation and Evidence*, pages 69–116. Springer, 2016.  
Cited on page(s): 9, 180, 183.
- [177] Fulton Wang and Cynthia Rudin. Falling rule lists. In *Artificial Intelligence and Statistics*, pages 1013–1022, 2015.  
Cited on page(s): 21.
- [178] Daniel S Weld and Gagan Bansal. The challenge of crafting intelligible intelligence. *Communications of the ACM*, 62(6):70–79, 2019.  
Cited on page(s): 7, 21, 35, 45, 47, 56, 58, 126, 127, 158, 164, 165, 169, 182, 183.
- [179] James Wexler. Facets: An open source visualization tool for machine learning training data. *Google Open Source Blog*, 2017.  
Cited on page(s): 21.
- [180] Ke Yang, Julia Stoyanovich, Abolfazl Asudeh, Bill Howe, HV Jagadish, and Gerome Miklau. A nutritional label for rankings. In *Proceedings of the 2018 International Conference on Management of Data*, pages 1773–1776. ACM, 2018.  
Cited on page(s): 55, 56, 59.
- [181] Mengjiao Yang and Been Kim. Benchmark attribution methods with ground truth. *2019 Workshop on Human-Centric Machine Learning (HCML 2019) at the 33<sup>rd</sup> Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada*, 2019. URL

<https://drive.google.com/file/d/1w1P0UB3bBVZ82g60blxM6mh6C3nxNyh/view>.

Cited on page(s): xxiii, 69.

- [182] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. MixUp: Beyond empirical risk minimization. *Proceedings of the 6<sup>th</sup> International Conference on Learning Representations (ICLR), Vancouver, Canada*, 2018.

Cited on page(s): xxiv, 90.

- [183] Yujia Zhang, Kuangyan Song, Yiming Sun, Sarah Tan, and Madeleine Udell. “Why should you trust my explanation?” Understanding uncertainty in LIME explanations. *AI for Social Good Workshop at the 36<sup>th</sup> International Conference on Machine Learning (ICML 2019), Long Beach, California*, 2019. URL <https://arxiv.org/abs/1904.12991>. arXiv preprint arXiv:1904.12991.

Cited on page(s): 5, 54, 63, 64, 66, 79, 89, 101, 135.

- [184] Luisa M Zintgraf, Taco S Cohen, Tameem Adel, and Max Welling. Visualizing deep neural network decisions: Prediction difference analysis. *Proceedings of the 5<sup>th</sup> International Conference on Learning Representations (ICLR), Toulon, France*, 2017. URL <https://openreview.net/forum?id=BJ5UeU9xx>. arXiv preprint arXiv:1702.04595.

Cited on page(s): 20.



## INDEX

- accountability, 17
- argumentation, 8
- audience, 12
- bias
  - confirmation bias, 50
  - outcome bias, 50
  - selection bias, 50
- black box, 1
- bLIMEy, 64, 78
  - building blocks, *see* bLIMEy, modules
  - data sampling, 65, 75, 90
  - evaluation, 98
  - explanation generation, 65, 76, 92
    - decision trees, 97
    - feature selection, 93
    - ordinary least squares, 96, 244
    - fitting surrogate model, 93
    - weighting, 92
  - framework, 75
  - interpretable data representation, 65, 66, 75, 80, *see also* interpretable data representation
  - determinism, 88
  - faithfulness, 82
  - information loss, 85
  - information removal, 85
  - proxy, 85
  - meta-algorithm, 75
  - modules, 75, 78
  - compatibility**, 95
  - data sampling, *see* bLIMEy, data sampling
  - explanation generation, *see* bLIMEy, explanation generation
  - interpretable data representation, *see* bLIMEy, interpretable data representation
- The Blind Men and the Elephant, 16
- cause, 8
- completeness, 2
- comprehensibility, 8
- contrastive explanations, 14
- counterfactuals, 14, 106
- CtreeX, 108
- data sampling, *see* bLIMEy, data sampling
- decision tree, 105
  - explainability overview, 121
  - explanation types, 107, 109, 113
    - contrastive explanations, 118
    - exemplars, 112
    - feature importance, 110
    - logical rules, 111
    - supportive explanations, 118
    - tree structure visualisation, 109
    - what-ifs, 112
  - meta-feature representation, 114
- decomposability, 52

- determinism, *see* bLIMEy, interpretable
  - data representation, determinism
- elephant, *see* Blind Men and the Elephant
- evaluation, 51
  - application-level, 51
  - function-level, 51
  - human-level, 51
- evidence, 8
- explainability, 8
  - benefits, 15
    - accountability, 17
    - fairness, 16
    - interpretability, 16
  - data, 6
  - definition, 9
  - equation, 10
  - literature overview, 18
  - models, 6
  - predictions, 7
  - process, 13
  - taxonomy, *see* taxonomy
- explanation generation, *see* bLIMEy, explanation generation
- explanatory process, 13
- explicability, 8
- fact sheet, 35
  - examples, 195
    - CtreeX, 206
    - LIME, 196
    - LIMEtree, 215
- fairness, 16
- faithfulness, *see* fidelity
- FAT Forensics, 227
  - accountability, 233
  - architecture, 229
  - documentation, 231
  - fairness, 232
  - implementations, 232
  - modes, 231
    - deployment, 231
    - research, 231
    - transparency, 233
- fidelity
  - completeness, 44
  - soundness, 44
- full fidelity, *see* LIMEtree, fidelity, full fidelity
- functional requirements, *see* taxonomy, functional requirements
- Glass-Box, 166
  - deployment, 176
  - feedback, 175
  - interaction, 168
  - lessons learnt, 176
  - properties, 173
  - reception, 175
- human aspects, 11
- The Illusion of Explanatory Depth, 50
- intelligibility, 8
- interactive explainability, 162
  - desiderata, 169
  - literature overview, 181
    - explanatory process, 183
    - Human–Computer Interaction, 181
    - interactive explainers, 182
    - interdisciplinary perspective, 183
    - social sciences, 183
- interpretability, 8
- interpretable data representation, *see* bLIMEy, interpretable data representation
- image data, 68
  - colour uniformity, 133
  - edges preservation, 134

- mean-colour occlusion, 133
  - occlusion colour, 239
  - segmentation granularity, 133
  - tabular data, 69
    - quartile-based, 242
    - segmentation granularity, 239
    - tree-based, 241, 242
  - text data, 67
- justification, 8
- LIME, 63, 71
- algorithm, 73
  - loss function, 72, 131
  - objective function, 72, 131
  - trade-offs, 132
    - fidelity issues, 135
    - impossibility of information removal, 135
    - linear model assumptions, 132
    - mean-colour occlusion, 133
    - one class limitation, 132
- LIMEtree, 127, 138
- advantages, 145
    - interactive explainability, *see* LIMEtree, personalisation
    - personalised explainability, *see* LIMEtree, personalisation
  - algorithm, 140
  - example, 128, 142
  - explanation complexity, 152
  - explanation complexity function, 139
  - fidelity, 140
    - data-driven, 142
    - full fidelity, 141, 142, 151
    - model-driven, 141, 142
  - loss function, 139
  - minimal representation, 140
  - personalisation, 146
    - explanation content, 148
    - explanation type, 147
    - interpretable representation, 146
  - tabular data, 149
  - text data, 149
  - validation
    - synthetic, 153
    - user study, 155
- machine learning process, 5
- mental model
  - functional, 2
  - structural, 2
- multi-output regression, 136
- observability, 8
- operational requirements, *see* taxonomy, operational requirements
- ordinary least squares, *see* bLIMEy, explanation generation, ordinary least squares
- proxy, *see* bLIMEy, interpretable data representation, proxy
- reason, 8
- safety requirements, *see* taxonomy, safety requirements
- simulatability, 6, 8, 52
- soundness, 2
- summarisation (statistical), 8
- supportive explanation, 106
- surrogates, 62, *see also* bLIMEy
  - images, 129
  - literature overview, 100
- synthetic experiments, *see* taxonomy, validation requirements, synthetic experiments
- taxonomy, 35

- delivery, 55
- functional requirements, 37
  - F6** applicable model class, 39
  - F9** caveats and assumptions, 39
  - F8** compatible feature types, 39
  - F5** computational complexity, 38
  - F4** explanation breadth/scope, 38
  - F3** explanation target, 38
  - F1** problem supervision level, 37
  - F2** problem type, 38
  - F7** relation to the predictive system, 39
- operational requirements, 40
  - O8** causality vs. actionability, 42
  - O5** data and model transparency, 41
  - O6** explanation audience, 42
  - O4** explanation domain, 41
  - O1** explanation family, 40
  - O2** explanatory medium, 40
  - O7** function of the explanation, 42
  - O10** provenance, 43
  - O3** system interaction, 41
  - O9** trust vs. performance, 43
- operationalisation, 56
- overview, 36
- safety requirements, 48
  - S3** explanation invariance, 49
  - S2** explanation misuse, 49
  - S4** explanation quality, 50
  - S1** information leakage, 48
- target audience, 54
- trade-off, 52
- usability requirements, 44
  - U5** actionability, 45
  - U6** chronology, 46
  - U7** coherence, 46
  - U2** completeness, 44
  - U9** complexity, 47
  - U3** contextfulness, 45
  - U4** interactiveness, 45
  - U8** novelty, 46
  - U11** parsimony, 47
  - U10** personalisation, 47
  - U1** soundness, 44
- validation requirements, 50
  - algorithmic transparency, 52
  - confirmation bias, 50
  - decomposability, 52
  - evaluation, *see* evaluation
  - The Illusion of Explanatory Depth, 50
  - outcome bias, 50
  - selection bias, 50
  - simulatability, 52
  - V2** synthetic experiments, 50
  - V1** user studies, 50
- textualisation, 8
- trade-off
  - LIME, *see* LIME, trade-offs
  - taxonomy, 52
  - transparency–predictive power, 4
- transparency, 8
- transparency–predictive power trade-off, 4
- usability requirements, *see* taxonomy, usability requirements
- user studies, *see* taxonomy, validation requirements, user studies
- validation requirements, *see* taxonomy, validation requirements
- visualisation, 8
- work sheet, 35