Sokol, K., & Flach, P. (2020). LIMEtree: Interactively Customisable Explanations Based on Local Surrogate Multi-output Regression Trees. Unpublished. https://arxiv.org/abs/2005.01427v1

LIMEtree: Interactively Customisable Explanations Based on Local Surrogate Multi-output Regression Trees

Kacper Sokol*, Peter Flach

Department of Computer Science, University of Bristol,
BS8 1UB Bristol, United Kingdom

Abstract

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 capable of outputting a single one-size-fits-all explanation that simply cannot address all of the explainees' diverse needs. In this work we introduce a model-agnostic and post-hoc local explainability technique for black-box predictions called LIMEtree, which 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 (LIME). Our method comes with local fidelity guarantees and can produce a range of diverse explanation types, including contrastive and counterfactual explanations praised in the literature. Some of these explanations can be interactively personalised to create bespoke, meaningful and actionable insights into the model's behaviour. 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 on which an interactive exploratory process can be built.

Keywords: Interactive, Customisable, Model-agnostic, Post-hoc,

*Corresponding author.

Email addresses: K.Sokol@bristol.ac.uk (Kacper Sokol), Peter.Flach@bristol.ac.uk (Peter Flach)

Surrogate, Explainability, Interpretability, Black-box Predictions

1. Introduction

Transparency of predictive systems based on Machine Learning (ML) and Artificial Intelligence (AI) 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 expect to satisfy everyone's needs and expectations. While this might 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* transparency 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 ML and AI 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 (HCI) community [1], the eXplainable Artificial Intelligence (XAI) and Interpretable Machine Learning (IML) communities opt for the second, which, they argue, is the cornerstone of natural and human-like explanations rooted deeply in social sciences [2]. 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 the departure from the one-size-fits-all explanation practices, thereby 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 [3]. 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 [4], e.g., counterfactual cues indicating disparate treatment, or help to debug the underlying black box [5].

Research into AI and ML transparency has recently seen major progress with numerous post-hoc and model-agnostic tools being proposed [6, 7, 8, 9, 10]. Some of these methods can implicitly produce customised explanations achieved by their off-line, non-interactive parametrisation. Work on counterfactual explanations [11, 9] is also quite prominent as they are natural to humans [2] and compliant with various legal regulations [11]. They are also capable of interactive personalisation [12, 13], 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 [14]. Some researchers [15, 16, 17] 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), however 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 [18] or customise and improve [19] the underlying ML algorithm. Interactive explainability systems allowing the user to request different types of static explanations have also been described [1, 3]. All of these techniques are discussed in more details in Section 7.

In our work 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 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 based on their ability to produce diverse explanations:

1. visualisation of the tree structure;
2. tree-based feature importance;
3. logical conditions extracted from a root-to-leaf path;
4. exemplar explanations taken from training data falling into the same leaf;
5. answers to what-if questions generated based on the tree structure; and
6. **counterfactuals** retrieved by comparing and applying logical reasoning to different tree paths.

The first two explanation types uncover the behaviour of a black box in a given predictive sub-space; the other target a specific prediction. While some of these explanations are inherently static, others can be embedded in an **interactive explanatory dialogue**, enabling the explainee to customise and personalise them in a natural way (more details in Section 5.1). We opted for **multi-output** regression trees – depicted in Figure 1 – to avoid common pitfalls associated with surrogate explainers and allow for modelling of multiple classes within the same surrogate model, creating a common source of explanations (see Section 3).

Our method builds upon LIME [7] (Local Interpretable Model-agnostic Explanations) and bLIMEy [8] (build LIME yourself), which are discussed in Section 2.1. LIMEtree addresses many of LIME's shortcomings and limitations (Section 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 *perfect fidelity* with respect to the underlying black-box model under certain conditions. We show the explanatory power of our method with qualitative experiments and quantitative comparison on image classification tasks using a black-box deep neural network (Section 6).

2. Background

The momentum behind surrogate explainability methods can be attributed to their numerous appealing properties, which make them a universal explainability framework. They mimic behaviour of a more complex model either locally [7] or globally [6] with a simpler, inherently interpretable model, thereby providing human-comprehensible insights into its operations. Surrogates are *model-agnostic*, i.e., can be used with any black-box model,

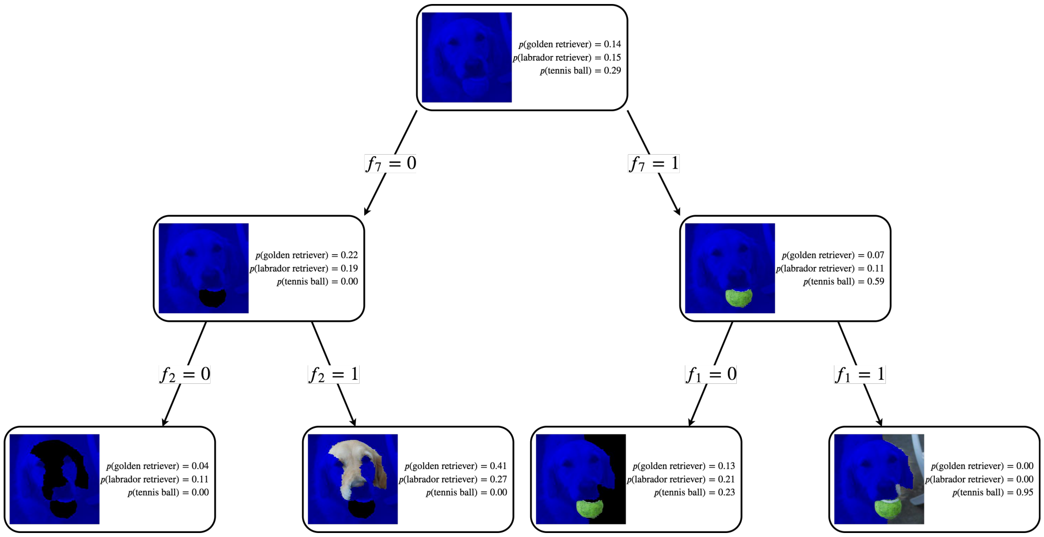
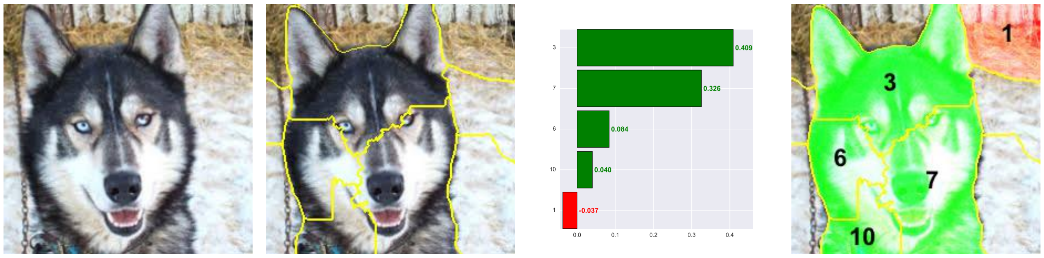

p(golden retriever) = 0.14
p(labrador retriever) = 0.15
p(tennis ball) = 0.29
$f_7 = 0$
$f_7 = 1$
p(golden retriever) = 0.22
p(labrador retriever) = 0.19
p(tennis ball) = 0.00
p(golden retriever) = 0.07
p(labrador retriever) = 0.11
p(tennis ball) = 0.59
$f_2 = 0$
$f_2 = 1$
$f_1 = 0$
$f_1 = 1$
p(golden retriever) = 0.04
p(labrador retriever) = 0.11
p(tennis ball) = 0.00
p(golden retriever) = 0.41
p(labrador retriever) = 0.27
p(tennis ball) = 0.00
p(golden retriever) = 0.13
p(labrador retriever) = 0.21
p(tennis ball) = 0.23
p(golden retriever) = 0.00
p(labrador retriever) = 0.00
p(tennis ball) = 0.95

Figure 1: An example of a multi-output regression tree used to explain an image (presented in Figure 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, hence should not be treated as probabilities.

and *post-hoc*, i.e., can be retrofitted into any existing predictive system regardless of its data domain – tabular, text or image – by using *interpretable representations*. LIME [7] is the most popular surrogate technique geared towards explaining predictions of black-box models (Figure 2).

2.1. Local Surrogates

LIME improves on vanilla surrogate explainers by introducing an *interpretable data representation*. 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 paper we focus on applying surrogate explainers to *image recognition* tasks, which facilitate 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



(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 – super-pixel segmentation.

(c) LIME explanation for *Eskimo dog* – the top five coefficients.

(d) LIME explanation for *Eskimo dog* – the top five segments.

Figure 2: Decomposition of the LIME explanatory process.

popular for images, exhibits properties that are necessary for LIMEtree to achieve perfect 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 3.

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

1. Find the human-interpretable representation $x' \in \mathcal{X}'$ of the data point 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 [8]. In case of image data the interpretable domain $\mathcal{X}'$ is a (super-pixel) segmentation of the image x represented as a binary vector $x' \in \mathcal{X}' = \{0, 1\}^d$, where d is the number of segments. Such a binary vector x' indicates whether a given segment should be preserved (1) or occluded (0), therefore the original image x expressed in the interpretable representation is an all-1 vector $x' = [1, \ldots, 1]$. In practice, this is achieved with an image segmentation technique such as *quick shift* [20] implemented as part of the `scikit-image` Python package[1] [21].
2. Sample n data points uniformly at random from the interpretable representation $\mathcal{X}'$ to get an $n \times d$ binary matrix describing the neighbour-

[1] The `skimage.segmentation.quickshift` function.

hood of the explained image 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^{th} segment if the i^{th} component of a binary vector 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[2] $L : \mathcal{X}' \times \mathcal{X}' \rightarrow \mathbb{R}$ 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 $k_w(s) = \sqrt{exp\left(-\left(\frac{s}{w}\right)^2\right)}$, where w is the kernel width that defaults to 0.25.
4. Train a linear *regression* $g : \mathcal{X}' \rightarrow \mathbb{R}$ as the surrogate model. A *sparse* regression is favoured to reduce the dimensionality of 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). The target of the regression is a probability – computed with the black-box model in step 2 – for a class selected by the user to be explained. 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 data point. The feature weights of the surrogate model can be directly compared because all of the features are within the same $[0, 1]$ range. Usually, a separate linear regression is fitted for each of the top 2 or 3 classes predicted by the black-box model f for the original image x as each surrogate can only explain a single class. In practice, this is achieved with a *ridge* regression algorithm[3] implemented in the `scikit-learn` [22] Python package.

[2] 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.

[3] The `sklearn.linear_model.Ridge` class.

A detailed description of LIME for other data domains can be found in the LIME paper [7]. A more recent publication that outlines a generic framework for surrogate explainers from an algorithmic perspective, called bLIMEy, also discusses generalisation and operationalisation of the LIME algorithm [8].

This 4-step process optimises the *fidelity* of the surrogate model and *complexity* of the resulting explanation. The first desideratum 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 Ω, in 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 1, where $\mathcal{G}$ is the set of all the possible (sparse linear) surrogate models.

$$\mathcal{O}(\mathcal{G}; f) = \operatorname*{arg\,min}_{g \in \mathcal{G}} \; \overbrace{\Omega(g)}^{\text{complexity}} + \overbrace{\mathcal{L}(f, g)}^{\text{fidelity}} \tag{1}$$

The *fidelity* of the surrogate model is measured empirically in the vicinity of the explained data point x by evaluating the loss function $\mathcal{L}$ given in Equation 2 for all the data points sampled in the interpretable representation $\mathcal{X}'$. The locality of the metric is enforced by the sampling strategy in $\mathcal{X}'$, which only covers a small region around x, and weighting individual squared differences 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 $IR(x)$, i.e., the explained data point transformed into the interpretable domain $\mathcal{X}'$, and the sampled data points $x' \in \mathcal{X}'$. In Equation 2, the c subscript in f_c indicates the probability of class c computed with the black-box model f.

$$\mathcal{L}(f, g; \mathcal{X}', x) = \sum_{x' \in \mathcal{X}'} k\left(L\left(IR(x), x'\right)\right) \left(f_c\left(IR^{-1}(x')\right) - g(x')\right)^2 \tag{2}$$

Figure 2 shows various stages of LIME. Panel 2a depicts the image to be explained, which has been classified by the black-box *Inception v3* neural network as an *Eskimo dog*. Panel 2b shows the interpretable representation of this image – a (super-pixel) segmentation with $d = 11$ interpretable features. The last two panels of Figure 2 depict a LIME explanation of the

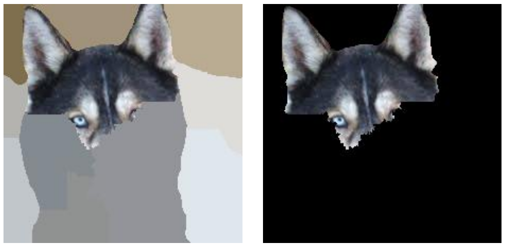
Eskimo dog prediction: panel 2c shows the importance of interpretable features (regression coefficient) and panel 2d displays these segments overlaid on top of the original image.

2.2. LIME Trade-offs

In this section we discuss various trade-offs of the LIME algorithm. 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 segments and object edges affect the explanations. We furthermore touch upon the impossibility of removing information from tabular and image data, which is doable for text, and fidelity issues with surrogates, which are due to inherent randomness and high parametrisation of the LIME algorithm.

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 – to different degrees – for two different classes, leading to a potential confusion. Explaining multiple classes requires training a separate linear model for each of them, 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 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 with features that are highly inter-dependent, e.g.,



(a) Mean-colour occlusion results in an *Eskimo dog* prediction with 77% probability.

(b) Solid black colour occlusion yields 9% probability of *Eskimo dog*, with the top two predictions being *chihuahua* (17%) and *Siberian husky* (59%).

Figure 3: Black-box predictions for a single segment (#3) using different occlusion techniques.

adjacent image segments. This phenomenon can be observed by occluding all of the segments but #3 – visualised in Figure 3a – which is the most important meta-feature according to LIME. In this case, the probability of *Eskimo dog* is 77%, according to the black box, compared to 83% with no occlusions (Figure 2a). However, with both segments #3 and #7 left out – the two most important, and adjacent, segments with respective 0.4 and 0.3 LIME scores – the probability of the same class increases by just 4 percentage points to 81%. 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 mean colour of a segment for its occlusion, for example see Figure 3a. This approach may have undesired effects for some segmentation and colour distribution in an image, in some cases undermining the utility of the occlusion procedure.

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.

Small Size The smaller the segments are, 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.

Preserved Edges 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 segments with their mean colour causes (slight) colour variations of adjacent segments, thus preserving the edges in the (partially) occluded image. Such patterns often convey enough information for the black-box model to recognise the image class correctly, for example in Figure 3a where despite occluding all of the segments but #3 with their mean colour the black-box model still recognises it as *Eskimo dog* with a slight decrease in probability: 77% down from 83%.

Since these issues are artefacts of using a 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. Such approach hides the edges between occluded segments and removes their content instead of just blurring the image, however the edges between occluded and preserved super-pixels will be preserved. 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, for example, see Figure 3b. In this case, when all of the segments but #3 are occluded, the top 3 classes predicted by the black-box model are *Siberian husky* with 59% probability, *chihuahua* with 17% and *Eskimo dog* 9%. This is a drastic change from predicting 77% probability of *Eskimo dog* when using the mean-colour occlusion as illustrated in Figure 3. With a similar effect on other images partially occluded with a solid black colour, the corresponding LIME explanation is different despite using the same data sample from the interpretable domain – see Figure 4. The implicit assumptions of linear models transferred onto the surrogate explanation are also pertinent with this occlusion technique. The 2 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 mean-colour occlusion.

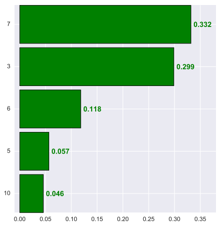


Figure 4: LIME explanation for the Husky image (Figure 2a) using black occlusions. It was generated using the same interpretable domain (binary) data sample as the explanation for the mean-colour occlusion presented in Figure 2c, making them directly comparable. (Segment #5 is the one below #1.)

Observing the influence of each algorithmic component on the variability of LIME explanations has prompted us to reexamine the conclusions drawn by Ribeiro et al. in the LIME paper [7]. In particular, the occlusion colour sensitivity and the resulting unintended consequences cast doubt on the importance of snow in the background of the image shown in Figure 2a as suggested by Ribeiro et al. [7]. Replacing the segments of this picture showing snow with their respective mean colours produces off-white mosaic that still resembles snow, for example, compare the bottom-left and the bottom-right segments in Figures 3a and 2a. These almost visually indistinguishable alterations may therefore have less influence on the probabilities output by a black-box model, as shown in Figure 3, affecting the soundness of LIME explanations.

Impossibility of Information Removal. The significance 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 in there – step 2 of the LIME algorithm outlined in Section 2.1. A similar phenomenon can be observed for tabular data explanations when using an interpretable representation such as discretisation or binning of continuous features [8, 7]. This type of interpretable representation combined with a linear surrogate model yields an explanation that indicates the importance of a particular feature value being within or outside of a given numerical range. Selecting these bin boundaries is non-trivial and biases the explanation in a similar way to the occlusion colour for image explanations. 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 [23, 24, 8]. 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 [25]. The number of parameters and possible component choices when building surrogates further contributes to this phenomenon: number of samples, distance metric, kernel (width), interpretable representation (segmentation) and occlusion colour, to name a few [26]. All in all, surrogates only (locally) approximate the complex behaviour of a black-box model and if their fidelity is miscommunicated to the explainee, such explanations may be misleading.

3. Surrogate Multi-output Regression Tree

In order to alleviate LIME's implicit assumption of a linear relation between the interpretable features and the target variable, and 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) [27] – choosing an appropriate tree type is crucial.

3.1. Motivation

Regression and Non-probabilistic Classification. When the 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 1 and the loss function $\mathcal{L}$ given in Equation 2 remain unchanged. The model complexity function Ω, however, is adapted to trees, thereby measuring either the *depth* of the surrogate or its *width* (number of leaves) as shown later in Equation 5. 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 1, but the loss function $\mathcal{L}$ given in Equation 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 3, with the underline indicating the altered part. Any other classification evaluation metric can be used with $\mathcal{L}$ by modifying it in this manner. Similarly to surrogate regression, the model complexity function Ω is adapted to trees using Equation 5.

$$\mathcal{L}(f, g; \mathcal{X}', x) = \sum_{x' \in \mathcal{X}'} k\left(L\left(IR(x), x'\right)\right) \underline{\mathbb{1}\left(f_c(IR^{-1}(x'), g(x')\right)} \tag{3}$$

Probabilistic Classification. In this paper we focus on a more common scenario, especially for image recognition 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 probabilities vector and proceeding as described above. Doing so, however, is suboptimal as it leads to losing vital information about the confidence of the model's prediction. For example, the top 2 classes maybe 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 [28] 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 [29], which get magnified when treating a probabilistic model 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 and 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 $\neg A = B \cup C \cup \cdots \cup Z$. 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 other classes. However, if $p(A) \leq 0.5$ we cannot be certain whether there is a single event B with $p(B) > p(A)$, 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 further, the numerical output of some surrogate regressors is unbounded, which may be confusing to the explainee, who expects a probability within the $[0, 1]$ range. This last property affects linear models but not regression trees since the latter output the mean of the 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 1 and 2 respectively. Since we are using regression trees, the model complexity function Ω has to be adapted appropriately, as given by Equation 5.

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-box classifiers. While the mechanics 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 $\arg\max$ to black-box classifier's probabilities.

3.2. LIMEtree: Multi-output Regression

To address the issues discussed in the previous section, we propose to use a **multi-output regression tree** as the surrogate model, which 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 predicted by the black box. 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 1 and use either of the decision tree-specific complexity functions Ω given in Equation 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 4, where $\mathcal{C}$ are the classes to be explained by g, for which the c subscript in $g_c(x')$ indicates prediction of a selected class $c \in \mathcal{C}$ for a data point x'. In practice, this is achieved by training a multi-output tree regressor[4] implemented by the `scikit-learn` Python package [22] 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 4 reaches a certain, user-defined level $\epsilon \in [0, 1]$, which corresponds to the fidelity of the local surrogate, i.e., $\mathcal{L}\left(f, g; \mathcal{X}', x\right) \geq \epsilon$. Increasing the complexity of the surrogate model $\Omega(g)$ improves its predictive power, which allows to further minimise the loss $\mathcal{L}$.

$$\mathcal{L}(f, g; \mathcal{X}', x) = \\ \frac{1}{\sum_{x' \in \mathcal{X}'} \omega(x')} \sum_{x' \in \mathcal{X}'} \left(\omega(x') \frac{1}{2} \sum_{c \in \mathcal{C}} \left(f_c \left(IR^{-1}(x') \right) - g_c(x') \right)^2 \right) \tag{4} \\ \text{where} \quad \omega(x'; x) = k \left(L_{\cos} \left(IR(x), x' \right) \right)$$

$$\Omega(g; d) = \frac{\text{depth}(g)}{d} \quad \text{or} \quad \Omega(g; d) = \frac{\text{width}(g)}{2^d} \tag{5}$$

Note that the inner sum $\sum_{c \in \mathcal{C}}$ over the explained classes is normalised by a factor $\frac{1}{2}$ since the biggest squared difference is 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 $\sum_{x' \in \mathcal{X}'}$ is normalised by the sum of weights $\omega(x')$ to ensure that the loss $\mathcal{L}$ is bounded between 0 and 1, facilitating a meaningful comparison of different surrogates and allowing for a meaningful user-defined parameter ϵ.

Putting everything together leads to Algorithm 1, which we call LIMEtree. While the LIMEtree algorithm is relatively lightweight, manipulating images and querying black-box models may become a bottleneck. The explainee 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

[4] The `sklearn.tree.DecisionTreeRegressor` class.

```
Algorithm 1: LIMEtree.
  Data: • black-box model $f$ • explained data point $x$ • interpretable
        representation transformation function $IR$ and its inverse $IR^{-1}$
        • samples number $n$ • set of classes to be explained $C \subseteq \mathcal{C}$
        • distance function $L$ • kernel $k$ • tree depth bound $d$
        • expected fidelity of the local surrogate $\epsilon$
  Result: local surrogate multi-output regression tree

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

number of data points sampled from the interpretable domain. Given the recent advances in dedicated hardware for machine learning applications, this step should not be a burden when utilising GPUs, and manageable with just CPUs. 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 improving the processing time. Other parts of the algorithm, which are executed just once, are relatively efficient: sampling a binary matrix 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.

3.3. Improved Surrogate Fidelity

Our multi-output regression trees can significantly improve the local fidelity of explanations, which, as already demonstrated, has been identified as a major drawback of surrogate explainers [25]. To this end, we use Definition 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 a lack of occlusion for this segment.

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

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

where x'_i is the i^{th} component of the binary vector x'. We can further define a **minimal** set of data points $X'_T \subseteq \mathcal{X}'$, uniquely representing a tree g and the set of its leaves T, which 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 $X_T = \{IR^{-1}(x'_t) : x'_t \in X'_T\}$ with a fixed occlusion colour, e.g., black. 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 overwriting 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 all would be represented by the original (non-occluded) image. This procedure ensures **perfect 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 *bijective* as outlined in Lemma 1, which follows from the discussion presented in the next paragraph.

Lemma 1. *A decision tree surrogate can achieve* ***perfect 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* ***bijective****. This means that:*

- *the mapping from* $\mathcal{X}$ *to* $\mathcal{X}'$ *is a* one-to-one correspondence *and*
- *the* IR *function has 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* $x' \in \mathcal{X}'$ *and vice versa.*

Intuitively, the two properties listed in Lemma 1 imply that each leaf in the surrogate tree is associated with only a single 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 $g_{\text{id}}(x'_t) = t$. 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 perfect fidelity for the minimal representation set, which in turn is the backbone of model-driven explanations. The interpretable representation for images introduced in this paper, and used by LIME [7], is bijective since the occlusion function is **deterministic**, which is achieved by fixing the occlusion strategy: an identical colour for all segments in our experiments (Figure 3b) and a segment-specific mean colour occlusion in LIME (Figure 3a).

While ensuring perfect 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 3 binary features $(x_1, x_2, x_3) \in \{0, 1\}^3$, a root-to-leaf path with $x_1 < 0.5 \wedge x_3 < 0.5$ conditions assigns $0, 0, 0$ and $0, 1, 0$ to this leaf. This observation prompted us to specify the minimal interpretable representation X'_T (Definition 1) that assigns a single data point to represent each leaf, thereby facilitating perfect fidelity of model-driven explanations without additional assumptions. However, to achieve *perfect fidelity* for *data-driven* explanations, the surrogate tree must faithfully model the 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 Ω accordingly guarantees perfect fidelity of all the explanations, which is expressed in the following Corollary.

Corollary 1. *If the complexity bound Ω (width) of the surrogate tree g is relaxed to be equal to the cardinality of the binary interpretable domain $\mathcal{X}'$, i.e., $\Omega(g) = |\mathcal{X}'|$, the surrogate is guaranteed to achieve **perfect 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.*

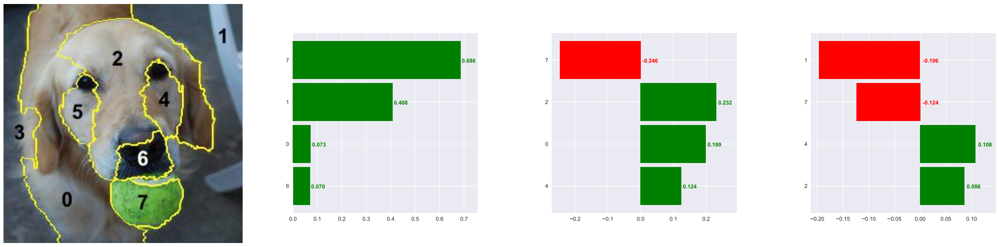
Therefore, a surrogate tree that guarantees faithfulness of *model-driven* explanations (Lemma 1) can only deliver trustworthy counterfactuals and exemplar explanations sourced from the minimal representation set. For such surrogates we can also generate what-if explanations with perfect fidelity by 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 1) whenever the black-box predictive function is accessible to the explainee and querying it is not prohibitively expensive (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.

4. LIMEtree Explanation Examples

To support the discussion and experimental results presented in the following sections, we first introduce examples of LIMEtree explanations and compare them with equivalent explanations produced by LIME [7]. After personalising the interpretable representation, as shown in Figure 5a, we explain the top three classes predicted by the black-box model: *tennis ball* (99.56%), *golden retriever* (0.42%) and *Labrador retriever* (0.02%). Their LIME explanations are given in Figure 5. As expected, segment #7, which depicts the ball, has an overwhelmingly positive influence on the *tennis ball* prediction – see Figure 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 – #1, #0 and #6 – are adjacent and fully surround it. The second most important segment for this class is #1, which magnitude is almost 6 times larger than the magnitude of the next 2 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 Figures 5c and 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 2 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 that segment #1 plays.

Next, we explain these 3 classes with LIMEtree, which can produce various types of explanations, helping us to analyse the behaviour of the black box. We have already shown one type of explanation – the surrogate tree structure visualisation – in Figure 1. The depth of this tree was limited to 2 for the purpose of presentation, therefore the tree complies with Lemma 1 but not with Corollary 1, only achieving perfect fidelity with respect to model-driven explanations. Another explanation type, which closely resembles LIME explanations, is the importance of interpretable features (calculated as *Gini importance* [30]) shown in Figure 6a. Since LIMEtree models all the 3 classes simultaneously, the importance captures the image segments



(a) Segments of the explained image.

(b) *Tennis ball* (99.56%) LIME explanation.

(c) *Golden retriever* (0.42%) LIME explanation.

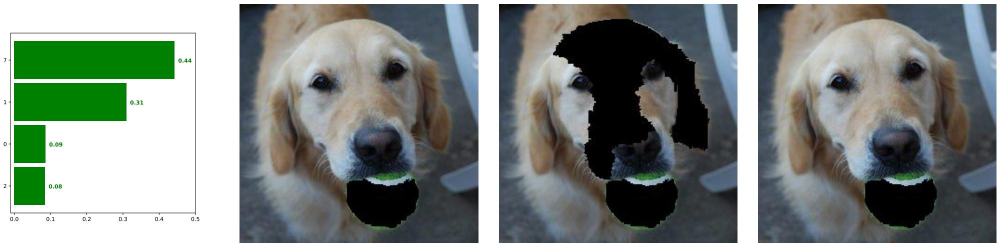
(d) *Labrador retriever* (0.02%) LIME explanation.

Figure 5: LIME explanations for the top 3 classes predicted by a black-box model for the image given in panel (a).

that help to differentiate between these classes. Comparing Figure 6a with analogous LIME explanations in Figure 5 shows a nice overlap, with each LIME explanation sharing 3 of its 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 this 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 1, we can discover the relation between each important feature and the 3 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 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



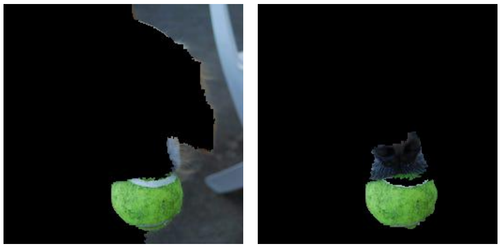
(a) LIMEtree feature importance. All 3 classes share one explanation.

(b) LIMEtree what-if explanation. “What if the ball was not there?” → 98% *golden retriever*.

(c) LIMEtree exemplar explanation. “Show me an example of a *Labrador retriever*.”

(d) LIMEtree exemplar explanation. “Show me an example of a *golden retriever*.”

Figure 6: Three types of LIMEtree explanations: (a) feature importance, (b) what-if explanation and (c)&(d) exemplar explanations.



(a) Segments #7 and #1 preserved – 90% *tennis ball*.

(b) Segments #7 and #6 preserved – 66% *tennis ball*.

Figure 7: The shortest LIMEtree explanations for *tennis ball*.

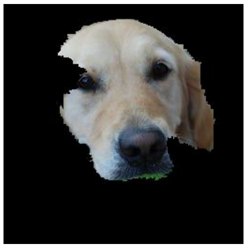
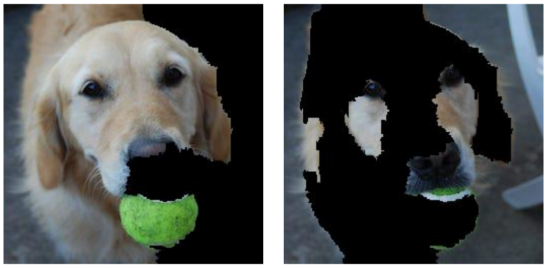


Figure 8: Visual representation of a LIMEtree rule explanation that maximises the *Labrador retriever* prediction (99%).

of explanations, for example, interactive what-if questions. Since the tree presented in Figure 1 is not complete (see Corollary 1), we use the black-box model instead of the tree to evaluate the 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 Figure 6b. We can also ask for exemplar explanations of the *Labrador retriever* and *golden retriever* classes, which are shown in Figures 6c and 6d respectively.

In order to take full advantage of LIMEtree explanations, we train a *complete* surrogate tree (see Corollary 1). 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



(a) How can we get a *golden retriever* prediction (91%) given #7.

(b) Occluding segments #0, #2 and #7 yields *volcano* (17%) according to the black box.

Figure 9: Customised counterfactual explanations.

and #1 and another with segments #7 and #6 preserved, both of which are shown in Figure 7. We can also ask the tree for a rule explanation (root-to-leaf path) of *Labrador retriever*, resulting in the maximal possible confidence of the black-box model for this class. The resulting explanation is $f_0 = 0 \land f_1 = 0 \land f_2 = 1 \land f_3 = 0 \land f_4 = 1 \land f_5 = 1 \land f_6 = 1 \land f_7 = 0$, giving us 99% confidence. Such representation is not particularly appealing, however, as discussed in Section 5.2.3, we can represent it in the visual domain – see Figure 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 Figure 9a – the model predicts *golden retriever* (91%). Since occluding segment #7, i.e., the ball, results in 98% *golden retriever* (see Figure 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 Figure 9b.

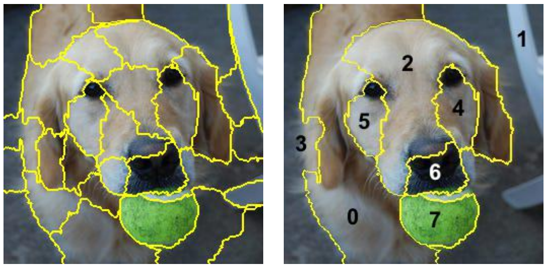
5. Discussion

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 data point, the explanation can be further customised. We explore the personalisation and interactiveness of LIMEtree explanations in Section 5.1, which explains how to customise the interpretable representation, the explanation type and its content. Using a multi-output regression tree as the surrogate model enables accurate local mimicking of black-box probabilistic models for multiple classes simultaneously, making it appealing and compatible with modern predictive models such as deep neural networks. LIMEtree works equally well for data types other than images, e.g., tabular and text, and its perfect fidelity desideratum can be achieved in practice while preserving low complexity of explanations, which is discusses in Section 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 predictions explanation.

5.1. Personalised and Interactive Explainability

No matter how comprehensive an explanation is, it may not appeal to all explainees or answer all their questions [13]. 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 [2]. LIMEtree allows various aspects of its explanations to be interactively personalised, in particular the interpretable representation, explanation type 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 case of images, computer generated segments (Figure 10a) may be good enough to produce meaningful explanations, we encourage the user to either provide custom segmentation or indicate which of the computer-generated segments should be merged (Figure 10b). This step aims to achieve an interpretable representation, i.e., image segmentation, that conveys meaningful



(a) Default, computer-generated segmentation of an image (quick shift).

(b) Personalised image segmentation achieved by merging user-specified super-pixels.

Figure 10: Default and custom interpretable representations. The top two classes predicted by a black box are 99.6% *tennis ball* and 0.4% *golden retriever*.

concepts, which may be different for individuals with different levels of domain expertise and background. Similar reasoning applies to tabular and text data where the explainee can respectively customise binning of continuous features and tokenisation of sentences, e.g., match selected words to form a tuple considered as a single token in the interpretable representation [13]. After fixing the interpretable representation, a surrogate tree is fitted and its leaves relabelled as per Lemma 1, which is then used to extract various explanations. Since personalised interpretable representations tend to be small in size (see Figure 10), often a complete tree – according to Corollary 1 – can be fitted, yielding more diverse and faithful explanations.

Explanation Type. A linear regression surrogate is limited to explaining interpretable features (image segments) with their importance for each class separately. On the other hand, a multi-output regression tree can explain its behaviour with a wide range high-fidelity artefacts discussed in the introduction (Section 1):

1. tree structure visualisation,
2. interpretable feature importance;
3. logical conditions;
4. exemplar explanations,
5. answers to what-if questions, and
6. counterfactuals.

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 explainees the control over their content [12, 13].

Explanation Content. The most prominent and appealing kinds of explanations are *contrastive* and *counterfactual* explanations, which, arguably, are the most natural explanations for humans [2], and are compliant with various legal regulations and requirements such as the European Union's General Data Protection Regulation (GDPR) [11]. These 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 provided 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 these conditions and uses an explicit class contrast. Contrastive and counterfactual explanations were shown to be capable of supporting interactions via an explanatory dialogue as well as being easy and efficient to obtain from decision trees [12, 13]. 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 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 explanations and completeness of the surrogate tree – see Section 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 explainees to inspect the influence of each logical condition on 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*, which can only be made interactive by embedding them in an interactive interface and are otherwise static. For example, the tree structure can be presented in an interface that allows the explainee 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 6 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 audience. Section 6 includes examples of these explanations, which showcase their power and the benefits of their interactiveness.

5.2. Generalisability and Applicability

LIMEtree explanations are versatile and appealing but their fidelity guarantees require a *bijective* interpretable representation transformation function IR, which has a unique inverse IR^{-1} (Lemma 1), and a *complete* surrogate tree (Corollary 1) as outlined in Section 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 preserving 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 ramification does not hold for the most important explanations. All of these arguments should convince the reader that in many cases LIMEtree can be easily generalised and safely deployed without affecting its performance.

5.2.1. Tabular and Text Data

This work focuses on explaining black-box probabilistic image classifiers, but in Section 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 components in all of these use cases are the *interpretable representation* and the bijective function responsible for transforming data between the original and the interpretable domains. For images, we provided an example of each component – a binary representation encoding super-pixel occlusions – analysed their properties and discussed their pros, cons and implications, showing how to design them to mitigate possible issues (Section 2.2). This overview led us to conclude that making the interpretable representation transformation function *deterministic* is crucial for LIMEtree to achieve perfect fidelity – see Section 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 presence (1) or absence (0) of a given token. This representation complies with all of the properties discussed in Section 3.3 and required for LIMEtree to achieve high fidelity. To guarantee that the interpretable representation transformation function is deterministic, the order of words (tokens) in the text excerpt has been 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 (discussed in Section 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-box models rather than the interpretable domain itself – language models do not take fix-length or -shape input.

In contrast, defining an interpretable representation transformation function for tabular data with numerical features that is bijective and has a unique inverse, i.e., complies with Lemma 1, is challenging. The most popular approach [7] is discretisation followed by binarisation via one-hot encoding, e.g., a numerical feature x_3 with value 7 can be discretised into 3 bins: $(-\infty, -3]$, $(-3, 8]$ and $(8, \infty)$, which are binarised to $[0, 1, 0]$, indicating that $x_3 = 7$ falls into the middle bin. While this does not affect categorical features – their original representation can be uniquely reconstructed from the binarised form since they do not have to be discretised – the same is not true for numerical features, making the function *non-injective surjective*, i.e., non-bijective. 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 [8]. For example, LIME [7] "computes" this inverse by sampling from a truncated (at bin boundaries) Gaussian distribution fitted to each numerical bin, hence 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 changes the characteristic and interpretation of the explanations. For example, when the surrogate is a linear model (LIME's approach), the explainer is not anymore a sensitivity analysis tool of interpretable features; instead the explanations convey the importance of raw features. In this case, dropping the interpretable representation also requires the numerical features to be normalised to $[0, 1]$ range and the categorical features to be one-hot encoded for the importance values to be 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 via interaction as discussed in Section 5.1.

Depending on the surrogate model choice, coming up with an interpretable domain may be unnecessary altogether. Importantly, decision trees learn their own discrete representation of data by applying binary splits, thereby creating locally faithful and meaningful binning for continuous and grouping for categorical features [8]. Furthermore, non-bijectiveness of this interpretable representation transformation function can be overcome from an algorithmic perspective by first *locally* sampling data from their original domain and then transforming them into the interpretable domain [8]. This is uniquely possible for this type of data and is the reverse of the standard procedure: steps 1–3 in Algorithm 1 (corresponding to step 2 described in Section 2.1), which mitigate the need of applying the ill-defined IR^{-1} function. Applying this "trick", however, will not allow the surrogate to achieve perfect fidelity, which requires the interpretable domain transformation function to be bijective (Lemma 1). Without satisfying this property it is also impossible to build a *complete* surrogate tree (Corollary 1), nevertheless since we are dealing with raw tabular data, we can overfit the tree to the local sample, thereby achieving high enough fidelity.

5.2.2. Perfect Fidelity in Practice

Assuming that the interpretable representation transformation function satisfies the properties outlined in Lemma 1, i.e., it is bijective and invertible, perfect fidelity of the surrogate is achieved in practice by adjusting the *sample size n* and relaxing the *complexity bound* Ω of the tree by removing the depth constrain d in Algorithm 1. Lemma 1 is easily satisfied in practice for image and text data. While it 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 in Section 5.2.1. Focusing on text and image data, an appropriate sample size and tree depth bound are achieved by operationalising Corollary 1. For these data types, each dimension of the interpretable domain can be treated as a human-comprehensible concept, e.g., ears, eyes and muzzle for a dog image, which will often result in relatively few concepts for each explained data point. Please note that words (tokens) or image segments 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 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 Ω relaxed to allow trees of depth 10 – a complete, balanced binary tree of depth d has 2^d leaves (its width), i.e., one leaf per data point, thus guaranteeing *perfect fidelity* of the whole tree. The depth bound and the sample size can be adjust dynamically prior to training the local surrogate tree since the size of the interpretable domain is known beforehand, thereby reducing the complexity of the tree. 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 perfect 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 relaticely 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 complexity of some explanations, however it does not affect the most important and versatile explanation types as discussed below.

5.2.3. 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 perfect fidelity requires relaxing the depth bound of the surrogate Ω, which the optimisation objective $\mathcal{O}$ tries to minimise (Equation 1). While high complexity of a surrogate tree may render the explanations based on the tree structure, e.g., model visualisation, incomprehensible, these are not the most appealing explanation types and possibly require machine learning expertise to interpret. The interpretable feature importance, what-if explanations, counterfactuals and exemplars are not affected by the tree complexity in any way and are still highly interpretable, compact and accessible [12] with their interactive and customisable nature adding to their appeal as discussed in Section 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.

For images and text data, regardless of the rule length, its presentation will always be comprehensible. These rules cannot have more literals than the number of dimensions in the interpretable domain, translating to 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 and in the latter producing a text excerpt with various 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, thereby implicitly reducing the size of the explanation. In this case visualisations are also not a viable alternative due to the inherent limitation of the human perceptual system to 3 dimensions, with an additional capacity enabled when considering time, e.g., when explaining time series. Finally, a general criticism of rule-based explanations postulating that it is difficult to understand how each logical condition affects the prediction makes them less appealing than other types of explanations.

Therefore, if tree structure-based explanations are not required for image and text data, and additionally rule-based explanations are not needed for tabular data, the complexity of the tree Ω 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 1 and the optimisation step 7 in Algorithm 1 can be skipped, paving the way for perfect 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 proprietary [31].

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 6.1) and "human-grounded" (Section 6.2) experiments [32]. 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 distributed within PyTorch [33], and the surrogate explainers were built on top of FAT Forensics [34] using bLIMEy algorithmic framework [8].

6.1. Synthetic Experiments

To understand how LIMEtree behaves in various settings we use a few proxy metrics to experimentally evaluate its performance. First, we measure the faithfulness of the surrogate with respect to the black box, i.e., its ability to mimic the black box, which indirectly indicates the trustworthiness of surrogate explanations. To this end, we report the fidelity as measured by the LIME loss $\mathcal{L}$ given in Equation 2 and the LIMEtree loss $\mathcal{L}$ defined in Equation 4. We do so when modelling the top 3 classes predicted by the black box for 4 different surrogate approaches: LIME and 3 variants of LIMEtree. To complement the discussion presented in Section 5.2.3 we also analyse the complexity Ω of LIMEtree surrogates as defined in Equation 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 [34], which uses black as the occlusion colour and does not use feature selection, making it the most powerful variant of

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

Table 1: Per-class fidelity computed with LIME loss (Equation 2) of different surrogate approaches for the top 3 black-box predictions computed for a sample of 100 surrogates. (Smaller is better.)

LIME since it has access to all of the interpretable features. The results presented in Tables 1 and 2 contain fidelity for 3 variants of LIMEtree:

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

LIMEt a tree optimised for complexity, which predictions are post-processed to guarantee perfect fidelity of model-driven explanations (see Section 3.2 for more details); and

LIMEt* a surrogate tree without complexity constraints, allowing the algorithm to learn complete trees.

Table 1 measures the fidelity of the surrogates with the LIME loss given in Equation 2 separately for each of the top 3 classes predicted by the black box. The LIME algorithm produces 3 independent linear surrogates – one for each class – while each LIMEtree variant gives a single surrogate that models all of the classes simultaneously. Measuring the fidelity of each class separately helps us to visualise 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 probability of the other two classes being much smaller. Similarly, Table 2 measures the fidelity of the surrogates with the LIMEtree loss given in Equation 4 for the top 1, 2 and 3 classes predicted by the black box. Again, the LIME algorithm produces 3 independent linear surrogates – one for each class – and each LIMEtree variant gives separate model for a 1-class, 2-class and 3-class problem. These results are a mean fidelity of surrogates trained for 100 random images from the ImageNet [28] validation set, computed over all the possible data points in the binary interpretable domain.

Both Tables 1 and 2 show that our base method – **LIMEt** – outperforms LIME. The LIMEtree variant that achieves perfect fidelity for model-driven

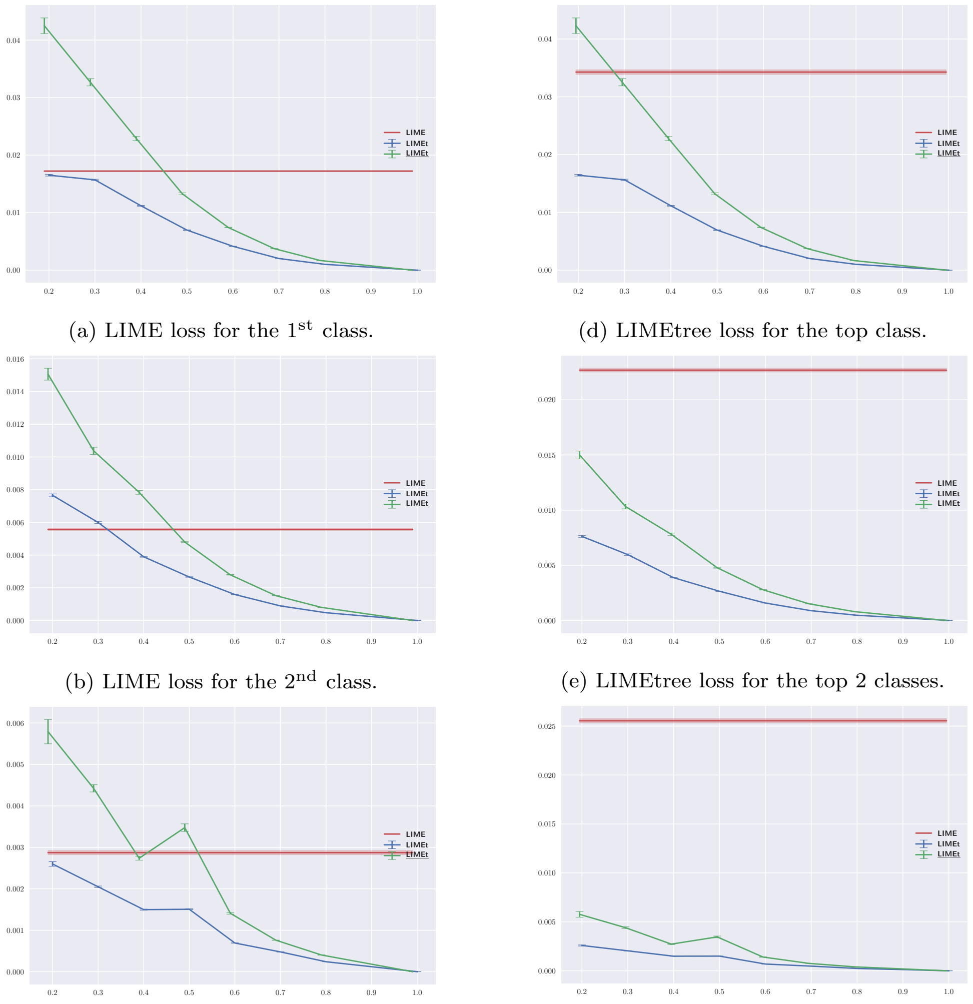
top n	**LIME**	**LIMEt**	$\underline{\textbf{LIMEt}}$	**LIMEt***
1 class	0.0343 ± 0.0004	**0.0069** ± 0.0001	0.0144 ± 0.0003	**0** ± 0
2 classes	0.0227 ± 0.0002	**0.0026** ± 0.0000	0.0045 ± 0.0000	**0** ± 0
3 classes	0.0255 ± 0.0002	**0.0012** ± 0.0000	0.0029 ± 0.0001	**0** ± 0

Table 2: Fidelity of the top n classes computed with LIMEtree loss (Equation 4) of different surrogate approaches for the top 3 black-box predictions computed for a sample of 100 surrogates. When computing the LIMEtree loss for 1 class the factor of $\frac{1}{2}$ is removed. (Smaller is better.)

explanations (via prediction post-processing) – $\underline{\textbf{LIMEt}}$ – performs comparably to LIME when measuring the fidelity with 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 they are tuned to be faithful for the minimal interpretable data points representing them. The surrogate complexity Ω 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 only requires half of the interpretable features (i.e., half of the maximum depth) to achieve this level of performance. Finally, a surrogate tree with unconstrained depth – **LIMEt*** – is achieving perfect fidelity across the board, which is expected since it is expressive enough to cover the whole interpretable data space, creating one leaf for each data point if needed.

Surrogate Complexity. Next, we investigate the relation between the depth-based complexity of the surrogate tree Ω 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 accounts for that by scaling the tree depth according to the number of segments, 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. This empirical evidence – visualised in Figure 11 – supports our discussion presented in Section 5.2.3.

When using the LIME loss as our fidelity metric **LIMEt** requires at most 33% and $\underline{\textbf{LIMEt}}$ needs at most 55% of all the interpretable features to perform on par with LIME regardless of the number or configuration of the explained classes. For LIMEtree loss **LIMEt** performs better than LIME with just 20% of interpretable features and $\underline{\textbf{LIMEt}}$ needs at most 30% of



(a) LIME loss for the 1^{st} class.

(d) LIMEtree loss for the top class.

(b) LIME loss for the 2^{nd} class.

(e) LIMEtree loss for the top 2 classes.

(c) LIME loss for the 3^{rd} class.

(f) LIMEtree loss for the top 3 classes.

Figure 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, for the top 3 classes predicted by the black box. Panels (a), (b) & (c) depict LIME loss; and panels (d), (e) & (f) depict LIMEtree loss. Please note different scales on the y-axes.

them. **LIMEt** requires deeper trees to achieve the same level of performance as **LIMEt** since the post-processing step applied to ensure perfect fidelity of model-driven explanations causes the surrogate to be a sub-optimal predictor for a majority of the interpretable data. By allowing deeper trees we reduce the variance of leaves, which improves the overall performance of a surrogate – a clear relation between the complexity of the tree Ω 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 (d), (e) & (f) of Figure 11 as different rates of convergence.

6.2. User Study

To assess the usefulness of LIMEtree 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 [7], 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 1, alongside 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 machine learning background, with 3 participants being familiar with ML explainability. The participants were not compensated for their involvement in the user study.

The study consisted of 2 main sections – one devoted to LIME and one concerning LIMEtree – displaying an image divided into 3 segments, with each segment enclosing a unique object, e.g., a cat, a dog and a ball. The 2 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 *tennis ball* and *croquet ball* for the ball object. Thus in this case, the explainee was exposed to 6 LIME explanations, each showing the importance of 3 segments (one per object), and a single tree of depth 3 modelling all 6 predictions. For each explainability method, the participants were asked about the expected behaviour of the black-box model in relation to any 2 out of the 3 displayed objects – 6 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 a presence of *the dog object*?", with 3 possible answers: confidence decreases, confidence not affected and confidence increases.

This particular question was chosen to avoid a bias towards either of the explainability method since 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 the opposite. 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 assess whether they still relied on their intuition when asked to work with explanations instead.

Our findings show that regardless of the exposure order LIMEtree helped the participants to answer 25% more questions correctly as compared to LIME. The negligible overlap between the answers based on the participants' intuition and for either of the two explainability methods shows that the participants based their answers on the explanation 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 via 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 explanations. 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.

7. Related Work

Our research 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* and *counterfactual* explanations since they occur naturally in human interactions [2]. Following this observation, XAI and IML research proliferated in recent years [11, 9] with Miller [2] summarising their importance, grounding them in social sciences, highlighting the fundamental role they play in human explainability and showing the lack of consideration for human aspects in the current literature [35]. However, another aspect of human-centred explainability pointed out by Miller [2] has largely gone unnoticed: their interactive and bi-directional, dialogue-like nature, which allows the explainee to guide the explainer, hence receive tailored explanations. Schneider and Handali [14] have recently review 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 and various frameworks to model explanatory interactions between two intelligent agents, be them humans, machines or one of each [15, 16, 17]. Weld and Bansal [3], on the other hand, discussed various properties of explanatory systems and hypothesised how such interactions could look like in the real life, albeit focusing more on multiple explanation modalities and not explanation personalisation per se. A mixture of explainability and interactivity has also been used to refine (e.g., personalise) and improve various data modelling techniques. Kulesza et al. [18] used explanations of a naïve Bayes classifier to help the user "debug" and "personalise" the classification of electronic mail and Kim et al. [19] 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 [36], were fitted into interactive user interfaces [1, 37] to provide the user with a freedom to explore these explanations. Finally, Sokol and Flach discussed the importance of interactive personalisation in ML interpretability [13] and showed how counterfactual explanations can be interactively customised based on explainee's preferences [12].

The second concept that our work builds upon is surrogate explainability [6, 7]: a model-agnostic and post-hoc technique that works 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 [7], 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 [6], which is based on a decision tree. High modularity and flexibility of these explainers [8] encouraged the community to compose their different variant, some of which use decision trees as a local surrogate [9, 10, 8]. Waa et al. [9] showed how a local one-vs-rest classification tree can be used to produce contrastive explanations; and Shi et al. [10] fitted a local shallow regression tree and used its structure as an explanation. Both of these methods use a local tree surrogate, however none of them utilises the full explainability (and interactivity) potential that they enable. Explainability of decision trees [12] and their ensembles [38] have also been investigated outside of the surrogate context. Sokol and Flach [12] showed how to extract personalised counterfactual explanations by interacting with a decision tree via a voice interface and Tolomei et al. [38] introduced a method to explain predictions made by ensembles of decision tree classifiers with class-contrastive counterfactuals.

8. Conclusions and Future Work

In this paper we introduced LIMEtree: a *local* surrogate explainer of black-box *predictions* based on *multi-output regression trees*. We discussed properties of interpretable domains, required to make these explainers work with any type of data (image, text and tabular), and showed how they can be designed and used to achieve the best performance, focusing on images but discussing text and tabular data as well. We then demonstrated how LIMEtree improves upon LIME [7] by simultaneously modelling multiple classes and discussed 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 and evaluated our approach with quantitative experiments and 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. At present, some explanation types have to be extracted manually from the tree, which we will address in future work with an algorithmic approach to parse the tree structure for counterfactual explanations based on a user-specified heuristic. To this end, we will supplement LIMEtree with an interactive interface via which the explainee can request and personalise the explanations. We will also investigate some of the technical properties of our method, namely alternatives to occlusion with a fixed colour and techniques for calibrating the probabilities output by the black-box model when it is overconfident, which may result in extreme probabilities for some of the classes. Moreover, we plan to research interpretable domains for all 3 data types, analyse their properties and include them in our LIMEtree implementation, thereby enhancing the versatility of our tool and extending its applicability to text and tabular data. Finally, we will carry out user studies to empirically evaluate the influence of each explanation type on its own and all of them collectively on the perceived transparency improvement, and assess the benefit of the *interactive* explanatory process over static explanations.

Acknowledgements

The authors would like to thank Alexander Hepburn and Raul Santos-Rodriguez for insightful discussions and their help with developing the code used for the experiments. Kacper Sokol was funded by the SPHERE IRC project under the UK Engineering and Physical Sciences Research Council (EPSRC), Grant EP/K031910/1. The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] J. Krause, A. Perer, K. 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, ACM, 2016, pp. 5686–5697.

[2] T. Miller, Explanation in artificial intelligence: Insights from the social sciences, Artificial Intelligence (2018).

[3] D. S. Weld, G. Bansal, The challenge of crafting intelligible intelligence, Communications of the ACM 62 (2019) 70–79.

[4] M. J. Kusner, J. Loftus, C. Russell, R. Silva, Counterfactual fairness, in: Advances in Neural Information Processing Systems, 2017, pp. 4066–4076.

[5] T. Kulesza, S. Stumpf, M. Burnett, W.-K. Wong, Y. Riche, T. Moore, I. Oberst, A. Shinsel, K. McIntosh, Explanatory debugging: Supporting end-user debugging of machine-learned programs, in: 2010 IEEE Symposium on Visual Languages and Human-Centric Computing, IEEE, 2010, pp. 41–48.

[6] M. Craven, J. W. Shavlik, Extracting tree-structured representations of trained networks, in: Advances in neural information processing systems, 1996, pp. 24–30.

[7] M. T. Ribeiro, S. Singh, C. Guestrin, Why should i trust you?: Explaining the predictions of any classifier, in: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, ACM, 2016, pp. 1135–1144.

[8] K. Sokol, A. Hepburn, R. Santos-Rodriguez, P. Flach, bLIMEy: Surrogate Prediction Explanations Beyond LIME, 2019 Workshop on Human-Centric Machine Learning (HCML 2019) at the 33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada (2019). URL: https://arxiv.org/abs/1910.13016, arXiv preprint arXiv:1910.13016.

[9] J. v. d. Waa, M. Robeer, J. v. Diggelen, M. Brinkhuis, M. Neerincx, Contrastive explanations with local foil trees, in: Proceedings of the ICML Workshop on Human Interpretability in Machine Learning (WHI 2018), Stockholm, Sweden, 37, 2018.

[10] S. Shi, X. Zhang, H. Li, W. Fan, Explaining the predictions of any image classifier via decision trees, arXiv preprint arXiv:1911.01058 (2019).

[11] S. Wachter, B. Mittelstadt, C. Russell, Counterfactual explanations without opening the black box: Automated decisions and the gpdr, Harv. JL & Tech. 31 (2017) 841.

[12] K. Sokol, P. A. Flach, Glass-box: Explaining ai decisions with counterfactual statements through conversation with a voice-enabled virtual assistant., in: IJCAI, 2018, pp. 5868–5870.

[13] K. Sokol, P. Flach, One explanation does not fit all: The promise of interactive explanations for machine learning transparency, KI-Künstliche Intelligenz (2020) 1–16.

[14] J. Schneider, J. P. Handali, Personalized explanation for machine learning: A conceptualization (2019).

[15] D. Walton, Dialogical models of explanation., ExaCt 2007 (2007) 1–9.

[16] A. Arioua, M. Croitoru, Formalizing explanatory dialogues, in: International Conference on Scalable Uncertainty Management, Springer, 2015, pp. 282–297.

[17] P. Madumal, T. Miller, L. Sonenberg, F. Vetere, A grounded interaction protocol for explainable artificial intelligence, in: Proceedings of the 18th International Conference on Autonomous Agents and Multi-Agent Systems, International Foundation for Autonomous Agents and Multiagent Systems, 2019, pp. 1033–1041.

[18] T. Kulesza, M. Burnett, W.-K. Wong, S. Stumpf, Principles of explanatory debugging to personalize interactive machine learning, in: Proceedings of the 20th international conference on intelligent user interfaces, ACM, 2015, pp. 126–137.

[19] B. Kim, E. Glassman, B. Johnson, J. Shah, ibcm: Interactive bayesian case model empowering humans via intuitive interaction (2015).

[20] A. Vedaldi, S. Soatto, Quick shift and kernel methods for mode seeking, in: European conference on computer vision, Springer, 2008, pp. 705–718.

[21] S. van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, T. Yu, the scikit-image contributors, scikit-image: image processing in Python, PeerJ 2 (2014) e453. URL: `https://doi.org/10.7717/peerj.453`. doi:`10.7717/peerj.453`.

[22] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, Journal of Machine Learning Research 12 (2011) 2825–2830.

[23] T. Laugel, X. Renard, M.-J. Lesot, C. Marsala, M. Detyniecki, Defining locality for surrogates in post-hoc interpretablity, 2018 ICML Workshop on Human Interpretability in Machine Learning (WHI 2018) (2018).

[24] Y. Zhang, K. Song, Y. Sun, S. Tan, M. Udell, "Why Should You Trust My Explanation?" Understanding Uncertainty in LIME Explanations, AI for Social Good Workshop at the 36th International Conference on Machine Learning (ICML 2019), Long Beach, California (2019). URL: `https://arxiv.org/abs/1904.12991`, arXiv preprint arXiv:1904.12991.

[25] C. Rudin, Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead, Nature Machine Intelligence 1 (2019) 206–215.

[26] D. Garreau, U. von Luxburg, Explaining the Explainer: A First Theoretical Analysis of LIME, The 23rd International Conference on Artificial Intelligence and Statistics (AISTATS 2020) – to appear (2020). ArXiv preprint arXiv:2001.03447.

[27] L. Breiman, J. Friedman, C. J. Stone, R. A. Olshen, Classification and regression trees, CRC press, 1984.

[28] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, ImageNet: A Large-Scale Hierarchical Image Database, in: CVPR09, 2009.

[29] M. Kull, M. Nieto, M. Kangsepp, T. Silva Filho, H. Song, P. Flach, Beyond temperature scaling: Obtaining well-calibrated multi-class probabilities with dirichlet calibration, in: Advances in Neural Information Processing Systems 31, 2019.

[30] L. Breiman, Random forests, Machine learning 45 (2001) 5–32.

[31] K. Sokol, P. Flach, Counterfactual explanations of machine learning predictions: Opportunities and challenges for ai safety, 2019 Workshop on Artificial Intelligence Safety (SafeAI 2019) at the 33rd AAAI Conference on Artificial Intelligence (AAAI-19), Honolulu, Hawaii, USA (2019).

[32] F. Doshi-Velez, B. Kim, Towards a rigorous science of interpretable machine learning, arXiv preprint arXiv:1702.08608 (2017).

[33] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner,

L. Fang, J. Bai, S. Chintala, Pytorch: An imperative style, high-performance deep learning library, in: H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, R. Garnett (Eds.), Advances in Neural Information Processing Systems 32, Curran Associates, Inc., 2019, pp. 8024–8035.

[34] K. Sokol, R. Santos-Rodriguez, P. Flach, Fat forensics: A python toolbox for algorithmic fairness, accountability and transparency, arXiv preprint arXiv:1909.05167 (2019).

[35] T. Miller, P. Howe, L. Sonenberg, Explainable ai: Beware of inmates running the asylum or: How i learnt to stop worrying and love the social and behavioural sciences, arXiv preprint arXiv:1712.00547 (2017).

[36] J. H. Friedman, Greedy function approximation: a gradient boosting machine, Annals of statistics (2001) 1189–1232.

[37] J. Krause, A. Perer, E. Bertini, Using visual analytics to interpret predictive machine learning models, 2016 ICML Workshop on Human Interpretability in Machine Learning (WHI 2016) (2016). ArXiv preprint arXiv:1606.05685.

[38] G. Tolomei, F. Silvestri, A. Haines, M. Lalmas, Interpretable predictions of tree-based ensembles via actionable feature tweaking, in: Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining, ACM, 2017, pp. 465–474.