Explainable Artificial Intelligence on Smart Human Mobility: a comparative study approach

Luís Rosa[1] [0000−0001−9967−2680], Fábio Silva[1,2] [0000−0001−9872−7117], and Cesar Analide[1] [0000−0002−7796−644X]

[1] University of Minho, ALGORITMI Center, Dep. of Informatics, Braga, Portugal
{id8123,analide}@di.uminho.pt
[2] CIICESI, ESTG, Politécnico do Porto, Felgueiras, Portugal fas@estg.ipp.pt

Abstract. Explainable artificial intelligence has been used in several scientific fields to understand how and why a machine learning model makes its predictions. Its characteristics have allowed for greater transparency and outcomes in AI-powered decision-making. This building trust and confidence can be useful in human mobility research. This work provides a comparative study in terms of the explainability of artificial intelligence on smart human mobility in the context of a regression problem. Decision Tree, LIME, SHAP, and Seldon Alibi are explainable approaches to describe human mobility using a dataset generated from New York Services. Based on our results, all of these approaches present relevant indicators for our problem.

Keywords: Explainable Artificial Intelligence · Machine Learning · Smart Cities· Smart Human Mobility.

1 Introduction

Machine Learning (ML) algorithms have been used in many application domains. These algorithms are being employed to complement humans' decisions in various tasks from diverse domains, such as finance, travel and hospitality, law enforcement, health care, news and entertainment, logistics, and manufacturing [13]. However, they still face acceptability issues. The major disadvantage of Deep Learning (DL) remains that its numerous parameters are challenging to interpret and explain. This especially holds true for opaque decision-making systems which are considered complex black box models.

The inability for humans to see inside black boxes can result in an increased need for interpretability, transparency, and explainability of AI products/outputs like predictions, decisions, actions, and recommendations. These elements are required to ensure the explanation of ML decisions or functionality. In [7], Explainable Artificial Intelligence (XAI) refers to techniques and methods of explaining so that AI solution results can be understood by humans.

On human mobility context some XAI projects have been studied. For example, [8] focuses on the topic of automatic detection of Search and Rescue (SAR) missions, by developing and evaluating a methodology for classifying the trajectories of vessels that possibly participate in such missions. Luca et al. provides a taxonomy of mobility tasks where a discussion on the challenges related to each task and how deep learning may overcome the limitations of traditional models and a description of the most relevant solutions to the mobility tasks are described [10]. However, unfortunately, scarce contributions have been made to explore the human mobility field from a conceptual perspective.

This paper aims at filling this gap by focusing on human mobility, from a conceptual and practical point of view, and proposes approaches on how to evaluate XAI methods. This project brings together human mobility and explainable AI algorithms like LIME, Decision Tree, SHAP and Seldon Alibi in the same discussion. Therefore, a conceptual framework is built based on the foundation of the proposed goal methods for explainability. We also introduce models using explanators where these can be evaluated by employing notions and metrics.

The rest of the paper is organized as follows. In Section 2, it compares Classification and Regression projects and defines the concept of four interpretability techniques. XAI methods in human mobility are discussed in Section 3. Then, in Section 4, we discuss the output created by XAI algorithms considering the proposed data. Finally, this paper shows that interpretable AI can be effectively used for future human mobility surveys.

2 Explainable Artificial Intelligence and Human Mobility Research

Based on the literature, Explainable Artificial Intelligence (XAI) is applied in several studies that address classification and regression problems. Many of these works analyze prediction of human tracking, looking at a large population of free-will and autonomous decision-making individuals, or at any event that implies a restriction in mobility. In the following subsections, we indicate relevant researches on human mobility and the challenges and opportunities that they bring to XAI area.

2.1 Classification *versus* Regression problems

Some studies have been conducted using classification algorithms. For example, [5] englobes 5G and 6G and it needs sophisticated AI to automate information delivery simultaneously for mass autonomy, human machine interfacing, and targeted healthcare. The survey analyses the results of three XAI methods, including LIME, SHAP, and CAM. [6] finds that LIME's explanation shows the most influenced image regions on the image classification problem's prediction. In its turn, [15] provides a review on interpretabilities suggested by different research works and categorizes them. The authors compare a set of xAI methods such as Layer-wise Relevance Propagation (LRP), LIME and others to ensure mainly that clinicians and practitioners can subsequently approach these methods with caution and insights into interpretability born with more considerations for medical practices.

On the other hand, regression problems have also been developed. In [12], researchers design hybrid models, combining the expressiveness of opaque models with the clear semantics of transparent models where linear regression is combined with neural networks. Then, [9] develops a ML algorithm using the XGBoost technique, and feature importance, partial dependence plot, and Shap Value is used to increase the model's explanatory potential. [1] provides an entry point for interested researchers and practitioners to learn key aspects of the recent and rapidly growing body of research related to XAI. In short, we provide several illustrative classification and regression examples from various fields, motivating the need for a distinct treatment of both explanation problems respectively.

2.2 XAI Research Methods in Human Mobility

Through the lens of the literature, we review the existing approaches regarding the XAI topic and present trends surrounding it. These elements should be further analysed in human mobility context. In other words, an explainability project that aggregates a discussion human mobility, XAI and their methods should be considered to understand ML algorithms perform so that governments or local authorities understand and solve the diverse pedestrian problems in smart cities. For that reason, we now synthesize and enumerate a set of XAI methods useful for our regression problem:

– Decision Tree - This approach/technique is based on Classification and Regression Trees (CART) which deal with all kinds of variables and predict both numerical and categorical attributes. We use a Decison Tree (DT) API developed by [3].
– LIME - This framework also generates prediction explanations for any classifier or ML regressor. Its main advantage is the ability to explain and interpret the results of models using text, tabular and image data.
– SHAP - This approach is based on a game theory to explain the output of ML models. It provides a means to estimate and demonstrate how each feature's contribution influence the model.
– Seldon Alibi - An open source Python library aimed at ML model inspection and interpretation. Its focus is to provide high-quality implementations of black-box, white-box, local and global explanation methods for classification and regression models.

Based on previous frameworks we have built a transparent, trusted AI to instil trust among our work. They are one of the key requirements for implementing responsible AI in human mobility problems. In the following section we present a solid understanding of how to compute and interpret them in our case study.

3 Analysis of Human Mobility via XAI

In this section, we take a practical hands-on approach, using proposed XAI methods on subsection 2.2. But first of all we outline considerations for analysing aggregated data from several open-source APIs made available by New York City authorities. Through these services we define the dataset for this work and when applied with these methods, we obtain important explanations about human mobility phenomena.

3.1 Data Collection

The data collection of this work is generated via a free public data published by New York agencies and other partners. However, much of this available data is scattered in several API services. In order to aggregate data from these open sources we develop a script in Python language. It uses operations Socrata Open Data API, developed and managed by the Department of Information Technology and Telecommunications (DoITT). Subsequently, we build the Python application taking into account the following sources:

- LinkNYC Kiosk Status - This application provides the most current listing of kiosks, their location, and the status of the Link's WiFi, tablet, and phone;
- 311 Service Requests - The 311 line provides its residents with a resource for assistance and general information outside of emergency situations;
- TLC Trip Record Data - The yellow and green taxi trip records are collected and provided to the NYC Taxi and Limousine Commission by technology providers authorized under the Taxicab & Livery Passenger Enhancement Programs.

We also add the *sentiment* attribute to the generated dataset from NYC APIs. It is calculated from a sentiment analysis process using Valence Aware Dictionary and Sentiment Reasoner (VADER). This "computationally' determines whether a piece of writing is positive (i.e, value 1), neutral (i.e, value 0), or negative (i.e, value -1) based on the *description* or *report* attribute. Additionally, weather information also enriches the original data, and for this purpose we use the OpenWeatherMap API. It allows collecting a vast amount of data associated with weather conditions such as clouds, feels like, humidity, pressure, temperature, temperature minimum, temperature maximum and speed[16].

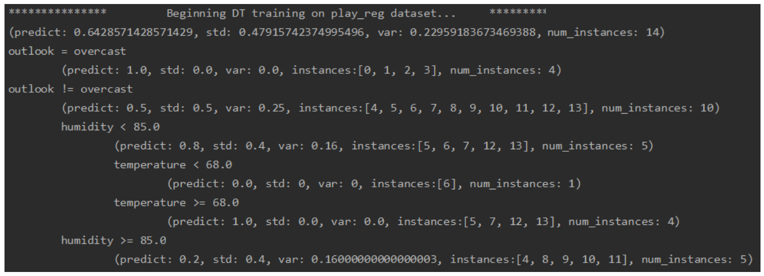
Once the services are synced, data is stored on a PostgreSQL database. Then, we aggregate the rows of original table via SQL query, reducing memory consumption and processing time. Finally, with download via REST API, we have our dataset.

3.2 Exploring XAI methods

As we demonstrated in Subsection 2.2, a set of xAI methods with a mutual case study/task (i.e., census prediction) is defined to analyze and provide meaningful insight on quantifying explainability, and recommend paths towards human mobility. Therefore, we leverage more detailed information about its implementation on our work.

Decision Tree API Inspired by the ML library, scikit-learn has an optimized version of the CART algorithm despite not yet supporting categorical variables [2]. Due to the explanatory potential of CART algorithm algorithm, Miguel Guimarães et al. implements a scratch of an Decision Tree (DT) [4]. The developed DT script is open-source where any programmer can download the source code and modify it, without necessarily depending on the API.

Either through the API or the execution of the DT script, the model training is not allowed if the settings given by the user are invalid. The user is also informed if there are possible improvements to be made in the settings. An example of running DT through the Command-Line Interface (CLI) is shown in Fig. 1.



```
***************         Beginning DT training on play_reg dataset...     *********
(predict: 0.6428571428571429, std: 0.47915742374995496, var: 0.22959183673469388, num_instances: 14)
outlook = overcast
        (predict: 1.0, std: 0.0, var: 0.0, instances:[0, 1, 2, 3], num_instances: 4)
outlook != overcast
        (predict: 0.5, std: 0.5, var: 0.25, instances:[4, 5, 6, 7, 8, 9, 10, 11, 12, 13], num_instances: 10)
        humidity < 85.0
                (predict: 0.8, std: 0.4, var: 0.16, instances:[5, 6, 7, 12, 13], num_instances: 5)
                temperature < 68.0
                        (predict: 0.0, std: 0, var: 0, instances:[6], num_instances: 1)
                temperature >= 68.0
                        (predict: 1.0, std: 0.0, var: 0.0, instances:[5, 7, 12, 13], num_instances: 4)
        humidity >= 85.0
                (predict: 0.2, std: 0.4, var: 0.16000000000000003, instances:[4, 8, 9, 10, 11], num_instances: 5)
```

Fig. 1: The generated Decision Tree from the command line.

LIME This XAI method manipulates the input data and creates a series of artificial data containing only a part of the original attributes [14]. It also provides local model interpretability. Basically, this explainable technique modifies a single data sample by tweaking the feature values and observes the resulting impact on the output. Moreover, its main characteristic is explaining to the dataset level which features are important. It fit the model using sample data points that are similar to the observation being explained. The explanations provided by LIME for each observation x is obtained as Eq. 1:

$$\Phi(x) = argmin_{g \in G} L(f, g, \pi_x) + \Omega(g) \tag{1}$$

where G is the class of potentially interpretable models such as linear models and decision trees, $g \in G$: an explanation considered as a model. $\pi_x(z)$ is proximity measure of an instance z from x. $\Omega(g)$ is a measure of complexity of the explanation $g \in G$.

The goal is to minimize the locality aware loss L without making any assumptions about f, since a key property of LIME is that it is model agnostic. L is the measure of how unfaithful g is in approximating f in the locality defined by $\pi(x)$.

SHAP Shap library is a tool proposed by Lundberg and Lee [11]. It adapts a concept coming from game theory and has many attractive properties. Additionally, we can "debug" our model and observe how it predicted an observation. In the most general form, the Shapley value is the average marginal contribution of a feature value across all possible coalitions. If there are N features, Shapley values will be computed from N different order combinations. From a computational perspective, has shown that the only additive method that satisfies the properties of local accuracy, missingness and consistency is obtained attributing to each variable $x'i$ an effect i defined by Eq. 2:

$$\phi_i(f, x) = \sum_{z' \subseteq x'} \frac{|z'|!(M - |z'| - 1)!}{M!} [f_x(z') - f_x(z' \backslash i)] \tag{2}$$

where f is the model, x are the available variables, and x are the selected variables. The quantity $f_x(z') - f_x(z' \backslash i)$ expresses, for each single prediction, the deviation of Shapley values from their mean: the contribution of the $i - th$ variable.

Seldon Alibi This XAI technique is an open source Python library aimed at Machine Learning (ML) model inspection and interpretation. It provides high-quality implementations of black-box and local explanation methods for regression models. Alibi currently ships with 8 different algorithms for model explanations including popular algorithms like anchors, counterfactuals, integrated gradients, Kernel SHAP, and Tree SHAP. For our work, we choose the Kernel SHAP.

When Alibi is invoked, training data is required and the fit method must be called. Then, an explain method is called to calculate an explanation on an instance or a set of instances. This returns an Explanation object containing dictionaries meta and data with the explanation metadata (e.g. hyperparameter settings, names) and the explanation data respectively. The structure of the Explanation object enables easy serialization in production systems for further processing (e.g. logging, visualization).

4 Results and Discussion

This guide is a practical instruction on how to use and interpret our models explainability. Our first explainable Machine Learning (ML) algorithm is Decision Tree (DT). It uses a linear model, it is also a relatively simple model, and explained by visualizing the tree represented in Fig. 2.

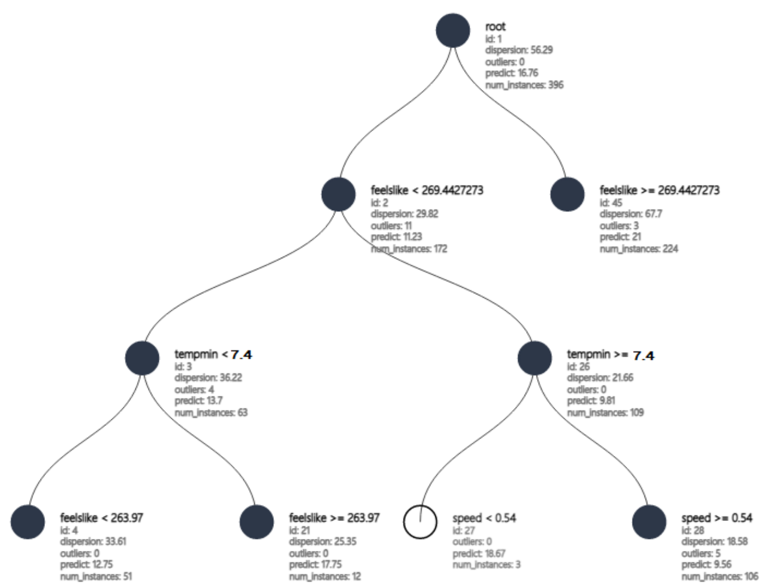


Fig. 2: The Decision Tree explanation.

In this XAI technique, we see a sample forecast path from the root node to the leaf node generated via DT from multivarite model outcome. In root, the DT predicts 16 individuals on NYC center. In the next forecast, with *feelslike* above or equal 269.442, DT predicts a number of 21, but argues the prediction based on the *feelslike* values below 269 with a slight error, 3 attributes (i.e, *feelslike*, *tempmin* and *speed*) are used to model this particular problem. In addition to this explanation, it also generates automatic counterfactual analysis. When the value of *tempmin* attribute is below 7.4(°C), tree gives a tree level and predicts 14 people in New York City center. In case of three level, *feelslike* attribute predicts 13 people, otherwise 18 people. In its turn, when *tempmin* is above or equal 7.4(°C) it also has an impact on census prediction, estimating 10 people. On child nodes, it should be noted that when *speed* value is below 0.54 the prediction of the number of people is 19. In *speed* value above or equal 0.54 the prediction is 10.

In its turn, the key functions in the LIME package are $LimeTabularExplainer()$, which creates an explanation, and $explain\_instance()$, which evaluates explanations. The $explain\_instance$ function requires three arguments: X_test, which specifies the test part of the first sequence (X), and predict function, which specifies the predict using the linear model. An additional, important argument is number features that indicates the maximum number of features present in explanation (K). Additionally, we specify that the number features of explanatory $num\_features$ variables is 3.

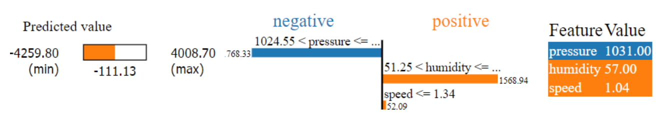


Fig. 3: The LIME explanation.

By applying the $as\_pyplot\_figure()$ to the object containing the explanation, we obtain a graphical presentation of the results. The output includes the colors blue and orange, depicting negative and positive associations, respectively. To interpret the above results, we can conclude that the relative census value (depicted by a bar on the left) depicted by the given test vector (X) can be attributed to (1) the high value of *pressure* feature indicating the less number of people on New York City (NYC) center, (2) the high value of *humidity* feature indicating the high value of the number of people, and (3) the low value of *speed* of wind indicating the high value increase of people in the center.

First column *Predicted Value* computed value corresponds to the number given in the column $model\_prediction$ in the printed output. This value is approximately 111 individuals on NYC center. On the other hand, the *Feature Value* column or $model\_intercept$ column provides of the value of the intercept. It indicates which explanatory variables were given non-zero coefficients in the Linear Regression method. Additionally, it provides information about the values of the original explanatory variables for the observations for which the explanations are calculated.

SHAP framework has a SHAP Explainer that supports any and every ML algorithm. For instance, since we are handling a regression problem, which is bucketed under Linear model, we compute using a Linear Explainer. Thus, we use the $Explainer()$ to build a new explainer for the passed model.

One the fundamental properties of Shapley values is all the input features will always sum up to the difference between expected model output and the current model output for the prediction being explained. Based on an observation, the easiest way to see this is through a waterfall plot that starts our background prior expectation for number of people on New York City center $E[f(X)]$. With features one at a time until we reach the current model output $f(x)$. In other words, this plot shows SHAP values for each of the features. Additionally, it tells how much each of the features have increased or decreased the predicted number of rings for this specific abalone.

Fig. 4 shows expected value of the model output, and then each row shows how the positive (red) or negative (blue) contribution of each feature moves the value from the expected model output over the background dataset to the model output for this prediction.

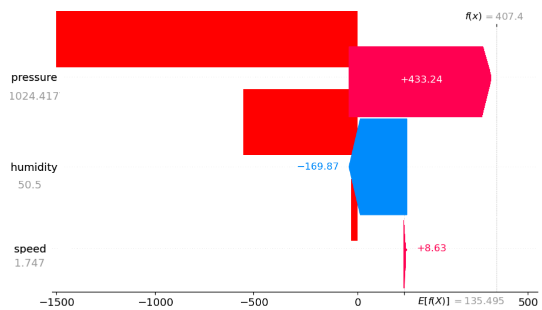


Fig. 4: The SHAP explanation.

Looking at the x-axis, we can see the base value is $E[f(x)] = 135.495$. This is the average predicted number of individuals on NYC center. The ending value is $f(x) = 407.4$. This is the predicted number of pedestrians in NYC center. The SHAP values are all the values in between. For example, the pressure the predicted number of people by 433.24 when compared to the average predicted census. Summarizing, each feature value shows how much each factor contributed to the model's prediction when compared to the mean prediction. Large positive/negative SHAP value indicate that the feature had a significant impact on the model's prediction.

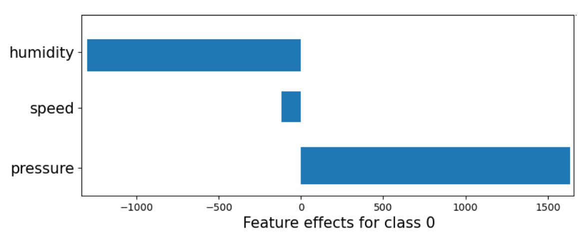


Fig. 5: The Seldon Alibi explanation.

An illustration of the Seldon Alibi explanation is shown in Fig. 5. This framework depicted a model which takes as an input features such as *Humidity*, *Speed* and *Pressure* and outputs a continuous value. We can see for example that the *Humidity* and *Speed* features contribute negatively to this prediction of census (i.e, number of individuals on NYC center) whereas the remainder of the feature have a positive contribution. To explain this particular data point, the *Pressure* feature seems to be the most important.

From the point of view of LIME and SHAP results, for the same model and the same data point, they provide different explanation. Although, they had the strongest impact in the prediction made, in LIME, the *Pressure*, *Humidity* and *Speed* features have a different priority than SHAP feature. But overall their explanations made sense. Unlike its predecessors, DT API simultaneously provides justification and attains with *Pressure* and *Speed* feature high accuracy, exactly opposite of *Humidity* feature. This allow for a new category of accuracy and interpretability. In the other hand, Seldon Alibi showed interesting results with Linear Regression model. Especially the class prototyping method is effective at accelerating counterfactual search, and in this case also improved the quality and plausibility of predictions. Only *Pressure* feature benefits a positive prediction result. However, regardless of the characteristics of previous tools, we consider that each of them are simple and easy to interpret.

5 Conclusions

In this work, we propose a set of Explainable Artificial Intelligence (XAI) methods like Decision Tree (DT), LIME, SHAP and Seldon Alibi. More important than the results that each technique presents, it is important to understand how each technique can be understood by humans, taking into account each of its features. In the initial phase, we presented a short intuition of each method, how to apply them to a dataset and compare the similarities between them and the pros and cons of each method for our problem. It is interesting to know there are increasingly more projects involving human mobility and XAI. This makes our research an important contribution for these areas.

In future work, we plan to introduce more plots to show how each XAI technique contributes positively or negatively to the output of the model. Whilst, DT is visualized from impressive tree, LIME has another plot with a bar chart of local feature importance based on weights derived from Linear Regression. In case of SHAP we can also add a graph with the most relevant features in the highest positions and indicate how they affect the prediction. Lastly, we can introduce a chart that shows how the counterfactual methods change the features to flip the prediction in Seldon Alibi. Finally, another XAI method to explain our problem and support regression approach can be also analyzed such as InterpertML.

Acknowledgments

This work has been supported by FCT - Fundacao para a Ciencia e Tecnologia within the R&D Units Project Scope: UIDB/00319/2020. It has also been supported by national funds through FCT – Fundação para a Ciência e Tecnologia through project UIDB/04728/2020.

References

1. Adadi, A., Berrada, M.: Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI). IEEE Access **6**, 52138–52160 (2018). https://doi.org/10.1109/ACCESS.2018.2870052
2. den Bossche, J.V.: A new categorical encoder for handling categorical features in scikit-learn (2017), `https://jorisvandenbossche.github.io/blog/2017/11/20/categorical-encoder/`
3. Carneiro, D., Silva, F., Guimarães, M., Sousa, D., Novais, P.: Explainable intelligent environments. In: Advances in Intelligent Systems and Computing. vol. 1239 AISC, pp. 34–43 (2021). https://doi.org/10.1007/978-3-030-58356-9_4
4. Carneiro, D., Silva, F., Guimarães, M., Sousa, D., Novais, P.: Explainable intelligent environments. In: Advances in Intelligent Systems and Computing. vol. 1239 AISC, pp. 34–43. Springer Science and Business Media Deutschland GmbH (2021). https://doi.org/10.1007/978-3-030-58356-9_4
5. Guo, W.: Explainable Artificial Intelligence for 6G: Improving Trust between Human and Machine. IEEE Communications Magazine **58**(6), 39–45 (jun 2020). https://doi.org/10.1109/MCOM.001.2000050
6. Hung Quoc Cao, Hung Truong Thanh Nguyen, Khang Vo Thanh Nguyen, P.X.N.: A Novel Explainable Artificial Intelligence Model In Image Classification Problem (2021)
7. Kalyanathaya, K.P., K., K.P.: A Literature Review and Research Agenda on Explainable Artificial Intelligence (XAI). International Journal of Applied Engineering and Management Letters pp. 43–59 (2022). https://doi.org/10.47992/ijaeml.2581.7000.0119
8. Kapadais, K., Varlamis, I., Sardianos, C., Tserpes, K.: A framework for the detection of search and rescue patterns using shapelet classification. Future Internet **11**(9), 192 (sep 2019). https://doi.org/10.3390/fi11090192
9. Lee, Y.: Applying explainable artificial intelligence to develop a model for predicting the supply and demand of teachers by region. Journal of Education and e-Learning Research **8**(2), 198–205 (2021). https://doi.org/10.20448/journal.509.2021.82.198.205
10. Luca, M., Barlacchi, G., Lepri, B., Pappalardo, L.: A Survey on Deep Learning for Human Mobility. ACM Computing Surveys **55**(1), 1–44 (nov 2023). https://doi.org/10.1145/3485125
11. Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. In: Advances in Neural Information Processing Systems. vol. 2017-Decem, pp. 4766–4775. Neural information processing systems foundation (may 2017). https://doi.org/10.48550/arxiv.1705.07874
12. Munkhdalai, L., Munkhdalai, T., Ryu, K.H.: A Locally Adaptive Interpretable Regression (may 2020). https://doi.org/10.48550/arxiv.2005.03350
13. Rai, A.: Explainable AI: from black box to glass box (jan 2020). https://doi.org/10.1007/s11747-019-00710-5
14. Ribeiro, M.T., Singh, S., Guestrin, C.: "why should i trust you?" explaining the predictions of any classifier. In: NAACL-HLT 2016 - 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Demonstrations Session. pp. 97–101. Association for Computational Linguistics (ACL) (feb 2016). https://doi.org/10.18653/v1/n16-3020
15. Tjoa, E., Guan, C.: A Survey on Explainable Artificial Intelligence (XAI): Toward Medical XAI. IEEE Transactions on Neural Networks and Learning Systems **32**(11), 4793–4813 (2021). https://doi.org/10.1109/TNNLS.2020.3027314
16. Weather Underground: OpenWeatherMap API (2018), `https://www.wunderground.com/weather/api`