

Contextualized LORE for Fuzzy Attributes

Najlaa MAAROOF^{a,1}, Antonio MORENO^a, Mohammed JABREEL^b and Aida VALLS^a

^a *ITAKA-Intelligent Technologies for Advanced Knowledge Acquisition - Departament d'Enginyeria Informàtica i Matemàtiques, Universitat Rovira i Virgili, Tarragona, Spain*

^b *Microsoft Advanced Technology Lab, Cairo, Egypt*

Abstract. Despite the broad adoption of Machine Learning models in many domains, they remain mostly black boxes. There is a pressing need to ensure Machine Learning models that are interpretable, so that designers and users can understand the reasons behind their predictions. In this work, we propose a new method called *C-LORE-F* to explain the decisions of fuzzy-based black box models. This new method uses some contextual information about the attributes as well as the knowledge of the fuzzy sets associated to the linguistic labels of the fuzzy attributes to provide actionable explanations. The experimental results on three datasets reveal the effectiveness of *C-LORE-F* when compared with the most relevant related works.

Keywords. Machine Learning, Fuzzy Logic, Explanation AI

1. Introduction

The past decade has witnessed significant improvements in the development of Machine Learning (ML)-based systems, due to the presence of huge amounts of data, the advances in Deep Learning and the affordability of more advanced computer equipment. As a result, ML has become a vital component of multiple applications in many fields. However, most ML-based systems are considered black boxes, and it is not straightforward to understand the reasons behind their decisions. Hence, developing methods for interpreting the decisions of such systems has become highly demanded [1, 2].

Explainability mechanisms for complex ML models can be *model-dependent* or *model-agnostic*. The former analyse the internal structure of the model (e.g. the weights inside a neural network) to understand their internal working and come up with a clear picture of how the solution is computed. The latter usually generate a set of inputs, analyse the answers provided by the black box to them, and then create a simpler model from which we can infer an explanation [3–5].

¹Corresponding Author: Najlaa MaarooF. E-mail: najlaamaarooFwahib.al-ziyadi@urv.cat.

Two well-known post-hoc explanation methods are *Local Interpretable Model-agnostic Explanations* (LIME, [5]) and *Local Rule-Based Explanations* (LORE, [6]). LIME samples points randomly from the same distribution of the point to be explained, sends them to the black box, and provides local explanations for the classifier's prediction by fitting a linear regression model locally around that point. LORE generates a set of neighbours of the input point using a genetic algorithm, applies the black-box on each of them, and builds a decision tree on these results. The LORE explanation contains the rule used to produce the decision and a set of *counterfactual rules* which represent the minimal number of changes in the feature values of the instance that would change the conclusion of the system.

Despite the popularity of LORE and LIME, they have some shortcomings. First, they do not use any knowledge about the characteristics of the input features. Secondly, the neighbour generation algorithms generate "close" points more or less blindly. In our previous work [7] we proposed *Guided-LORE* (or *G-LORE*), a version of LORE in which the neighbourhood generation is formalised as a search problem and solved using Uniform Cost Search, making it more focused. However, like LIME and LORE, Guided-LORE does not consider explicitly the case in which the attributes that define the objects are fuzzy. For example, when we use Guided-LORE, a neighbour of a point is generated by adding (or subtracting) a fixed amount (which is called *step*) to the value of an attribute. In the case of using fuzzy attributes this option could be reasonable if the fuzzy sets associated with the linguistic labels were equally and uniformly distributed across all the domain. However, in many situations, this property does not hold.

This paper proposes *C-LORE-F* (Contextualized LORE for Fuzzy attributes), a variant of Guided-LORE that addresses those issues. Our first motivation is that, if we know that an attribute is fuzzy and we have the information on its fuzzy labels and their associated fuzzy sets, we can profit from that knowledge to make a more focused neighbourhood generation. More precisely, we can generalise its step from being a fixed value to being a function that depends on that knowledge. In that way the proposed method is more general, and it works in the cases in which the fuzzy sets associated with the linguistic labels are uniformly or non-uniformly distributed. To the best of our knowledge, this work is the first one that utilises such knowledge to develop explanation methods for ML systems based on fuzzy logic. The second novel point of the system is the use of the knowledge about the type of attribute to guide the neighbourhood generation process and search for *actionable* explanations. For example, if we have an attribute like age, which automatically increases in time, it probably does not make much sense to look for neighbours that have a lower value in this attribute, as it would not be very interesting for the user to receive an explanation like "if you were 10 years younger, the prediction of the system would be different" (even if this explanation was correct). This knowledge about the type of attribute is not currently being used in the most popular explanation systems.

The rest of this article is structured as follows. Section 2 provides an overview of the related works. Section 3 explains the proposed neighbourhood generation procedure and the types of attributes that have been considered. In Section 4 we describe the experimental setup and discuss the obtained results. Finally, in section 5, we conclude the paper and list some points for future work.

2. Related Work

In the last years the research of methods for explaining black box decision systems has received a lot of attention [8], and there has been an extraordinary amount of articles in ML interpretability in the last years. This section comments some of the most relevant ones to this work. We can categorise the works on ML interpretability into those based on features' importance (§2.1), counterfactual examples (§2.2) and visualisation mechanisms (§2.3).

2.1. Feature relevance

There are two main directions for developing explanation methods based on the importance of features, *global* and *local* explanation methods. Global methods try to explain the entire model behaviour using surrogate models whereas local models explain a single prediction. It can be useful, in some scenarios, to understand the global logic of a model. However, the major issue with such approaches is that, as the explanations are extracted from simpler surrogate models, there is no guarantee that they are faithful to the original model [3, 5, 9].

Local explanation methods are considered to be one of the fundamental approaches to post-hoc explanations. *LIME* [5], already mentioned in the introduction, is a well-known example of such approaches. It is independent of the type of data and the black box to be explained. Given a black box model f , an instance x , and a decision y produced by f on x , LIME constructs a simple linear model that approximates f 's input-output behaviour to justify why f predicts y . It generates some neighbours of x randomly in the feature space, that are centred on x . Such an approach is becoming a conventional method. We can find now LIME implementations in multiple popular packages, including Python, R and SAS.

The authors of LIME observed that it does not measure its fidelity. As a result, the local behaviour of a notably non-linear model may lead to faulty linear approximations. Hence, they were motivated to work on a new model-agnostic method, *Anchors*, based on if-then rules [10]. This method highlights the part of the input that is adequate for the classifier to make the prediction, delivering more intuitive and easy-to-understand explanations.

SHAP [11] is a method that provides an explanation of the prediction of the output of a black box for an instance x by estimating each feature's contribution to that prediction. These contributions are collected by measuring the *Shapley values* from coalitional game theory. The features act like players in a coalition. Each player can be formed by a single feature or a subset of features. The Shapley values show the payout distribution of the prediction among the features.

2.2. Counterfactual examples

Another important category of explanations is based on the generation of counterfactuals. These methods seek minimal changes to the feature values such that the model's predicted outcome changes. These changes should actually be actionable to be useful (e.g. an applicant for a bank loan might want to know which part of her application could be changed to get her application approved).

The pioneering work by Martens and Provost [12] used a best-first search to develop a model-agnostic method for finding counterfactuals that explain the predictions of any classifier. LORE [6] is another example of this approach, although it can also be seen as an example of the determination of features' importance. It constructs a decision tree c based on a synthetic neighbourhood of the input point. Then, an explanation e , composed of a decision rule and a set of counterfactual rules, based on some extracted counterfactual examples, is obtained from the logic of c . The work presented in [13] proposed a general optimisation framework to generate sets of diverse counterfactual examples for any differentiable ML classifier. Russell [14] proposed in 2019 a "mixed polytope", a set of constraints that can be used with integer programming solvers to extract counterfactual explanations without making a brute-force enumeration.

2.3. Methods focused on visualisation

We can find several visualisation methods proposed in the literature to help ML engineers and domain experts to understand, debug, and refine ML models. For example, Ming et al. [15] proposed an interactive visualisation method to help users, even those without expertise in Machine Learning, to understand, explore and confirm predictive models. This method by Ming et al. extracts a set of rules that approximates a classifier's prediction and visualises them using an interactive visual interface. Neto and Paulovich proposed Explainable Matrix (ExMatrix) [16], a visualisation method to interpret Random Forests. They used a matrix-like visual metaphor in which rows represent rules, columns are features, and cells are rules predicates. They showed that their method is capable of offering global and local explanations of Random Forest models.

Like LORE and Guided-LORE, our proposal can be considered as a hybrid of the approaches based on features' importance and counterfactual examples. However, unlike them, it focuses on the case in which the attributes that define the objects are fuzzy.

3. Contextualised LORE for Fuzzy attributes

C-LORE-F uses contextual information (the type of attribute and the fuzzy sets associated to the linguistic values of the fuzzy attributes) to produce explanations. Its inputs are a trained fuzzy-based ML model, f , and an example x_0 . First, we apply f to x_0 to get a decision y_0 . Then, we apply *C-LORE-F* to generate an explanation, which is composed of a rule r and a set of counterfactual rules δ that produce a different outcome. To this end, the general LORE process is used:

1. Generate a set of neighbours \mathcal{G} of x_0 .
2. Train a decision tree t using \mathcal{G} .
3. Inspect t and extract the rule r used to classify x_0 ,
4. Generate a set of counterfactual examples to x_0 , pass them to t to get their labels and get the set of counterfactual rules δ .

Algorithm 1: Neighbours Generator

```

input : An example  $x_0$ , a black-box model  $f$ , the maximum level  $L$ , and the set of attributes  $\mathcal{A}$ .
output: The set of neighbours,  $N$ .
1  $root \leftarrow node(x_0, NULL, 0)$ ;
2  $root.label = y_0 \leftarrow f(x_0)$ ;  $q \leftarrow [root]$ ;  $N \leftarrow []$ ;
3 while Not need to stop do
4    $n \leftarrow head[q]$ ;
5   if  $n.label = root.label$  and  $n.level \leq L$  then
6     foreach attribute  $\alpha \in \mathcal{A}$  do
7       if  $\alpha$  is Fixed then
8         continue;
9       else
10        if  $\alpha$  can increase then
11           $x_l \leftarrow clone(n.x)$ ;
12           $x_l[\alpha] \leftarrow next(n.x[\alpha], \alpha)$ ;
13           $n_l \leftarrow node(x_l, n, n.level + 1)$ ;
14           $n_l.label \leftarrow f(x_l)$ ;
15           $n_l.d \leftarrow HVDM(x_l, x_0)$ ;
16          add  $n_l$  to  $q$  and  $N$ ;
17        end
18        if  $\alpha$  can decrease then
19           $x_r \leftarrow clone(n.x)$ ;
20           $x_r[\alpha] \leftarrow prev(n.x[\alpha], \alpha)$ ;
21           $n_r \leftarrow node(x_r, n, n.level + 1)$ ;
22           $n_r.label \leftarrow f(x_r)$ ;
23           $n_r.d \leftarrow HVDM(x_r, x_0)$ ;
24          add  $n_r$  to  $q$  and  $N$ ;
25        end
26      end
27    end
28  end
29 end

```

Figure 1. Neighbours Generator.

For more details about the counterfactual examples generation and rules extraction, we refer the reader to [6]. The set \mathcal{G} is obtained by merging two subsets, \mathcal{G}^+ and \mathcal{G}^- . The first one is called the *positive set*, and it contains a set of instances that belong to the same class of x_0 . We get this subset by passing the instance x_0 to Algorithm 1. The second one, the *negative set*, contains examples with a different class. We obtain \mathcal{G}^- by looking at an auxiliary set T and finding the closest example to x_0 , i.e., x_0^- , that has a different label than y_0 . T can be the training set used to train the black-box model, if accessible, or any other data set from the same distribution. Once we get x_0^- , we pass it to Algorithm 1 to generate the *negative set*.

The *Neighbours Generation* step aims to find the set $\hat{\mathcal{G}}$ with the points that are close to a given instance \hat{x} and have the same class. $\hat{\mathcal{G}}$ can be either \mathcal{G}^+ if $\hat{x} = x_i$ or \mathcal{G}^- if $\hat{x} = x_i^-$.

As a first change with respect to LORE and Guided-LORE, we have defined the following types of attributes.

- Attributes with a fixed value (e.g. sex).
- Attributes whose value increases in time (e.g. age).
- Attributes whose value decreases in time (e.g. years left until retirement).
- Variable attributes, that can change positively and negatively (e.g. weight).

Neighbourhood generation is defined as a search problem in which we explore the neighbourhood space of a point $x_0 = \hat{x}$ by applying a Uniform Cost Search based on the Heterogeneous Value Difference Metric (HVDM, [17]), using some contextual information about the attributes (the attribute range and the step value). This search problem can be formulated as follows:

- **State Space:** the set of all possible examples S .
- **Initial State:** (x_0, y_0) , where x_0 is the instance of which we want to generate its neighbours and y_0 is the label of this instance obtained by the black-box f .
- **Actions:** Modifications of the value of a single attribute (feature). These actions leverage some contextual information about the feature to make the desired changes to generate new neighbours. In our case we define two types of actions, *next* and *prev*, described later.
- **Transition Model:** returns a new instance in which the value of a feature is changed by applying all actions.
- **Goal Test:** We check, for each generated individual, if, according to the black box, it has the same label as the root, y_0 . If that is the case, we generate its neighbours in the same way (i.e. applying one positive/negative change in the value of a single attribute). Otherwise, we have found an individual close to x_0 that belongs to another class; thus, we have reached a boundary of y_0 , and we terminate the search from that instance.
- **Path Cost:** The path cost of each example is calculated by measuring the HVDM distance between the generated example and x_0 .

Algorithm 1 shows how the neighbours of a given instance, x_0 , are generated. The search tree starts in x_0 , and in each node all the possible actions to move from one instance to another are applied. For each feature f , the number of possible actions can be zero (f is *Fixed*), one (either *next* if the feature is temporally increasing or *prev* if it is temporally decreasing) or both, if f is variable. Each action only changes the value of one feature. The candidate node to be expanded, n , is the one closest to x_0 , based on the path cost. If the outcome of the black-box model for n is different from y_0 , then it is a leaf of the tree. Otherwise, we expand that node. Consequently, for each node in the second level, we would have changes in two attributes or double changes in the same attribute, and so on. The expanding process finishes when we reach a predefined max-level, or when there are no more nodes to be expanded (all the leaves have led to changes in the initial classification). Repeated instances are ignored to avoid cycles.

The expanding process is done by cloning the instance of the current node, i.e., $n.x$ (lines 11 and 19 in Algorithm 1) and applying the *next* and/or *prev* actions. After that, we pass the obtained instance to the black box model f to get its corresponding label. To apply the actions *step* and *prev* for a given attribute we consider some *separate zones* based on its fuzzy sets, which are defined as shown in Figure 2, taking into account the intersection point between two consecutive fuzzy sets and the intervals of maximum activation. In Figure 2 the zones would be 0-5, 5-10, 10-15, 15-20, 20-25, 25-40, 40-50, 50-60, 60-75, 75-90 and 90-100. Given the value of the attribute, we locate its zone, and then we take the middle of the previous zone as the lower neighbour (the result of the *prev* action), and

the middle of the next zone as the upper neighbour (the result of the *next* action). Figure 2 shows an example. The input value is 22, which belongs to the zone 20-25. Thus, the middle of the previous zone is the lower neighbour, $(15 + 20)/2 = 17.5$, and the middle of the next zone is the upper neighbour, $(25 + 40)/2 = 32.5$. We might end up applying only either the *next* action, if the located zone was the first one, or the *prev* action, if it was the last one.

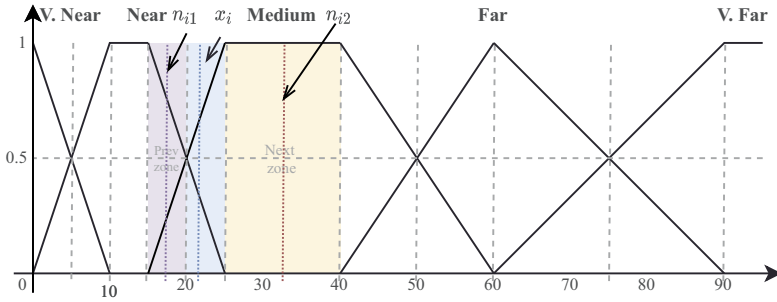


Figure 2. Illustration of the *next* and *prev* actions.

4. Experiments and Results

4.1. Experimental Setup

We used three data sets in our experiments: German-Credit, Adult-Income and Diabetic-Retinopathy. The first two ones are publicly available in the well-known UCI Machine Learning Repository, whilst the last one is a private data set for the assessment of the risk of developing diabetic retinopathy (DR) for diabetic patients. All of them are examples of binary classification. Considering the public data sets, as in [6], each data set was randomly split into a training set with 80% instances, and a test set, i.e., the set of instances for which the black box decision has to be explained, with 20% instances. The black box predictors used in the test were Fuzzy Random Forest (FRF) and Fuzzy Decision Tree (FDT). In case of Diabetic-Retinopathy, we directly used our fuzzy random forest-based system, Retiprogram, that is currently being used in the Hospital de Sant Joan in Reus (Tarragona). Table 1 illustrates the number of features and the number of training and testing examples used in the test for each data set. It also shows the accuracy scores of the black-box models for each data set.

Table 1. Data sets used in the experiments.

	Features	Train	Test	Total	Acc. FRF	Acc. FDT
Adult-Income	8	39,074	9768	48,842	0.781	0.654
German-Credit	20	8,000	2,000	10,000	0.715	0.715
Diabetic-Retinopathy	9	1,212	1,111	2,323	0.804	0.781

4.2. Evaluation Metrics

We used the following evaluation metrics:

- **hit**: this metric computes the accuracy between the output of the decision tree t and the black box model f for the testing instances. It returns 1 if $t(x)$ is equal to $f(x)$ and 0 otherwise.
- **fidelity**: this metric measures to which extent the decision tree accurately reproduces the black-box predictor, by comparing its predictions and the ones of the black-box on the instances \mathcal{G} .
- **l-fidelity**: it is similar to the fidelity, but it is computed on the instances covered by a decision rule in a local explanation for x . It is used to measure to what extent the rule is good at mimicking the black-box model.
- **c-hit**: it compares the predictions of the decision tree and the black-box model on a counterfactual instance of x that is extracted from the counterfactual rules in a local explanation of x .
- **cl-fidelity**: it is also similar to the fidelity, but it is computed on the instances covered by the counterfactual rules in a local explanation for x .

4.3. Results and Discussion

Table 2 shows the means and standard deviations of the metrics for C-LORE-F, LORE and G-LORE on the three data sets with the FRF and FDT models. In general, C-LORE-F outperforms the other methods in all metrics with the FRF model. In the case of FDT, it shows better performance than LORE and G-LORE in *hit* and *fidelity*. In the other metrics it is very close to the best one. LORE is the worst in most cases, especially with the FRF model. The reason is that G-LORE and C-LORE-F try to find the closest "frontier" between the class of x_0 and the other classes, producing a clearer decision boundary.

Table 2. The results on the three datasets.

Model	Method	<i>hit</i>	<i>fidelity</i>	<i>l-fidelity</i>	<i>cl-fidelity</i>	<i>c-hit</i>
FRF	LORE	0.96±0.19	0.98±0.02	0.97±0.07	0.45±0.43	0.43±0.39
	G-LORE	0.99±0.02	0.99±0.02	0.99±0.03	0.52±0.43	0.47±0.40
	C-LORE-F	1.00±0.0	0.99±0.0	0.99±0.0	0.59±0.39	0.58±0.42
FDT	LORE	0.95±0.22	0.98±0.03	0.98±0.03	0.48±0.41	0.45±0.44
	G-LORE	0.98±0.10	0.98±0.01	0.85±0.24	0.54±0.45	0.50±0.48
	C-LORE-F	0.99±0.05	0.99±0.0	0.97±0.08	0.43±0.43	0.41±0.46

Focusing on the black-box dimensions, all the methods show a better performance with the FRF in the *hit*, *fidelity* and *l-fidelity* metrics. This can lead us to conclude that the accuracy of a model is crucial in getting a better explanation.

At the data sets level, as shown in Figure 3, the best performance is obtained by Diabetic-Retinopathy, followed by Adult-Income. The reason is that all the explanation methods are sensitive to the accuracy of the black-box model. The more accurate is the model, the best is the obtained explanation, as confirmed by the *accuracy* scores reported in Table 1. In terms of *c-hit* and *cl-fidelity*, the best results are obtained with the Diabetic-Retinopathy data set. We can attribute

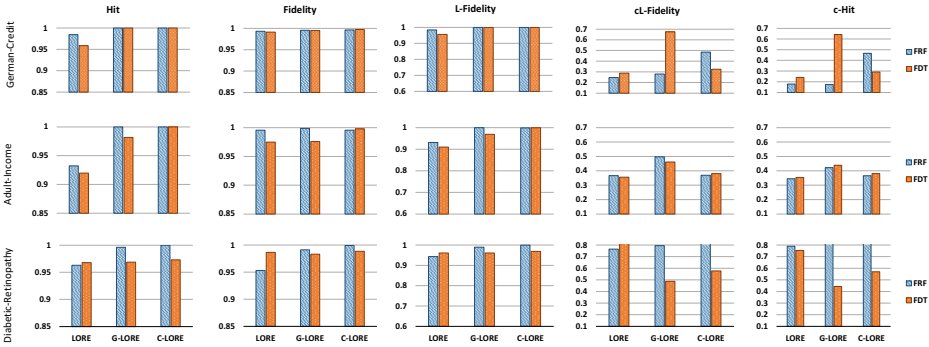


Figure 3. Comparison results: LORE vs G-LORE vs C-LORE-F.

this fact to the quality design of the fuzzy sets in this problem. The fuzzy sets of the Diabetic-Retinopathy data set were defined by an expert of the domain, whereas the fuzzy sets of Adult-Income and German-Credit were obtained automatically by applying a fuzzification algorithm [18]. The argument here is that these two metrics rely on the quality of the counterfactual examples that are used to generate counterfactual rules (which may be affected by the generated neighbours). Moreover, the intelligent design of the fuzzy sets is also a key factor in C-LORE-F as it utilises them as contextual information in the neighbourhood generation process. This can be confirmed by comparing the results of C-LORE-F vs others on the Diabetic-Retinopathy data set and comparing the performance of C-LORE-F method on the Diabetic-Retinopathy data set vs the other data sets. C-LORE-F outperforms LORE and G-LORE in almost all evaluation metrics. The cl-fidelity and c-hit are exceptions with the FDT and German-Credit case. In general, all the explanation methods showed a poor performance in terms of cl-fidelity and c-hit. That may be due to the bad quality of the counterfactual examples, and we intend to analyse this issue in our future work.

5. Conclusion

This paper has presented C-LORE-F, a new method to explain the decisions of fuzzy-based systems, that uses the information about the fuzzy sets that define the meaning of the linguistic values of the fuzzy attributes. It also considers the character of the attribute (whether its value is fixed, increasing, decreasing or variable). Its main advantage is that the generation of neighbours for a point x is more informed due to the usage of contextual information. Moreover, we search for boundaries with relevant meanings for the user (e.g. in order to avoid creating counterfactuals that depend on the change of a fixed attribute, or on the positive change of an attribute that only decreases on time). The experimental results on different data sets demonstrate the effectiveness of the proposed method. It outperformed the state-of-the-art methods in several metrics. The main issue in the proposed method is its poor performance on the c-hit and cl-fidelity metrics, although it showed a performance comparable to the best one. In our future work, we will focus on resolving this issue by improving the counterfactual examples

generation. We should study the relationships between attributes to avoid generating impossible instances. Moreover, we intend to study how we can extract the explanations directly from fuzzy decision trees and use them as surrogate models.

Acknowledgements

This work has been funded by the research project PI18/00169 from Instituto de Salud Carlos III & FEDER funds and URV grant 2019PFR-B2-61. The first author is supported by a URV Martí Franquès predoctoral grant.

References

- [1] Adadi A, Berrada M. Peeking inside the black-box: a survey on explainable artificial intelligence (XAI). IEEE access. 2018;6:52138–52160.
- [2] Carvalho DV, Pereira EM, Cardoso JS. Machine learning interpretability: A survey on methods and metrics. Electronics. 2019;8(8):832.
- [3] Burkart N, Huber MF. A Survey on the Explainability of Supervised Machine Learning. Journal of Artificial Intelligence Research. 2021;70:245–317.
- [4] Krause J, Perer A, Ng K. Interacting with predictions: Visual inspection of black-box machine learning models. In: Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems; 2016. p. 5686–5697.
- [5] Ribeiro MT, Singh S, Guestrin C. 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. p. 1135–1144.
- [6] Guidotti R, Monreale A, Ruggieri S, Pedreschi D, Turini F, Giannotti F. Local rule-based explanations of black box decision systems. arXiv preprint arXiv:180510820. 2018;.
- [7] Maarooif N, Moreno A, Valls A, Jabreel M. Guided-LORE: Improving LORE with a Focused Search of Neighbours. In: Heintz F, Milano M, O’Sullivan B, editors. Trustworthy AI - Integrating Learning, Optimization and Reasoning. Springer; 2021. p. 49–62.
- [8] Guidotti R, Monreale A, Ruggieri S, Turini F, Giannotti F, Pedreschi D. A survey of methods for explaining black box models. ACM computing surveys. 2018;51(5):1–42.
- [9] Molnar C. Interpretable Machine Learning. Lulu.com; 2020.
- [10] Ribeiro MT, Singh S, Guestrin C. Anchors: High-Precision Model-Agnostic Explanations. In: AAAI. vol. 18; 2018. p. 1527–1535.
- [11] Lundberg SM, Lee SI. A unified approach to interpreting model predictions. In: Advances in neural information processing systems; 2017. p. 4765–4774.
- [12] Martens D, Provost F. Explaining data-driven document classifications. Mis Quarterly. 2014;38(1):73–100.
- [13] Mothilal RK, Sharma A, Tan C. Explaining machine learning classifiers through diverse counterfactual explanations. In: Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency; 2020. p. 607–617.
- [14] Russell C. Efficient search for diverse coherent explanations. In: Proceedings of the Conference on Fairness, Accountability, and Transparency; 2019. p. 20–28.
- [15] Ming Y, Qu H, Bertini E. Rulematrix: Visualizing and understanding classifiers with rules. IEEE transactions on visualization and computer graphics. 2018;25(1):342–352.
- [16] Neto MP, Paulovich FV. Explainable MatrixVisualization for Global and Local Interpretability of Random Forest Classification Ensembles. IEEE Transactions on Visualization and Computer Graphics. 2020;.
- [17] Wilson DR, Martinez TR. Improved heterogeneous distance functions. Journal of artificial intelligence research. 1997;6:1–34.
- [18] Yuan Y, Shaw MJ. Induction of fuzzy decision trees. Fuzzy Sets and systems. 1995;69(2):125–139.