

“Even if ...” – Diverse Semifactual Explanations of Reject

1st André Artelt
CITEC - Faculty of Technology
Bielefeld University
Bielefeld, Germany
aartelt@techfak.uni-bielefeld.de

2nd Barbara Hammer
CITEC - Faculty of Technology
Bielefeld University
Bielefeld, Germany
bhammer@techfak.uni-bielefeld.de

Abstract—Machine learning based decision making systems applied in safety critical areas require reliable high certainty predictions. For this purpose, the system can be extended by an reject option which allows the system to reject inputs where only a prediction with an unacceptably low certainty would be possible. While being able to reject uncertain samples is important, it is also of importance to be able to explain why a particular sample was rejected. With the ongoing rise of eXplainable AI (XAI), a lot of explanation methodologies for machine learning based systems have been developed – explaining reject options, however, is still a novel field where only very little prior work exists.

In this work, we propose to explain rejects by *semifactual explanations*, an instance of example-based explanation methods, which them self have not been widely considered in the XAI community yet. We propose a conceptual modeling of semifactual explanations for arbitrary reject options and empirically evaluate a specific implementation on a conformal prediction based reject option.

Index Terms—XAI, Semifactual Explanations, Reject Options

I. INTRODUCTION

The usage of machine learning (ML) based systems in safety critical areas such as autonomous driving [1] requires high certainty predictions – i.e. making mistakes must be avoided at all costs because those might have serious consequences. However, most ML models do not provide proper and reliable certainty scores of their predictions, and the models usually always output a prediction no matter how certain this prediction and how reasonable the input is. A potential remedy is to extend ML models with *reject options* [2]. Reject options allow the model to reject inputs (i.e. refuse to make a prediction) where the model is not certain in its prediction – by this it can be ensured that we only make high certainty predictions and avoid making serious mistakes. For instance consider the example of a phishing mail filter: *Imagine a mail filtering application that tries to filter out phishing mails in order to protect the end users and their surrounding from serious consequences. The filter is supposed to automatically sort out mails where it is certain that the particular mails are malicious, and pass all benign mails to the user without any delay. However, in cases where the*

filter is not absolute certain about its prediction (distinguishing benign vs. malicious), it should reject to classify this mail – such rejected mails might be passed to the user with an additional warning of taking care or to the IT-security department of the company for further investigations and possible improvement of the filtering application.

Because of the increasing usage of ML based systems in practice, transparency of such systems is nowadays a widely accepted standard requirement which also made its way into legal regulations like the EU’s GDPR [3]. Transparency is usually realized by means of explanations – i.e. explanations of the system’s behavior are provided to the user [4]. Although it is still not perfectly understood what exactly makes up “good” explanations [5], [6], there have been developed a wide variety of different explanation methods [4], [7], [8]: Feature relevance/importance methods [9] and examples based methods [10]. Instances of example based methods are contrasting explanations like counterfactual explanations [11], [12] and prototypes & criticisms [13] – these methods use a set or a single example for explaining the behavior of the system.

Besides being able to reject inputs that lead to low-certainty predictions, it is also important to be able to explain why some inputs got rejected while others got accepted – in order to trust the reject option, the user has to get an understanding on how this particular reject option works. E.g. coming back to the previous example: *If the filtering application rejects to classify a mail as either benign or malicious, it would be beneficial to get an explanation why the filtering application was not able to come up with a high certainty prediction – in particular when it comes to understand the reasons for this behavior and potential improvements of the filtering application.*

Related Work: Despite the obvious importance of explaining reject options in ML systems, surprisingly little work exists on this topic:

The authors of [14] propose to use counterfactual explanations for explaining reject options of learning vector quantization models – i.e. there approach is limited to this particular type of model and reject option. Furthermore, only a single explanation is computed – note that according to the Rashomon effect there might exist more than one possible explanation, and therefore a lot of information might be lost by just computing and presenting a single explanation.

We gratefully acknowledge funding from the VW-Foundation for the project *IMPACT* funded in the frame of the funding line *AI and its Implications for Future Society*.

André Artelt is also affiliated with KIOS Research and Innovation Center of Excellence, University of Cyprus, Nicosia, Cyprus.

Another work [15] proposes a model agnostic approach for locally explaining arbitrary reject options of arbitrary systems – they propose to use a local and interpretable approximation (e.g. a linear model or decision tree) to locally explain the reject option. However, a local approximation might not always capture the entire behavior of the global function and it is also debatable whether a linear model or a decision tree is always truly “interpretable”.

Our Contributions: Our contributions in this work is of conceptional nature: We propose a model agnostic method for explaining reject options by means of a set of diverse semifactual explanations – we propose the abstract concept, a formalization of this concept, and an implementation of it. We not only consider an explanation method that has not been considered before for explaining reject options, we also compute a set of diverse explanation to maximize the amount of information that is presented to the user – i.e. dealing with the “Rashomon effect” which states that there often exist more than one possible explanation.

The remainder of this work is structured as follows: First, we introduce the necessary foundations of this work: we review reject options in Section II-A and semifactual explanations in Section II-B. Next in Section III, we introduce our model agnostic modeling for computing diverse semifactual explanations of reject options. We empirically evaluate quantitative aspects of our proposed diverse semifactual explanations of reject options on several standard benchmark data sets in Section IV. Finally, this work closes with a conclusion in Section V.

II. FOUNDATIONS

A. Reject Options

For a given prediction function $h : \mathcal{X} \rightarrow \mathcal{Y}$, a reject option [2] is usually added by means of a certainty function $r_h : \mathcal{X} \rightarrow \mathbb{R}_+$ that measures the certainty of the prediction $h(\vec{x})$ – we reject a sample \vec{x} if the certainty is below a given threshold θ :

$$r_h(\vec{x}) < \theta \quad (1)$$

where the subscript h denotes a potential dependency on the prediction function $h(\cdot)$.

We can think about adding a reject option to $h(\cdot)$ as constructing a new function $h' : \mathcal{X} \rightarrow \mathcal{Y} \cup \{\infty\}$ where we add a reject symbol ∞ to the set of possible outputs \mathcal{Y} :

$$h'(\vec{x}) = \begin{cases} h(\vec{x}) & \text{if } r_h(\vec{x}) \geq \theta \\ \infty & \text{otherwise} \end{cases} \quad (2)$$

There exist a lot of different implementation of certainty based reject options Eq. (1) [2]. In this work, we use the same conformal prediction based reject option [16] as it was done in [15]: Assume a (black-box) probabilistic classifier $h : \mathcal{X} \rightarrow \mathcal{Y}$ $h(\vec{x}) = \arg \max_{y \in \mathcal{Y}} p(y | \vec{x})$, where $p(y | \vec{x})$ denotes the class wise probability as estimated by the classifier $h(\cdot)$. The so called non-conformity measure $\phi_h : \mathcal{X}, \mathcal{Y} \rightarrow \mathbb{R}$ – which measures how different a given labeled sample is from a given set of labeled samples we have seen before – is a central

building block of a conformal predictor [17]. In case of a probabilistic classifier $h(\cdot)$, a common non-conformity measure is given as follows: $\phi_h(\vec{x}, y = j) = \max_{i \neq j} p_h(y = i | \vec{x}) - p_h(y = j | \vec{x})$. For fitting a conformal predictor based on $h(\cdot)$, we need another labeled data set $\mathcal{D}_{\text{calib}} \subset \mathcal{X} \times \mathcal{Y}$ (called calibration set) which was not used during the fitting of $h(\cdot)$ – we compute the non-conformity α_i of every sample from the calibration set by applying $\phi_h(\cdot)$: $\alpha_i = \phi_h(\vec{x}_i, y = y_i)$.

For every new data point $\vec{x}_* \in \mathcal{X}$ that has to be classified, we compute the non-conformity measure for every possible label in \mathcal{Y} . Next, the non-conformity scores of \vec{x}_* are compared with the non-conformity scores from the calibration set to compute p-values $p_{y=i}(\cdot)$ for every possible classification of \vec{x}_* . The conformal predictor then selects the label with the largest p-value as a prediction $h(\vec{x}_*) = \arg \max_{i \in \mathcal{Y}} p_{y=i}(\vec{x}_*)$. The credibility of the prediction – i.e. how well the training set supports the prediction – is given by the largest p-value: $\psi(\vec{x}_*) = \max_i p_{y=i}(\vec{x}_*)$.

We implement a reject option Eq. (1) [16] using the credibility score:

$$r_h(\vec{x}) = \psi(\vec{x}) = \max_i p_{y=i}(\vec{x}) \quad (3)$$

B. Semifactual Explanations

A semifactual explanation (often just called *semifactual*) states some changes that even if they had been applied to the input, would *not change* the models behavior/prediction. We can think of a semifactual explanation as an “Even if...” explanation. Consider the popular example of local application [4], [18]. *Imagine that a loan application was rejected and we want to know why. A semifactual explanation could be that the application would have been still rejected, even if the applicant had earned 500\$ more per month and even if they had not have a second credit card.*

Note that in contrast to contrasting explanations like counterfactual explanations [11], a semifactual does not recommend actions how to change the system’s behavior – this is what a contrasting explanation would do – but it rather tells the user what changes would not change the system’s behavior. A semifactual can be interpreted in at least two different ways: A semifactual tries to convince the user that the system’s behavior can not be changed (easily) – this could be kind of an “evil” use because it creates the impression that there is no way that the system’s behavior could be changed and therefore the user must simply accept it. Another interpretation could be that a semifactual tells the user where not to put effort into because it does not help to change the system’s behavior – for improved usefulness, this could be enriched with a contrasting explanation recommending some actions to the user for how to change the system’s behavior. In this case, one could also interpret a semifactual explanation as kind of an explanation of the contrasting explanation – i.e. why it must be done as suggested in the contrasting explanation and not somehow different.

From a computer science perspective, an interesting ques-

tion is how to exactly model the intuition of a semifactual explanation and how to compute these kinds of explanation efficiently. While semifactual explanations haven been studied for quite some time and are well known in philosophy [19]–[21] and psychology [22]–[24], there are – to the best of our knowledge – more or less unknown in the ML community. The only work, we are away of is [25] in which the authors study how to compute realistic semifactual explanation of deep neural networks applied to computer vision problems (in particular image classification).

III. SEMIFACTUAL EXPLANATIONS OF REJECT

In this Section we propose a model agnostic modeling of (diverse) semifactual explanations (see Section II-B) of reject options – i.e. our proposed modeling is not tailored towards a specific reject option or model.

We formalize a single semifactual explanation as an solution to an optimization problem:

$$\arg \min_{\vec{x}_{\text{sf}} \in \mathbb{R}^d} \ell(\vec{x}_{\text{sf}}) \quad (4)$$

where the loss function $\ell(\vec{x}_{\text{sf}})$ consists of different parts covering different aspects of semifactual explanations:

$$\ell(\vec{x}_{\text{sf}}) = \ell_{\text{feas-sf}}(\vec{x}_{\text{sf}}) + \ell_{\text{diverse}}(\vec{x}_{\text{sf}}) + \ell_{\text{similarity}}(\vec{x}_{\text{sf}}) + \ell_{\text{simple}}(\vec{x}_{\text{sf}}) \quad (5)$$

Since Eq. (5) might not be differentiable for arbitrary models, we might have to solve Eq. (4) by using a general black-box methods such as Downhill-Simplex method.

In the following, we introduce and motivate the different parts of the loss function Eq. (5) – note that all these sub-loss functions contain a regularization strength $C_\gamma > 0$ allowing the user to balance between the different objectives.

a) Feasibility & Semifactual property: As stated in Section II-B, there are two fundamental properties that make up a semifactual explanation: i) the semifactual \vec{x}_{sf} must be still rejected but ii) its certainty $r(\vec{x}_{\text{sf}})$ must not be worse, ideally it should be larger, than the certainty of the original sample \vec{x} . We model these two properties as two objectives that are merged together as a weighted sum:

$$\ell_{\text{feas-sf}}(\vec{x}_{\text{sf}}) = C_{\text{feasibility}} \cdot \max(r(\vec{x}_{\text{sf}}) - \theta, 0) + C_{\text{sf}} \cdot \max(r(\vec{x}) - r(\vec{x}_{\text{sf}}), 0) \quad (6)$$

b) Low complexity: Since semifactual explanations are feature based explanations, a low-complexity explanation (i.e. easy to understand explanation) should use only very few features. We model this by adding a penalty to the objective if more than $\mu \geq 1$ features are changed:

$$\ell_{\text{simple}}(\vec{x}_{\text{sf}}) = C_{\text{simple}} \cdot \max\left(\sum_{i=1}^d \mathbb{1}((\vec{x}_{\text{sf}} - \vec{x})_i \neq 0) - \mu, 0\right) \quad (7)$$

With the hyperparameter $\mu \geq 1$ enables the user to control the complexity of the generated explanations.

c) Similarity: While low-complexity is a common requirement of explanations, including semifactual explanations, another special property of semifactuals is that the stated

Algorithm 1 Computation of Diverse Semifactuals

Input: Original input \vec{x} , $k \geq 1$: number of diverse semifactuals, Reject option $r(\cdot)$

Output: Set of diverse semifactuals $\mathcal{R} = \{\vec{x}_{\text{sf}}^i\}$

```

1:  $\mathcal{F} = \{\}$  ▷ Initialize set of black-listed features
2:  $\mathcal{R} = \{\}$  ▷ Initialize set of diverse semifactuals
3: for  $i = 1, \dots, k$  do ▷ Compute  $k$  diverse semifactuals
4:    $\vec{x}_{\text{sf}}^i = \text{SF}_{r(\cdot)}(\vec{x}, \mathcal{F})$  ▷ Compute next semifactual
5:    $\mathcal{R} = \mathcal{R} \cup \{\vec{x}_{\text{sf}}^i\}$ 
6:    $\mathcal{F} = \mathcal{F} \cup \{j \mid (\vec{x}_{\text{sf}}^i - \vec{x})_j \neq 0\}$  ▷ Update set of black-listed features
7: end for
```

changes must be reasonable large to make the “even if ...” explanation useful – i.e. the semifactual \vec{x}_{sf} must be “sufficiently different” from the original sample \vec{x} , which we propose to model by the Euclidean distance:

$$\ell_{\text{similarity}}(\vec{x}_{\text{sf}}) = -C_{\text{similarity}} \cdot \|\vec{x}_{\text{sf}} - \vec{x}\|_2 \quad (8)$$

Note the negative sign in front of Eq. (8) which is due to the fact that we minimize the overall loss function Eq. (5). Furthermore, note that Eq. (8) and Eq. (7) are kind of contradictory and for particular scenarios an appropriate balance must be found.

d) Diversity: Since the precise meaning of “diverse semifactuals” might be different for different use-cases, we propose a very general definition of diversity, namely that the number of simultaneously changed features should be small:

$$\psi(\vec{x}_{\text{sf}}^j, \vec{x}_{\text{sf}}^k) = \sum_{i=1}^d \mathbb{1}\left((\vec{\delta}^j)_i \neq 0 \wedge (\vec{\delta}^k)_i \neq 0\right) \quad (9)$$

where $\vec{\delta}^j = \vec{x}_{\text{sf}}^j - \vec{x}$ and $\mathbb{1}(\cdot)$ denotes the indicator function that returns 1 if the boolean expression is true and 0 otherwise. Note that lower values of Eq. (9) correspond to more diverse semifactuals.

Here we propose a sequential approach for computing a set of diverse semifactual explanations: For a set of already computed semifactual explanations $\{\vec{x}_{\text{sf}}^i\}$ (this set is empty in the beginning), we deduce a set of features \mathcal{F} that are already used by the current semifactual explanations and therefore should not be used again:

$$\mathcal{F} = \{j \mid \exists i : (\vec{x}_{\text{sf}}^i - \vec{x})_j \neq 0\} \quad (10)$$

Since diverse explanations should not use the same features, we implement a penalty for using already used features:

$$\ell_{\text{diverse}}(\vec{x}_{\text{sf}}) = C_{\text{diverse}} \cdot \sum_{j \in \mathcal{F}} \mathbb{1}\left((\vec{x}_{\text{sf}} - \vec{x})_j \neq 0\right) \quad (11)$$

After computing a diverse semifactual explanation, we update Eq. (10) and compute the next one by solving Eq. (4) – for convenience, we use $\text{SF}_{r(\cdot)}(\vec{x}, \mathcal{F})$ to denote the computation of a semifactual \vec{x}_{sf} of a reject option $r(\cdot)$ at a given sample \vec{x} subject to a set \mathcal{F} of black-listed features.

The complete pseudo-code of computing a set of diverse semifactuals is given in Algorithm 1.

TABLE I: Quantitative evaluation of our proposed semifactual explanations – all numbers are rounded to two decimal places.

	<i>DataSet</i>	<i>Feas.</i>	<i>Spars.</i>	<i>Div.</i>	<i>Recall</i>
k-NN	Wine	1.0 ± 0.0	0.08 ± 0.0	0.0 ± 0.0	0.14 ± 0.03
	Breast cancer	1.0 ± 0.0	0.04 ± 0.0	0.0 ± 0.0	0.09 ± 0.0
	Flip	0.8 ± 0.16	0.09 ± 0.0	0.0 ± 0.0	0.21 ± 0.03
	t21	1.0 ± 0.0	0.07 ± 0.0	0.0 ± 0.0	0.11 ± 0.01
GNB	Wine	0.99 ± 0.0	0.08 ± 0.0	0.0 ± 0.0	0.13 ± 0.04
	Breast cancer	0.89 ± 0.0	0.04 ± 0.0	0.0 ± 0.0	0.09 ± 0.0
	Flip	0.4 ± 0.24	0.1 ± 0.0	0.0 ± 0.0	0.11 ± 0.02
	t21	0.49 ± 0.17	0.06 ± 0.0	0.0 ± 0.0	0.12 ± 0.01
RandForest	Wine	1.0 ± 0.0	0.08 ± 0.0	0.0 ± 0.0	0.11 ± 0.03
	Breast cancer	1.0 ± 0.0	0.04 ± 0.0	0.0 ± 0.0	0.08 ± 0.0
	Flip	0.6 ± 0.24	0.09 ± 0.0	0.0 ± 0.0	0.21 ± 0.04
	t21	0.98 ± 0.0	0.06 ± 0.0	0.02 ± 0.07	0.11 ± 0.01

IV. EXPERIMENTS

We empirically evaluate our proposed mode agnostic semifactual explanations of reject on several ML models and data sets. All experiments are implemented in Python and are publicly available on GitHub¹.

A. Data Sets

We use the following diverse data sets in our empirical evaluations – all data sets are standardized.

1) *Wine*: The “Wine data set” [26] is used for predicting the cultivator of given wine samples based on their chemical properties. The data set contains 178 samples and 13 numerical features such as alcohol, hue and color intensity.

2) *Breast cancer*: The “Breast Cancer Wisconsin (Diagnostic) Data Set” [27] is used for classifying breast cancer samples into benign and malignant. The data set contains 569 samples and 30 numerical features such as area, smoothness and compactness.

3) *Flip*: This data set is used for the prediction of fibrosis. The set consists of samples of 118 patients and 12 numerical features such as blood glucose, BMI and total cholesterol, and was provided by the Department of Gastroenterology, Hepatology and Infectiology of the University Magdeburg [28]. As the data set contains some rows with missing values, we chose to replace these missing values with the corresponding feature mean.

4) *T21*: This data set is used for early diagnosis of chromosomal abnormalities, such as trisomy 21, in pregnant women. The data set consists of 18 numerical features such as heart rate and weight, and contains over 50000 samples but only 0.8 percent abnormal samples (e.g. cases of trisomy 21) – i.e. it is highly imbalanced. It was collected by the Fetal Medicine Centre at King’s College Hospital and University College London Hospital in London [29].

B. Setup

Since our method Algorithm 1 (see Section III) is completely model agnostic, we evaluate it on a set of diverse classifiers: k-nearest neighbors classifier (kNN), Gaussian naive Bayes classifier (GNB), random forest classifier (RandForest). We

always use a conformal prediction based implementation of a reject option [15] (see Section II-A).

For each data sets and each classifier: First, we fit the model and reject option to the data training data. Next, we select a random subset of 30 percent of the features that are going to be perturbed. We then perturb these features by adding Gaussian noise to the original values. We compute 3 diverse semifactual explanations (see Section III) of those samples that are rejected due to the perturbation – i.e. samples that were accepted before applying the perturbation, but rejected afterwards.

We also perform hyperparameter tuning in order to find the best performing model parameters (of the reject option) – including the hyperparameters of the respective classifiers, which are obtained by a grid search. In addition, we try to find an appropriate rejection threshold by using the Knee/Elbow method [30]. In a real world scenario the threshold might be tuned to allow for a more relaxed or strict rejection scenario, however for the purpose of our research finding the knee point gives us a fairly good approximation of what would usually be considered an appropriate rejection threshold.

Note that we run all experiments in a 5-fold cross validation.

C. Evaluation

We evaluate several quantitative aspects of the obtained explanations:

- *Feas.*: Feasibility of the explanations – i.e. whether there are valid semifactual explanations according to Section III – i.e. is the semifactual explanation \vec{x}_{cf} still rejected but the assigned certainty $r(\vec{x}_{cf})$ is at least as good as the one of the original sample \vec{x} . Larger numbers are better.
- *Spars.*: Sparsity of the explanations – i.e. how many (percentage) of the available features where changed, smaller numbers are better. Note that sparsity acts as a proxy of “low-complexity”.
- *Div.*: Diversity of the explanations – i.e. counting the number of simultaneously changed features, smaller numbers are better.
- *Recall*: Recall of perturbed features in the explanations – smaller numbers are “kind of better” because semifactual explanations are not supposed to highlight the “relevant features”.

We always report mean and variance of these scores.

¹<https://github.com/andreArtelt/DiverseSemifactualsReject>

D. Results

The results are shown in Table I – all numbers are rounded to two decimal points.

We observe that our proposed method Algorithm 1, across different models and data sets, consistently yields sparse and (very) diverse semifactual explanations. Only when applying Gaussian naive Bayes to the flip and t21 data set, our method struggles to compute feasible (i.e. valid) semifactual explanations. Also note that the performance on the flip data set drops for all other methods as well – we think that the chosen hyperparameters might not be optimal for this particular data set. Furthermore, we observe that the generated semifactual explanations often do not use the perturbed features – i.e. these features are the cause/reason of the reject – this highlights the usefulness of semifactual explanations (i.e. identifying non-relevant features for the “even if ...” explanations).

V. CONCLUSION

In this work we introduced semifactual explanations for explaining reject options. We proposed the conceptual idea of using diverse semifactual explanations of reject, we proposed a formalization of this concept as well as an implementation of this concept. We empirically evaluated quantitative aspects of our proposed methods on several data sets and observed a good performance across many different scenarios.

Based on this work, there exist several possible directions for future research:

Our proposed modeling of semifactual explanations of reject is model agnostic – i.e. it can be applied to any model and reject option. However, exploiting model internals might lead to better performance – e.g. runtime improvements, formal guarantees, etc. – and therefore it might be of interest to develop model specific methods for computing semifactual explanations.

The final semifactual explanations as computed by our proposed Algorithm 1 solely depend on the hyperparameters for balancing between the different objectives as discussed in Section III. While it is nice to give the user the ability to “fine tune” the explanations, it might be of interest to formalize the semifactuals as a multi-criteria optimization problem and develop a method for automatically exploring (i.e. computing) the pareto front of solutions.

In this work, we ignored the aspect of plausibility – i.e. we can not guarantee that the semifactual explanations are realistic and plausible in the data domain. However, in particular for example based explanations, missing plausibility might drastically reduce the benefits of the explanation [31]. Therefore, future work could address the issue by adding some kind of plausibility constraints to our proposed methods.

Although semifactual explanations are well known in psychology, it is somewhat “unclear” how to formalize this concept and consequently compute such explanations. We proposed a reasonable approach that showed good quantitative results. Still it is not clear if our proposed formalization always yields explanations that are useful to humans. Furthermore, it is also unclear if and how these types of explanations are useful for understanding machine learning based decision making

systems in general. However, in order to study the usefulness of such explanations, we must first be able to compute such explanations – which is now possible because of this work. Consequently, we suggest to evaluate and study the usefulness of our computed semifactuals in a user-study.

REFERENCES

- [1] A. E. Sallab, M. Abdou, E. Perot, and S. Yogamani, “Deep reinforcement learning framework for autonomous driving,” *Electronic Imaging*, vol. 2017, no. 19, pp. 70–76, 2017.
- [2] K. Hendrickx, L. Perini, D. Van der Plas, W. Meert, and J. Davis, “Machine learning with a reject option: A survey,” *arXiv preprint arXiv:2107.11277*, 2021.
- [3] E. parliament and council, “General data protection regulation: Regulation (eu) 2016/679 of the european parliament,” 2016.
- [4] C. Molnar, *Interpretable Machine Learning*, 2019, <https://christophm.github.io/interpretable-ml-book/>.
- [5] F. Doshi-Velez and B. Kim, “Towards a rigorous science of interpretable machine learning,” 2017.
- [6] F. Offert, ““i know it when i see it”: visualization and intuitive interpretability,” 2017.
- [7] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi, “A survey of methods for explaining black box models,” *ACM Comput. Surv.*, vol. 51, no. 5, Aug. 2018.
- [8] E. Tjoa and C. Guan, “A survey on explainable artificial intelligence (XAI): towards medical XAI,” *CoRR*, vol. abs/1907.07374, 2019.
- [9] A. Fisher, C. Rudin, and F. Dominici, “All Models are Wrong but many are Useful: Variable Importance for Black-Box, Proprietary, or Misspecified Prediction Models, using Model Class Reliance,” *arXiv e-prints*, p. arXiv:1801.01489, Jan 2018.
- [10] A. Aamodt and E. Plaza, “Case-based reasoning: Foundational issues, methodological variations, and system approaches,” *AI communications*, 1994.
- [11] S. Wachter, B. Mittelstadt, and C. Russell, “Counterfactual explanations without opening the black box: Automated decisions and the gdpr,” *Harv. JL & Tech.*, vol. 31, p. 841, 2017.
- [12] S. Verma, J. Dickerson, and K. Hines, “Counterfactual explanations for machine learning: A review,” 2020.
- [13] B. Kim, O. Koyejo, and R. Khanna, “Examples are not enough, learn to criticize! criticism for interpretability,” in *Advances in Neural Information Processing Systems 29*, 2016.
- [14] A. Artelt, J. Brinkrolf, R. Visser, and B. Hammer, “Explaining reject options of learning vector quantization classifiers,” *arXiv preprint arXiv:2202.07244*, 2022.
- [15] A. Artelt, R. Visser, and B. Hammer, “Model agnostic local explanations of reject,” *arXiv preprint arXiv:2205.07623*, 2022.
- [16] H. Linusson, U. Johansson, H. Boström, and T. Löfström, “Classification with reject option using conformal prediction,” in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2018, pp. 94–105.
- [17] G. Shafer and V. Vovk, “A tutorial on conformal prediction,” *J. Mach. Learn. Res.*, vol. 9, pp. 371–421, 2008. [Online]. Available: <https://dl.acm.org/citation.cfm?id=1390693>
- [18] A. E. Khandani, A. J. Kim, and A. Lo, “Consumer credit-risk models via machine-learning algorithms,” *Journal of Banking & Finance*, vol. 34, no. 11, 2010.
- [19] N. Goodman, *Fact, Fiction, and Forecast*. Harvard University Press, 1983.
- [20] C. B. Cross, “Jonathan bennett on ‘even if’,” *Linguistics and Philosophy*, vol. 8, no. 3, pp. 353–357, 1985.
- [21] S. Barker, “Even, still and counterfactuals,” *Linguistics and Philosophy*, vol. 14, no. 1, pp. 1–38, 1991.
- [22] D. S. Boninger, F. Gleicher, and A. Strathman, “Counterfactual thinking: From what might have been to what may be,” *Journal of personality and social psychology*, vol. 67, no. 2, p. 297, 1994.
- [23] C. Santamaría, O. Espino, and R. M. Byrne, “Counterfactual and semifactual conditionals prime alternative possibilities,” *Journal of Experimental Psychology: Learning, Memory, and Cognition*, vol. 31, no. 5, p. 1149, 2005.
- [24] G. Macbeth and E. Razumiejczyk, “Implicit facilitation effect on counterfactual and semifactual thinking,” *Education Sciences & Psychology*, vol. 54, no. 4, 2019.

- [25] E. M. Kenny and M. T. Keane, "On generating plausible counterfactual and semi-factual explanations for deep learning," in Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021. AAAI Press, 2021, pp. 11 575–11 585. [Online]. Available: <https://ojs.aaai.org/index.php/AAAI/article/view/17377>
- [26] D. C. S. Aeberhard and O. de Vel, "Comparison of classifiers in high dimensional settings," Tech. Rep. no. 92-02, 1992.
- [27] O. L. M. William H. Wolberg, W. Nick Street, "Breast cancer wisconsin (diagnostic) data set," [https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+\(Diagnostic\)](https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)), 1995.
- [28] J.-P. Sowa, D. Heider, L. P. Bechmann, G. Gerken, D. Hoffmann, and A. Canbay, "Novel algorithm for non-invasive assessment of fibrosis in nafld," PLOS ONE, vol. 8, no. 4, pp. 1–6, 04 2013. [Online]. Available: <https://doi.org/10.1371/journal.pone.0062439>
- [29] K. H. Nicolaides, K. Spencer, K. Avgidou, S. Faiola, and O. Falcon, "Multicenter study of first-trimester screening for trisomy 21 in 75 821 pregnancies: results and estimation of the potential impact of individual risk-orientated two-stage first-trimester screening," Ultrasound in Obstetrics & Gynecology, vol. 25, no. 3, pp. 221–226, 2005. [Online]. Available: <https://obgyn.onlinelibrary.wiley.com/doi/abs/10.1002/uog.1860>
- [30] V. Satopaa, J. Albrecht, D. Irwin, and B. Raghavan, "Finding a "kneedle" in a haystack: Detecting knee points in system behavior," in 2011 31st International Conference on Distributed Computing Systems Workshops, 2011, pp. 166–171.
- [31] U. Kuhl, A. Artelt, and B. Hammer, "Keep your friends close and your counterfactuals closer: Improved learning from closest rather than plausible counterfactual explanations in an abstract setting," in 2022 ACM Conference on Fairness, Accountability, and Transparency, ser. FAccT '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 2125–2137. [Online]. Available: <https://doi.org/10.1145/3531146.3534630>