Counterfactual Explanations for Incorrect Predictions Made by AI Models

Amir Asrzad
University of Massachusetts Lowell, amir_asrzad@student.uml.edu

Xiaobai Li
University of Massachusetts Lowell, xiaobai_li@uml.edu

Counterfactual Explanations for Incorrect Predictions Made by AI Models

Short Paper

Amir Asrzad
University of Massachusetts Lowell
Lowell, MA 01854, U.S.A.
amir_asrzad@student.uml.edu

Xiao-Bai Li
University of Massachusetts Lowell
Lowell, MA 01854, U.S.A.
xiaobai_li@uml.edu

Abstract

Advanced AI models are powerful in making accurate predictions for complex problems. However, these models often operate as black boxes. This lack of interpretability poses significant challenges, especially in high-stakes applications such as finance, healthcare, and criminal justice. Explainable AI seeks to address the challenges by developing methods that can provide meaningful explanations for humans to understand. When black box models are used for prediction, they inevitably produce errors. It is important to appropriately explain incorrect predictions. This problem, however, has not been addressed in the literature. In this study, we propose a novel method to provide explanations for misclassified cases made by black box models. The proposed method takes a counterfactual explanation approach. It builds a decision tree to find the best counterfactual examples for explanations. Incorrect predictions are rectified using a trust score measure. We validate the proposed method in an evaluation study using real-world data.

Keywords: Interpretable machine learning, counterfactual explanation, decision trees, misclassification

Introduction

Artificial intelligence (AI) and machine learning (ML) have become ubiquitous technologies in the modern world. They offer enormous potential to revolutionize various industries and businesses. While these technologies have become increasingly powerful in making accurate predictions for complex problems, their models often operate as "black boxes," meaning that the inner workings of the models and the logic behind the models' output are opaque and difficult to explain. This lack of interpretability poses significant challenges, especially in high-stakes applications, such as finance, healthcare, and criminal justice, where decisions based on these models can have significant consequences. Interpretable machine learning (IML) or explainable AI (XAI) seeks to address the challenges by developing methods and techniques that can provide meaningful explanations for humans to understand how the black box models make predictions (Molnar 2023). By increasing the interpretability of the models, we can build trust, transparency, and accountability in AI systems and make sure that their outputs are fair, ethical, and reliable.

An increasingly popular XAI/IML method is counterfactual explanations (Verma et al. 2022; Wachter et al. 2018). Unlike many traditional XAI and IML methods, counterfactual explanations do not directly answer the "why" part of a decision; instead, they provide alternative scenarios, or *counterfactuals*, that explore what would have happened if certain inputs or features had been different. In this way, they explain to the end-users receiving an unfavorable decision what improvements are needed in order to achieve the desired outcome. For example, if an ML model predicts that a loan applicant would default on the loan, a counterfactual explanation could advise the applicant what factors would need to change in order to have the loan approved. Counterfactual explanations are easy to understand, highly persuasive, and capable of generating actionable insights (Fernández-Loría et al. 2022). As such, these methods have gained increasing attention in XAI/IML research in recent years (Guidotti 2022; Stepin et al. 2021).

One of the important criteria for evaluating XAI/IML methods is the *fidelity*, which measures the extent to which an interpretable method can accurately approximate the predictions of the black box model. It is desirable for the interpretable method to have high fidelity, making predictions consistent with those of the black box model. However, when black box models are used for prediction or classification, they inevitably produce errors or misclassifications. Nevertheless, in existing XAI/IML approaches, fidelity is pursued entirely without considering the errors. That is, when the black box model misclassifies an instance, the interpretable method would, based on the fidelity criterion, treat the misclassified result as correctly classified and then attempt to explain the actually incorrect classification outcome as if it is correct. This misinterpretation clearly has a significant impact on subsequent actions. In our literature search, however, we have not found any study that addresses this problem.

In this study, we explore the problem of how to rectify or reduce incorrect predictions made by AI and ML models and offer more suitable explanations accordingly. We focus on classification problems with two categorical outcomes: *beneficial* (or desired) and *adverse* (or undesired). There are two types of errors in this problem. The first is to misclassify a beneficial class to adverse (b2a) and the second is to misclassify an adverse class to beneficial (a2b). Suppose the classification model is used by an organization to support decision making on individual customers (e.g., loan application). It is easy to see that the first type of error (b2a) causes harm to the customers while the second type of error (a2b) causes harm to the organization. Thus, the audience of the explanations for the two types are different. And the purposes and methods of explanations would also be different. As a result, our research questions are two folds: (1) How can we rectify and explain the misclassification outcome to individuals when the black box model incorrectly categorizes a beneficial class to adverse? and (2) How can we rectify and explain the misclassification outcome to organizations when the model incorrectly categorizes an adverse class to beneficial.

In this paper, we propose a novel and practical method to provide explanations for misclassified cases made by AI and ML models. The proposed method takes a counterfactual explanation approach that can be used no matter what model is employed for classification. In our method, we first apply a black box model to classify given instances. We then fit a decision tree with the classification results made by the black box model. This decision tree, called explanation tree, is then used to find the best counterfactual examples for explanations. For the two types of classification errors, we offer two different types of explanation, one to the individual customers and the other to the organization's decision makers and data analysts.

This work makes important contributions to machine learning, AI, and data science research in several aspects: (1) We investigate the problem of how to explain misclassified outcomes made by AI and ML models. This is an important and interesting research problem unexplored in literature. (2) We propose a novel and practical method to provide counterfactual explanations for correctly and incorrectly classified cases. (3) We validate the proposed method in an empirical evaluation study using real-world data.

Related Work

Interpretable machine learning or explainable AI methods can be classified into several categories based on the scope and nature of the models (Adadi and Berrada 2018; Du et al. 2020; Guidotti et al. 2018). In terms of the scope of interpretability, *globally interpretable* methods provide understanding of the entire model behavior and the whole logic behind the model's predictions. Examples of these methods include decision trees, rule-based models, linear models, and additive models (Wang et al. 2022). *Locally interpretable* methods explain how a particular prediction is made by the model for a specific input (Kim et al. 2023). Explanation methods can also be classified based on whether the method is dependent on the ML model. *Model-specific* methods are designed for a specific ML model. They rely on the inner workings of the model to generate explanations. On the other hand, *model-agnostic* methods can be applied to any ML model, regardless of its architecture or training process. These methods rely on analyzing the inputs and outputs of the model to generate explanations. Model-agnostic methods are usually *post hoc*, meaning that explanations are generated after the black box model has been trained to make predictions. Counterfactual explanation methods are usually local, model-agnostic, and post-hoc method.

Recent trends in the XAI/IML research tend to focus on post-hoc model-agnostic approach. A popular stream of this approach is the feature importance method, which explains the output of a model by providing the importance of individual features for the model's prediction. A well-known example of these methods is LIME (Local Interpretable Model-agnostic Explanations) (Ribeiro et al. 2016). LIME works by generating a simple interpretable model that approximates the behavior of the black box model locally around the instance of interest. Given the instance, LIME first generates a set of perturbed instances and gets their black box model predictions. These instances are next weighted according to their proximity to the original instance. A simple interpretable model (e.g., a linear model) is then trained on the dataset including the weighted perturbed instances and their predicted labels. This model can then be used to explain the black box predictions based on the importance of the features in the model.

Another well-known model-agnostic feature-importance method is SHAP (SHapley Additive exPlanations) (Lundberg and Lee 2017). SHAP is based on the concept of Shapley values from cooperative game theory. The basic idea behind SHAP is to assign an importance value, based on the Shapley value, to each feature in the input data that contributes to the final prediction. This is attained by evaluating the model output for every possible ordered subset of features to get the marginal contribution of each feature to the output. The SHAP values are then calculated as a weighted average of the marginal contributions over all possible subsets of features. The SHAP importance values can then be used to understand the importance of each feature and to explain the result of the model. SHAP provides a unified framework for interpreting the predictions of complex machine learning models. Both LIME and SHAP have been widely used in various applications involving structured data and unstructured text and image data.

Counterfactual explanations represent another common strategy for post-hoc model-agnostic approach. Counterfactual explanations describe how a model's output would change if some inputs were different (Martens and Provost 2014). Most approaches for generating counterfactual explanations are based on optimization techniques (Molnar 2023; Wachter et al. 2018). The basic idea is to find an instance or a small set of instances (counterfactuals) that are as similar to the instance of interest as possible based on feature values but have different (desired) prediction outcome. The objective of the optimization problem includes a loss function that measures the distance between the counterfactuals and the instance of interest. Desired prediction outcome can be specified either as a constraint or as another component of the objective function. An optimization method, such as a gradient-based method or a metaheuristic method, is then used to find the best counterfactuals. Along the line of optimization approach, Dhurandhar et al. (2018) introduce the contrastive explanation method (CEM). The basic idea of CEM is to identify the smallest set of changes to the features required to produce a different prediction, making the explanations easy to understand. Van Looveren and Klaise (2021) use prototypes, which are representative instances of each class, to guide the generation of counterfactuals. Their method, called Counterfactual Explanations Guided by Prototypes (CEGP), finds the smallest feature changes required to transform the input into a prototype of a different class to generate a counterfactual explanation. Sokol et al. (2022) implement a brute force approach to generate counterfactuals in their library for Fairness, Accountability and Transparency (FAT) algorithms.

Counterfactual explanations can also be generated using decision trees (Fernández et al. 2019; Guidotti et al. 2020; van der Waa et al. 2018). Similar to LIME, decision tree based methods first generates a set of synthetic instances locally around the instance of interest and obtain their predictions using the black box model. Next, this set of instances and their predictions are used to train a decision tree, which becomes a local interpretable model for the black box model. This decision tree is then used to locate the instance of interest and its counterfactuals. The best counterfactuals are selected based on some criteria that can be easily represented with the decision tree structure, such as the shortest path between the instance of interest and the counterfactuals.

In order to faithfully represent the behavior of a black box model, post-hoc model-agnostic approaches often use the predictions of the black box model to train the interpretable model, as described above. When the black box model makes an incorrect prediction, the interpretable model would still attempt to explain the incorrect prediction as if it is correct. This misinterpretation problem clearly has a significant practical implications. Rudin (2019) observes that recent work tends to explain only the correct predictions. In our literature search, we have not found any study that addresses this misinterpretation problem. There are some studies in XAI/IML literature addressing misclassification issue (Abid et al. 2022; Vermeire et al. 2022). However, these studies consider identifying the reasons for the mistakes, rather than the misinterpretation issue, and their works tend to focus on image recognition applications. In this paper, we address the problem of how to correctly explain misclassified cases in business applications.

The Proposed Method

This study applies a computational design science methodology to develop a new and innovative artifact, called explanation tree, for explaining incorrect predictions made by AI and ML models. We first use a small example to illustrate the misclassification problem and explain the basic idea of the explanation tree. We then provide a detailed description of the proposed method.

Explanation Tree and Trust Score

Consider an illustrative example of a dataset in Table 1. The dataset is used to train machine learning models to support decision making for small business loan applications. It contains information on 10 customers with three features: Income, requested Loan Amount, and whether the customer owns a house (House Owner). Suppose a black box model is built and its prediction is shown in the Predicted Outcome column (where 'adverse' or 'beneficial' indicates the customer incurs a loss or gain to the lender, respectively). Since it is training data, the actual outcome is known, as shown in the last column. It can be seen that there are two misclassified cases. Customer #9, who is actually beneficial, is misclassified as adverse (b2a), while customer #10, an adverse case, is misclassified as beneficial (a2b).

When the model is used to classify new applications, misclassification can also occur. As a result, a beneficial applicant would be denied and provided with a wrong explanation, and an adverse applicant would be accepted. These scenarios clearly show that it is important to consider misclassification problem when explaining machine learning predictions. We develop a novel approach to address this problem.

Our proposed method uses a decision tree, which we call the **explanation tree**, to find counterfactual examples for the individuals with undesired classification outcome and to explain what changes should be made in order to have the desired outcome. To build the explanation tree, we first remove the misclassified instances (i.e., #9 and #10 in the dataset in Table 1). The black-box predicted outcomes (which are also the actual outcomes) are used as the class label to grow the tree as large as possible so that it fits the predicted outcome perfectly. As such, the explanation tree has 100% fidelity to the black box model in terms of correctly classified cases. We should point out that the explanation tree is not intended for making predictions. Instead, it serves as an explanation model to explain the predictions made by the black box model and to find the counterfactuals that would improve the prediction outcome. Therefore, it is desirable that the explanation tree produces predictions as consistent with those of the black box model as possible. Because the explanation tree will use the training instances as the counterfactuals, it should include only the correctly classified instances. Otherwise, if a misclassified instance is included, the instance may be misused as a counterfactual for post-hoc explanation, making wrong recommendation to the audience.

Customer ID	Income ($000)	Loan Amount ($000)	House Owner	Predicted Outcome	Actual Outcome
#1	94	165	yes	beneficial	beneficial
#2	86	225	yes	adverse	adverse
#3	112	190	yes	beneficial	beneficial
#4	67	95	no	beneficial	beneficial
#5	78	105	no	beneficial	beneficial
#6	69	160	no	adverse	adverse
#7	87	173	no	adverse	adverse
#8	98	105	no	beneficial	beneficial
#9	74	125	no	adverse	beneficial
#10	94	149	no	beneficial	adverse

Table 1. An Illustrative Example of Loan Application

Figure 1 shows the explanation tree built on the data in Table 1 (after removing #9 and #10). Each internal node is represented by a rectangle box, inside which the splitting criterion is specified. Each leaf is represented by a rectangle box with rounded corners. The first row inside the leaf box shows the leaf ID, followed by the IDs of the instances in the leaf. For example, leaf L1 contains customer #1, while leaf L4 contains customers #4 and #5. Since the tree has 100% fidelity, the instances in each leaf have the same class. This is indicated on the second row in the leaf box.

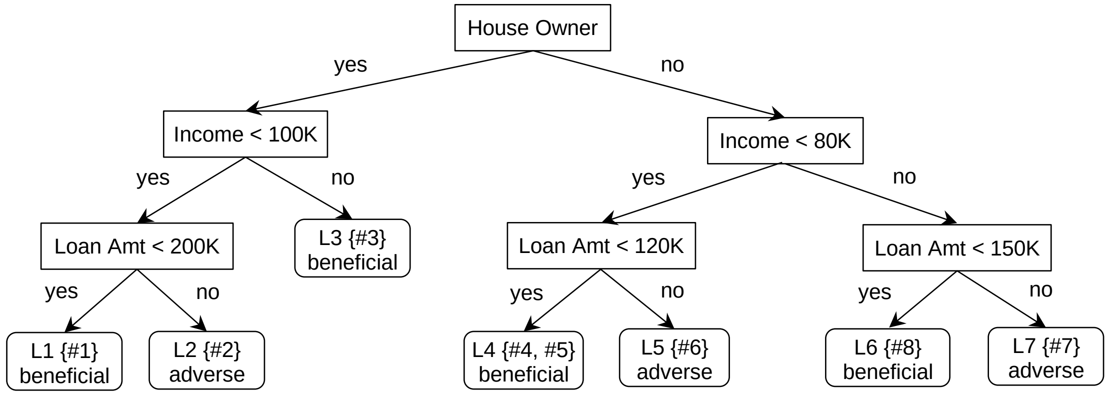

Figure 1. Explanation Tree for the Dataset in Table 1

When a new instance is classified by a machine learning model, it does not have the actual class label to tell whether it is misclassified or not. Therefore, it is necessary to have a metric to decide if the model correctly classifies a new instance. A well-known measure, called *trust score* (Jiang et al. 2018), can be used for this purpose. Given a new instance **x** and its predicted class $\hat{y}$, the trust score (TS) of $(\mathbf{x}, \hat{y})$ is the ratio between the distance from the new instance to the nearest class different from the predicted class ($\tilde{d}$) and the distance from the new instance to the predicted class (d), i.e., $TS(\mathbf{x}, \hat{y}) = \tilde{d}/d$. The larger the trust score, the closer the new instance to the predicted class, and the more trustworthy the predicted outcome. Trust score is computed independent of the classification model. Jiang et al. (2018) have shown that it can effectively measure if a classification result is correct or not. Various empirical studies have demonstrated that the trust score outperforms other popular model confidence and discriminant scores (de Bie et al. 2021; Delaney et al. 2021; Jiang et al. 2018). Therefore, trust score is used in our method to evaluate the trustworthiness of a prediction.

The Proposed Counterfactual Explanation Method

Let $D_N = \{[x_{i1}, \ldots, x_{im}], y_i\}_{i=1}^{N} = \{\mathbf{x}_i, y_i\}_{i=1}^{N}$ be the set of training and validation data for building the black box model. Without loss of generality, assume that $y_i = \{0,1\}, \forall i$. Let $\hat{y}_i$ be the predicted value of y_i by the model. To build the explanation tree, we remove misclassified instances from D_N. For the remaining n correctly classified instances, $\hat{y}_i = y_i, \forall i$. In this case, we can use y_i to represent both the predicted and actual class. So, let $D_n = \{\mathbf{x}_i, y_i\}_{i=1}^{n}$ be the dataset after removing misclassified instances from D_N. Given a new instance $\mathbf{x}'$, let $\hat{y}'$ be the predicted class from the black box model. Let y' be the true but unknown class. Then, if the instance is correctly classified, $y' = \hat{y}'$; otherwise, $y' = 1 - \hat{y}'$.

Our proposed method first builds an explanation tree that fits the output of the black box model perfectly. Given a new instance and its black box prediction, we consider two scenarios, i.e., when the instance is correctly classified and misclassified. We compute the trust score for each scenario and find the counterfactual instances under each scenario. Then, we compare the two trust scores to decide if the new instances is misclassified or not. The counterfactual instance corresponding to the decision is provided for explaining the final outcome. The proposed method includes the following computational steps.

1. Given a new instance $\mathbf{x}'$, obtain the black box prediction $\hat{y}'$. Build an explanation tree T using the instances in D_n, plus $(\mathbf{x}', \hat{y}')$.
2. Find the leaf in T where the new instance $\mathbf{x}'$ is located and denote the leaf by L'.

3. Consider the scenario when $\mathbf{x}'$ is correctly classified. Set $y' = \hat{y}'$ and compute the trust score $TS(\mathbf{x}', y' = \hat{y}')$. Then, find the counterfactual instance for $\mathbf{x}'$ as follows:
    a. Find the leaf in T that has the shortest path to L' among all leaves with the opposite class to y'. Denote this leaf by L^c and its class by $y^c = 1 - y'$. Without loss of clarity, we also use L^c to denote the set of instances in L^c. These instances are the candidates of the counterfactual instances for $\mathbf{x}'$.

    b. Select the best counterfactual instance as the one having the maximum trust score among all candidates in L': $\mathbf{x}^{c+} = \max_{\mathbf{x} \in L^c} TS(\mathbf{x}, y^c)$, where $c+$ indicates that the counterfactual is for the correct classification scenario.
4. Consider the scenario when $\mathbf{x}'$ is misclassified. Set $y' = 1 - \hat{y}'$ and compute the trust score $TS(\mathbf{x}', y' = 1 - \hat{y}')$. Then, find the counterfactual instance for $\mathbf{x}'$ by following the same procedure as specified in step 3. Steps 4a and 4b are the same as steps 3a and 3b, respectively, except that in step 4b, we use $\mathbf{x}^{c-}$ to denote the counterfactual instance for the misclassification scenario.
5. If $TS(\mathbf{x}', y' = \hat{y}') \geq TS(\mathbf{x}', y' = 1 - \hat{y}')$, use $\mathbf{x}^{c+}$ obtained in step 3b as the counterfactual for $\mathbf{x}'$; otherwise, use $\mathbf{x}^{c-}$ obtained in step 4b as the counterfactual.

We use the illustrative example in Figure 1 to explain how the proposed method works. Consider a new instance of loan application, $\mathbf{x}'$. Following steps 1 and 2, suppose $\mathbf{x}'$ is classified as 'adverse' and located in $L' = L5$. In step 3, we first set $y' = adverse$ and compute the trust score $TS(\mathbf{x}', y' = adverse)$. Step 3a then finds $L^c = L4$, because L4 has the shortest path to L5 among all leaves with the opposite class $y^c = beneficial$. Suppose customer #5 in L4 is the best counterfactual found in step 3b, i.e., $\mathbf{x}^{c+} = \mathbf{x}_5$. In step 4, we first set $y' = beneficial$ and compute the trust score $TS(\mathbf{x}', y' = beneficial)$. Step 4a then finds $L^c = L5$, because L5 has the shortest path to itself with the opposite class $y^c = adverse$. Now, customer #6 in L5 is the only counterfactual found in step 4b, i.e., $\mathbf{x}^{c-} = \mathbf{x}_6$.

In step 5, if $TS(\mathbf{x}', y' = \hat{y}') \geq TS(\mathbf{x}', y' = 1 - \hat{y}')$, then we determine that $\mathbf{x}'$ is classified correctly as 'adverse' and use $\mathbf{x}_5$ as the counterfactual. The explanation to the applicant $\mathbf{x}'$ is: "your application is declined, but it will be approved if you can reduce the requested loan amount to below $120K" ("Loan Amount < 120K" is the splitting criterion between L4 and L5 in Figure 1). On the other hand, if $TS(\mathbf{x}', y' = \hat{y}') < TS(\mathbf{x}', y' = 1 - \hat{y}')$, we determine that $\mathbf{x}'$ is misclassified by the model and should be reclassified as 'beneficial'. In this case, the counterfactual $\mathbf{x}_6$ will be provided to the data analyst (who built the black box model) to explain what causes the model to misclassify $\mathbf{x}'$, providing insights for further improvement of the model.

Suppose we have another new application that is classified as 'beneficial'. We follow the same steps above to determine if it is correctly classified and find corresponding counterfactuals. If it is correctly classified, then no explanation is needed, because the model works well and the applicant is happy about getting the loan approved. On the other hand, if it is misclassified, then the application is reclassified to 'adverse'. The counterfactual will be provided to the applicant to explain what changes should be made in order to have the loan approved. It will also be provided to the data analyst to explain what causes the model to misclassify the application.

Experimental Evaluation

We conducted a preliminary experiment on real-world datasets to evaluate the effectiveness of the proposed method against some state-of-the-art-baseline methods.

Data and Experimental Setup

Both datasets are publicly available and commonly used for experimental evaluation in XAI/IML literature. The first dataset, Diabetes, was originally collected by the National Institute of Diabetes and Digestive and Kidney Diseases (https://www.kaggle.com/datasets/akshaydattatraykhare/diabetes-dataset). It contains diabetes information on Pima Indian females of 21 years or older, including 768 patients, with eight numeric features and two classes (present or absent). The second dataset was provided by FICO (https://community.fico.com/s/explainable-machine-learning-challenge). It contains financial and credit data on 10,459 consumers with 23 numeric features. The class attribute has binary values, indicating if the individual is of high or low risk.

Two black box models were used for predictions: neural networks and random forest. We used the implementation from the *scikit-learn* package (Pedregosa et al. 2011), with default parameters. Each dataset was divided into two sections, approximately 80% for training and 20% for testing. The classification models were built based on training data and evaluated on the testing data.

Three state-of-the-art XAI/IML methods are used as the baselines to evaluate our proposed method: (1) the CEM method from Dhurandhar et al. (2018), (2) the CEGP method from Van Looveren and Klaise (2021), and (3) the brute force method in FAT from Sokol et al. (2022). These baseline methods are discussed in the Related Work section. Guidotti (2022) conducted a set of experiments to compare more than a dozen counterfactual explanation methods with a number of performance measures. The results show that there is not a single dominant winner, but the above three methods are among the better ones. Therefore, we selected them as the baselines in our evaluation study. However, none of them addresses the problem of incorrect predictions made by black box models, as discussed in our literature review.

Performance Measures

We evaluate the performance of the baseline and proposed methods using three commonly used measures. The first measure is classification *accuracy*, which is a commonly used measure for classification performance. Classification accuracy is computed on the test data.

The second measure is *plausibility*, which measures the distance between a counterfactual and the instances in the training dataset (Guidotti 2022). A close counterfactual to the training set is more plausible for explanation. So, the plausibility is calculated by the distance between a counterfactual instance and the nearest instance in the training dataset.

The third measure is *sparsity*, which is measured by the number of features that are different between the new instance and its counterfactual instance divided by the total number of features (Guidotti 2022; Verma et al. 2022). A smaller value in sparsity is desirable because it is easier to explain the differences between the new instance and the counterfactual in this case.

Experimental Results

Table 2 shows the results of classification accuracy. The accuracy with the baseline explanation methods are the same as those with the classification models (i.e., neural networks and random forest models), because they do not make any changes to the prediction results. The proposed method produces higher classification accuracy. This is understandable because the proposed method is able to correct some misclassified instances and thus improve the classification accuracy.

Dataset	Classification Model	Accuracy of Baselines	Accuracy of Proposed Method
Diabetes	Neural Networks	80.08 %	82.63 %
	Random Forest	75.32 %	79.44 %
FICO	Neural Networks	67.06 %	69.32 %
	Random Forest	68.16 %	70.28 %

Table 2. Results of Classification Accuracy

The plausibility results are reported in Table 3. A smaller value in this measure is preferred because it indicates that the counterfactual explanation is more plausible. It can be seen that the proposed method outperforms all baseline methods in every scenario. The CEM method does not support the random forest model. The search-based FAT method has a weak performance as it does not utilize the reference training data to reach the counterfactuals. In contrast, our proposed method, which considers trust scores, ensures that all generated counterfactuals are located in close proximity to the training data, leading to more plausible explanations.

Dataset	Classification Model	CEM	CEGP	FAT	Proposed Method
Diabetes	Neural Networks	0.249	0.217	0.330	0.204
	Random Forest	NA	0.332	0.403	0.266
FICO	Neural Networks	0.262	0.245	0.278	0.215
	Random Forest	NA	0.306	0.400	0.293

Table 3. Plausibility Results

Table 4 shows the results of sparsity. A smaller value in sparsity is desirable because the explanation is simpler in this case. The proposed method is the best performer in one case where the random forest model is used for the FICO data. It is the second best in the other three scenarios. CEGP is the best performer in two scenarios with the FICO data but is ranked the third in the other two scenarios with the Diabetes data. On the other hand, FAT is the best in two scenarios with the Diabetes data but ranked the third in the other two scenarios with the FICO data. CEM is either the worst or not applicable. In short, the proposed method performs very well in terms of sparsity. This suggests that the counterfactuals generated by the proposed method generally involve fewer changes and thus are easier to explain than the baseline methods.

Dataset	Classification Model	CEM	CEGP	FAT	Proposed Method
Diabetes	Neural Networks	0.321	0.284	0.215	0.223
	Random Forest	NA	0.265	0.216	0.221
FICO	Neural Networks	0.092	0.058	0.087	0.071
	Random Forest	NA	0.055	0.067	0.052

Table 4. Sparsity Results

Conclusion and Future Plans

We have proposed a novel method for generating counterfactual explanations for the predictions of black box models. The proposed method is designed to rectify and explain incorrect predictions by these models. This is a preliminary and ongoing research. We plan to perform more comprehensive experimental evaluation with additional baselines and evaluation metrics. We also plan to compare the proposed method, which uses black box models, with the white-box models that are inherently interpretable. We will also examine the fairness issue in counterfactual explanations, ensuring that explanations and decisions are fair and consistent to all stakeholders.

We have used trust score to help to determine if a new instance is misclassified or not. However, because a new instance does not have a true class label, there is no guarantee that the trust score can capture all actual misclassifications. To increase the trust score's reliability, we plan to develop a procedure for training and validating trust score. There are some hyperparameters for computing trust scores (Jiang et al. 2018), which we have not explored in this preliminary study. We can use training data, which have the true labels, to tune the hyperparameters to enhance the effectiveness of the trust score for finding the ground truth.

References

Abid, A., Yuksekgonul, M., and Zou, J. 2022. "Meaningfully Debugging Model Mistakes Using Conceptual Counterfactual Explanations," in *Proceedings of the 39th International Conference on Machine Learning*. Baltimore, Maryland, pp. 66-88.

Adadi, A., and Berrada, M. 2018. "Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI)," *IEEE Access* (6), pp. 52138-52160.

de Bie, K., Lucic, A., and Haned, H. 2021. "To Trust or Not to Trust a Regressor: Estimating and Explaining Trustworthiness of Regression Predictions," arXiv:2104.06982.

Delaney, E., Greene, D., and Keane, M.T. 2021. "Uncertainty Estimation and Out-of-Distribution Detection for Counterfactual Explanations: Pitfalls and Solutions," *International Conference on Machine Learning Workshop on Algorithmic Recourse*. arXiv:2107.09734

Dhurandhar, A., Chen, P.Y., Luss, R., Tu, C.C., Ting, P., Shanmugam, K., and Das, P. 2018. "Explanations Based on the Missing: Towards Contrastive Explanations with Pertinent Negatives," *Advances in Neural Information Processing Systems* (31), pp. 590-601.

Du, M., Liu, N., and Hu, X. 2020. "Techniques for Interpretable Machine Learning," *Communications of the ACM* (63:1), pp. 68-77.

Fernández, R.R., De Diego, I.M., Aceña, V., Fernández-Isabel, A., and Moguerza, J.M. 2020. Random Forest Explainability Using Counterfactual Sets," *Information Fusion* (63), pp. 196-207.

Fernández-Loría, C., Provost, F., and Han, X. 2022. "Explaining Data-Driven Decisions Made By AI Systems: The Counterfactual Approach," *MIS Quarterly* (46:3), pp. 1635-1660.

Guidotti, R., Monreale, A., Giannotti, F., Pedreschi, D., Ruggieri, S., and Turini, F. 2019. "Factual and Counterfactual Explanations for Black Box Decision Making," *IEEE Intelligent Systems* (34:6), pp.14-23.

Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., and Pedreschi, D. 2018. "A Survey of Methods for Explaining Black Box Models," *ACM Computing Surveys* (51:5), pp. 1-42.

Guidotti, R. 2022. "Counterfactual Explanations and How to Find Them: Literature Review and Benchmarking," *Data Mining and Knowledge Discovery* (April 28), pp. 1-55.

Jiang, H., Kim, B., Guan, M., and Gupta, M. 2018. "To Trust or Not to Trust a Classifier," *Advances in Neural Information Processing Systems* (31), pp. 5541-5552.

Kim, B., Srinivasan, K., Kong, S.H., Kim, J.H. Shin, C.S., and Ram, S. 2023. "ROLEX: A Novel Method for Interpretable Machine Learning using Robust Local Explanations," MIS Quarterly, Forthcoming.

Lundberg, S.M., and Lee, S.I. 2017. "A Unified Approach to Interpreting Model Predictions," *Advances in Neural Information Processing Systems* (30), pp. 4765-4774.

Martens, D., and Provost, F. 2014. "Explaining Data-Driven Document Classifications," MIS Quarterly (38:1), pp.73-99.

Molnar, C (2023) *Interpretable Machine Learning: A Guide For Making Black Box Models Explainable*. Accessed April 1, 2023, https://christophm.github.io/interpretable-ml-book/index.html.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., and Vanderplas, J. 2011. "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research* (12), pp. 2825-2830.

Ribeiro, M.T., Singh, S., and Guestrin, C. 2016. "'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*. San Francisco, CA, pp. 1135-1144.

Rudin, C. 2019. "Stop Explaining Black Box Machine Learning Models for High Stakes Decisions and Use Interpretable Models Instead," *Nature Machine Intelligence* (1:5), pp. 206-215.

Sokol, K., Santos-Rodriguez, R. and Flach, P., 2022. "FAT Forensics: A Python Toolbox For Algorithmic Fairness, Accountability And Transparency," *Software Impacts* (14), Article 100406, 6 pages.

Stepin, I., Alonso, J.M., Catala, A., and Pereira-Fariña, M. 2021. "A Survey of Contrastive And Counterfactual Explanation Generation Methods for Explainable Artificial Intelligence," *IEEE Access*, (9), pp.11974-12001.

Verma, S., Boonsanong, V., Hoang, M., Hines, K.E., Dickerson, J.P., and Shah, C. 2022. "Counterfactual Explanations and Algorithmic Recourses for Machine Learning: A Review," arXiv:2010.10596.

Vermeire, T., Brughmans, D., Goethals, S., de Oliveira, R.M.B., and Martens, D. 2022. Explainable *image classification with evidence counterfactual*," *Pattern Analysis and Applications* (25:2), pp. 315-335.

van der Waa, J., Robeer, M., van Diggelen, J., Brinkhuis, M., and Neerincx, M. 2018. Contrastive Explanations with Local Foil Trees," arXiv:1806.07470.

Van Looveren, A., and Klaise, J. 2021. "Interpretable Counterfactual Explanations Guided by Prototypes," in *Machine Learning and Knowledge Discovery in Databases. Research Track. ECML PKDD 2021. Lecture Notes in Computer Science* (12976), N. Oliver, F. Pérez-Cruz, S. Kramer, J. Read, and J.A. Lozano (eds.), Cham, Switzerland: Springer, pp. 650–665.

Wachter S, Mittelstadt B, Russell C. 2018. "Counterfactual Explanations Without Opening the Black Box: Automated Decisions and the GDPR," *Harvard Journal of Law & Technology* (31:2), pp. 841-887.

Wang, T., He, C., Jin, F., and Hu, Y.J. 2022. "Evaluating the Effectiveness of Marketing Campaigns for Malls Using a Novel Interpretable Machine Learning Model," *Information Systems Research* (33:2), pp.659-677.