CSSE - An Agnostic Method of Counterfactual, Selected, and Social Explanations for Classification Models

Marcelo de Sousa Balbino[a,b] (marcelobalbino@gmail.com), Luis Enrique Zárate Gálvez[a] (zarate@pucminas.br), Cristiane Neri Nobre[a] (nobre@pucminas.br)

[a] Pontifical Catholic University of Minas Gerais - PUC Minas - 500, Dom José Gaspar Street, Coração Eucarístico, Belo Horizonte, MG 30535-901, Brazil
[b] Federal Center for Technological Education of Minas Gerais - 121, 19 de Novembro Street, Centro Norte, Timóteo, MG 35180-008, Brazil

Corresponding Author:
Marcelo de Sousa Balbino
Pontifical Catholic University of Minas Gerais - PUC Minas - 500, Dom José Gaspar Street, Coração Eucarístico, Belo Horizonte, MG 30535-901, Brazil
Tel: (55) 31 98711-3402
Email: marcelobalbino@gmail.com

ARTICLE INFO

Keywords:
Counterfactual explanations
Explainable artificial intelligence
Interpretability
Machine learning
Genetic algorithm
Classification
ADHD

ABSTRACT

In some contexts, achieving high predictive capability may be sufficient for a machine learning model. However, in many scenarios, it is necessary to understand the model's decisions to increase confidence in the predictions and direct the actions to be taken based on them. Therefore, it is essential to provide interpretable models. However, some authors have pointed out the need to improve current interpretability methods to provide adequate explanations, especially for non-specialists in machine learning. The solution is to expand studies beyond computational issues to understand better how people receive explanations. Based on the literature, we identified three aspects to be considered in the explanations: contrastive, selected, and social. The counterfactual approach, contrastive in nature, inform the user of how the decision by the model can be altered through minimal changes to the input features. Given this, we introduce the Agnostic Method of Counterfactual, Selected, and Social Explanations (CSSE), capable of generating local explanations for classification models using a genetic algorithm. Thus, as contributions, we highlight that the CSSE offers counterfactual explanations from learning models, presents explanations with diversity, without prolixity, and allows the user to restrict the features that appear in the explanation (actionability), besides other parameterization options for the user to communicate their preferences. A particular novelty of our work is the possibility for the user to adjust the importance he will give to sparsity (minimum number of changes) or similarity (minimizing the distance). Furthermore, we indicate other possibilities for the actionability functionality, inherently used to lock immutable features, allowing users to block features according to their interests or expertise. These resources can help the user obtain explanations more targeted to their objective and advance further in interpretability, considering computational and social aspects in generating explanations. The experiments showed that CSSE presents relevant results compared to some existing approaches. The work also includes a case study of predicting the academic performance of children and adolescents with ADHD, in which we applied the CSSE. Thus, the proposed method advances interpretability by offering explanations aimed at the end user, which can generate greater acceptance, confidence, and understanding regarding the models' decisions. The method implementation is available at `https://github.com/marcelobalbino/CSSE`.

1. Introduction

Machine learning (ML) systems exist in several high-impact domains, products, and services. The models have supported issues where a correct decision is essential. Given this, research has been directed toward advancing the predictive capacity of the models. However, improved predictive performance has been achieved at the expense of greater complexity and less transparency in decisions, especially in black box models (Abdul, Vermeulen, Wang, Lim, & Kankanhalli, 2018; Du, Liu, & Hu, 2019). El Shawi, Sherif, Al-Mallah, and Sakr (2019) also show this trade-off between performance and transparency and highlight how limitations in interpretability undermine confidence in the model and its consequent use.

Restrictions on the model interpretability generate numerous problems for those involved. For users, minority groups can be discriminated against, injustices may not be perceived, and the harmed people still have few resources to argue. Furthermore, in most cases, the entities that own the systems cannot explain how the decisions were made due to the opacity of the systems. In high-risk scenarios, such as healthcare, the decision-maker feels insecure without explaining the model's results (Carvalho, Pereira, & Cardoso, 2019; Tjoa & Guan, 2020). Therefore, these systems need to allow developers and users (experts or not) to understand the system's decisions better, increasing confidence in the results and, when necessary, allowing incorrect conclusions to be noticed. Therefore, depending on the context, it is essential to provide interpretable models.

The focus on improving the interpretability of ML models has intensified recently, but there are points to be improved. The format of the existing explanations would be satisfactory to developers and researchers, for example, but they are only sometimes appropriate for non-specialists in ML (Du et al., 2019; Molnar, 2020). Moreover, Karim, Mishra, Newton, and Sattar (2018) highlight the importance of considering domain-specific explanations and approximating how people interpret them. Thus, there is a demand for investments in methods of interpretability that generate outputs with greater understanding from the user's point of view.

Miller (2019) also presents the need for explanations to be user-oriented. The author states that people appear cognitively programmed to process contrastive explanations, so a non-specialist will find contrastive explanations

ORCID(s):

more intuitive and valuable. To understand contrastive explanations, suppose a disease prediction model. Individuals wish to be included in the low-risk class for the disease. If the model classifies a particular instance into the high-risk type, the natural question is: "What do I need to do differently to get a favorable result next time?" Such a scenario refers to contrasting explanations or counterfactual cases.

According to Stepin, Alonso, Catala, and Pereira-Fariña (2021), counterfactual explanations are naturally considered contrastive but specify minimal changes needed in the input for a contrastive output to be obtained. However, in the context of Artificial Intelligence (AI), contrastive and counterfactual are often treated as similar or equivalent.

Miller (2019) also highlights two other elements to promote user-oriented explanations. First, explanations should be selected (providing ways for the user to choose aspects relevant to their context) and social (as part of an explainers/receiver communication of the explanation and generating knowledge). These elements highlighted by the author are based on relevant research in philosophy, psychology/cognitive science, and social psychology that study these topics in depth.

Thus, given a classification model p and an original instance x, the problem of explaining to the end-user the decision $p(x) = y$ is presented. For this purpose, we developed an Agnostic[1] Method of Counterfactual, Selected, and Social Explanations (CSSE) to generate explanations considering the aspects highlighted by Miller (2019) for an end-user-oriented approach. The method generates k counterfactual examples $c_1, c_2, ..., c_k$ as close as possible to x in which $p(c_i) = y'$, to $y' \neq y$. From each counterfactual, a local explanation is extracted consisting of the differences between x and c_i, that is, the set of minimal changes required in the features of x that make it change from class y to y'. Note that the counterfactual c_i does not necessarily have to be in the original dataset. CSSE works with tabular data and classification problems with binary output.

According to Van Looveren and Klaise (2021), identifying counterfactual cases can be described as an optimization problem. Minimizing the changes in the original instance that can reverse the prediction output is necessary. More specifically, we need to minimize an objective function that encodes the desirable properties of the counterfactual example. In addition, ML models usually work with a dataset with many features, including continuous features that can take on infinity values. Therefore, the search space for the counterfactual solution tends to be extensive. The central insight of this formulation is the need to design an objective function that allows us to generate high-quality counterfactual examples.

Genetic Algorithms (GA) are robust stochastic evolutionary algorithms widely used to solve scientific and industrial optimization challenges (Ghorbani et al., 2019). Although not the only option, the GA is among the techniques that best deal with optimization problems with many variables (Shahab et al., 2021). Furthermore, evolutionary approaches have been successfully used in different ML-related situations (Chen, Lin, Ke, & Tsai, 2015; Hamdia, Zhuang, & Rabczuk, 2021; Kim, 2006; Tsai, Eberle, & Chu, 2013; Xue, Zhu, Liang, & Słowik, 2021; Zeebaree, Haron, Abdulazeez, & Zeebaree, 2017), including issues similar to the one addressed in this research (Derrac, García, & Herrera, 2012). In particular, Guidotti et al. (2019) obtained satisfactory results in generating counterfactuals with GA. Given this scenario, it is relevant to evaluate and explore the application of GA in solving this problem.

In the proposed method, at each generation, the GA seeks to generate counterfactuals that approximate the desirable class through minimal changes (*minimality*) in the original instance. Therefore, even if the counterfactual examples are generated from modifications to the original instance, it is essential to maintain *minimality*.

According to Guidotti (2022), *minimality* can have different meanings and depends on each application. One possibility is that the minimal changes in the original instance refer to the number of features changed to generate the counterfactuals. This notion is typically referred to in the literature as *sparsity*. However, minimizing the distance (considering a distance function) for the original instance is still possible. In this case, we seek a minor impact on each changed feature, even if more features are used. This second notion is called *similarity*[2] (or *proximity*). In our method, the user can define which idea of minimality he wants to apply by adjusting some parameters. The user can also compose the minimality combining both notions, even prioritizing one if interested.

Another essential aspect we want to deal with is the undesirable generation of prolix counterfactuals. According to Keane and Smyth (2020), most systems generate prolix counterfactuals. The prolix term refers to counterfactuals generated from increments or decrements in the same features without contributing new explanations. For example, suppose a loan system where the customer would have a salary of 300$ more to reverse a loan denial prediction. Of course, you can generate other counterfactuals for every $ incremental change in one's salary, but this would not be

[1] An agnostic explanation method is defined as one that is independent of the type of the original model (Carvalho et al., 2019).

[2] In this paper, we use the term *similarity* when dealing with the distance from the counterfactual to the original instance (measured by a distance function).

very helpful. Thus, CSSE overcomes this limitation of many existing methods and presents k explanations (where the user defines k) with diversity (different sets of features) e without prolixity.

CSSE also allows the use of a list, called $static\_list$, which makes it possible to restrict the features to appear in the explanation. In the literature, this property is commonly called *actionability* and is related to preventing the use of inherently immutable features (non-actionable) in the real world (Verma et al., 2020). In our approach, we expand the notion of actionability. In addition to blocking inherently immutable features, we indicate other possibilities for using this list, such as limiting explanations to features of interest (or expertise) specific to a given user. Even not using this list brings a particular perspective to the result to be presented. The method also has other parameterization options that aim to adapt its output to each application and purpose.

In this way, the functionalities provided by CSSE aim to meet the guidelines of selectivity and socialization of the explanations pointed out by Miller (2019). The objective is to provide the user with opportunities to communicate their preferences and needs so that the explanations are contextualized and relevant to a purpose of interest.

We evaluated CSSE quantitatively and qualitatively. In the quantitative scope, we performed experiments with CSSE to assess its ability to generate counterfactuals compared to methods WACH (Wachter, Mittelstadt, & Russell, 2017) and LORE (Guidotti et al., 2019). From a qualitative point of view, we compared the resources offered by CSSE with other relevant methods in the literature. We also analyzed such resources applied to real datasets. In addition, we present and discuss the application of the method in a case study of school performance prediction of children and adolescents with ADHD.

The main contribution and novelty of this work are to propose a method of explanations for the decisions of classification models with an approach oriented to the end-user, including non-ML specialists, based on the principles of the theory of Miller (2019), which includes computational and social sciences aspects in generating explanations. Following the guidelines indicated by this author, the proposed method contains resources that make its explanations contrastive, selected, and social. This means generating explanations with the high communicability inherent in counterfactual explanations and in which the user's context and objectives are considered. These are relevant advances considering current methods. Among the resources in the method, it is worth mentioning the possibility of defining the notion of *minimality* to be applied in the counterfactuals, generating *multiple* explanations with *diversity* and *without prolixity*, and the option for the user to restrict the list of features that the method can use in the explanation (*actionability*). Also, CSSE is *agnostic*, making it widely applicable.

Furthermore, we emphasize that the qualitative and quantitative comparative analyses showed that our approach has significant advances compared to other methods in the literature. Finally, we highlight the results' relevance by applying CSSE to the case study of performance prediction in children and adolescents with ADHD. For those involved in the problem, the results serve as an aid tool in decision-making. For the development of this work, the case study presents itself as a meaningful scenario for the experimentation and evaluation of possible potentialities and limitations of the proposed method.

This paper presents the following structure: Section 2 presents the main concepts related to explanations. Section 3 refers to related work, presents their contributions, and points out the differences with the method proposed in this work. Section 4 details the CSSE method, highlighting aspects of its implementation and presenting its resources. Section 5 shows experiments with the developed method and comparisons with other methods. Section 6 refers to a case study applying the CSSE in a context related to the academic performance of children and adolescents with ADHD. Finally, Section 7 presents the discussion and conclusion, with limitations and future work.

2. BACKGROUND

2.1. Interpretability and explanation

EXplainable AI (XAI) has gained the scientific community's attention due to the need for more transparent systems, whether due to ethical issues or lack of user trust. The objective is to contribute to creating methods that allow interpreting the models, preserving high levels of predictive performance (Adadi & Berrada, 2018; Miller, 2019). The interpretability methods present a new concept for ML solutions, including an explanation model (g) to the prediction model (p) (Figure 1) (Mokhtari, Higdon, & Başar, 2019).

However, research points to inadequate current interpretability methods in communicating with end-users (Du et al., 2019; Karim et al., 2018; Miller, 2019; Molnar, 2020). In this sense, Miller (2019) says that if we want to design and implement intelligent agents capable of providing explanations to people, it is essential to understand how humans define, generate, select, evaluate, and present explanations. However, most practices use researchers' intuitions of

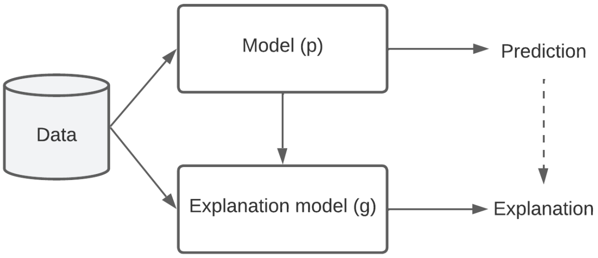


Figure 1: How an explanation model is used to interpret a prediction. Adapted from (Mokhtari et al., 2019).

what is considered a "good" explanation. The solution is to expand studies beyond computational issues. The author adds that a vast and mature body of work studies these topics in philosophy, psychology/cognitive science, and social psychology. Therefore, based on studies that consider computational and social sciences aspects, Miller (2019) lists three points that should be considered to build a genuine XAI: explanations are contrastive, selected, and social.

2.1.1. Contrastive/counterfactual explanations

Contrastive theories argue that causal explanations inevitably generate interest in a counterfactual case, a cause or event that has not occurred (Mittelstadt, Russell, & Wachter, 2019). Mainly when the prediction refers to an undesired class, it is natural to want to understand "why P instead of Q".

According to Miller (2019), contrastive explanations are more efficient in the search for confidence in the decisions of a model because people better understand explanations generated from specific facts (events, properties, judgments, etc.) instead of trying to understand and build general theories.

In modern ML, the algorithm's behavior can consist of many connected variables. Therefore, establishing an explanation logic that allows non-ML expert users to reason about the algorithm's decisions is challenging. In this sense, a great advantage of counterfactual explanations is that it is only necessary to understand the difference between the two cases (Wachter et al., 2017). The authors also highlight the contribution of counterfactual explanations with the "right of explanation" present in the General Data Protection Regulation (GDPR). Other forms of explanation with a more technical character may have little practical value for data subjects. On the other hand, counterfactual explanations allow us to understand the reasons for a decision, contest them and change future behavior for a better result.

2.1.2. Selected and social explanations

Multiple explanations are usually possible for a given event, assigning different causes. However, people are rarely interested in the complete causal chain of the event and will prefer an explanation that conveys valuable information for a particular context. Consequently, different stakeholders in this event may judge one explanation as more relevant. In addition, certain cognitive biases can also influence the receiver's preference in this evaluation. Such aspects show human selectivity regarding explanations (Miller, 2019; Mittelstadt et al., 2019).

Regarding the social aspect of explanations, this implies establishing an interactive process between the explainer/recipient of the explanation and the knowledge transfer. Therefore, the information must be appropriate to the recipient's beliefs and comprehension abilities (Mittelstadt et al., 2019).

Notably, the selectivity and contextualization of explanations contribute to their socialization. First, selectivity is a way of establishing interaction between the explainer and the receiver. Second, there is no knowledge transfer if the explanation has no meaning or relevance for the receiver. Furthermore, the counterfactual explanation, as an answer to a given why question, has a format that supports the idea of explanation as a conversation.

3. Related Work

Wachter et al. (2017) propose counterfactual explanations applied to models based on neural networks for two problems: prediction of academic performance in a law school and diabetes risk classification. The technique used for contrast generation was *"adverse disturbations"*. The authors' main interest is to discuss how counterfactual explanations can contribute to the restrictions imposed by the GDPR on automated decision-making. The authors concluded that counterfactual explanations could assist in solving some of the main problems regarding the right to explanation: 1) by being able to explain the internal logic of automated systems to ML experts and non-experts; 2) not requiring excessive disclosure of information about the internal logic of systems to the point of infringing on the rights of third parties, either concerning protected trade secrets or by violating the privacy of individuals whose data is contained in the training dataset.

Dhurandhar et al. (2018) proposed the *Contrastive Explanations Method* (CEM) specific for neural networks that explores the concept of positive and negative pertinents. A pertinent positive is a factor whose presence is minimally sufficient to justify the final classification. A negative pertinent is a factor whose absence is necessary to affirm the classification. The authors claim that the method proved effective in different domains, generating presumably easier-to-understand and more accurate explanations. Dhurandhar et al. (2018) point out that identifying pertinent negatives is particularly useful when close entries can generate different classifications (for example, distinguishing a diagnosis of flu or pneumonia). If the entries are very different, the pertinent positives are enough to characterize the entrance, as there are likely to be many pertinent negatives overwhelming the user.

Guidotti et al. (2019) propose LOcal Rule-based Explanation (LORE), a rule-based method that includes the factual and counterfactual approach. The method learns an interpretable local classifier (the decision tree classifier) through a set of instances neighboring the instance to be explained. Neighboring instances are generated by an ad-hoc GA and used as training data for learning the local decision tree. From this classifier, the method generates a rule that explains the factual reasons for the decision and a set of counterfactual rules that indicate changes in the original instance that would imply a different output. Guidotti et al. (2019) perform experiments comparing LORE with the method developed by (Wachter et al., 2017) to evaluate counterfactuals. The authors highlight the two main contributions of the work. First is the proposed explanation's high expressiveness since it provides evidence about why an instance received a specific label and counterfactuals that suggest what should be done to reverse the predicted result. Second, the local decision limit in the neighborhood of the instance to be explained is explored using a genetic algorithm capable of producing high-quality training data to learn the local decision tree.

Rathi (2019) presents a counterfactual explanation method supported by the SHAP approach (Lundberg & Lee, 2017). From a given instance, the basic idea is to obtain the features that have a negative impact through SHAP, that is, those that distance this instance from the counterfactual class to which one wishes to migrate. The counterfactuals result from altering the values of the features of negative impact. The author states that the evaluation criteria used differ from other research. Rathi (2019) measures the method's effectiveness by the total number of new counterfactual data points (not in the dataset) and the average number of counterfactuals generated by the original base instance. According to the author, this is a more appropriate assessment for the approach, as it considers how far the data distribution is far from the border of the decision. As a novelty, the author points out that the method is based on the SHAP and cites the agnostic character of the technique as an advantage. It also points out that the method closely aligns with the concept of counterfactual explanation because it changes only the resources that adversely affect the classification task.

Gomez, Holter, Yuan, and Bertini (2020) present the *Visual Counterfactual Explanations* (ViCE), a method of explanations whose differentials are the presentation of counterfactuals in a visual interface and interactive resources for users to explore data and model. According to the authors, the method includes a first graphical interface that can display this type of explanation effectively and coherently. Moreover, the technique allows the user to indicate immutable features. ViCE adopts the following heuristic to obtain the counterfactuals: starting from the original values of the instance to be explained, it moves the value of each of the unlocked features above and below the current one and chooses the one eliciting the most considerable change in the model's output (in the direction of the opposite class). It then takes the maximum change on all unlocked features and uses that as input for the next iteration. This procedure continues until the modified instance crosses the decision boundary or until the constraints can no longer be satisfied.

Mothilal, Sharma, and Tan (2020) developed *Diverse Counterfactual Explanations* (DICE), a method that emphasizes the diversity of examples and similarity to the original instance. From a trained ML model (p) and an original instance (x), k counterfactual examples are generated. The idea is that diverse counterfactual examples increase the chances that at least one will be actionable to the user. The method uses a loss function that combines all the counterfactuals. The function is optimized using a descending gradient. The authors compared the performance of DICE with LIME (Ribeiro, Singh, & Guestrin, 2016). The results showed that the examples generated by DICE are as good as those obtained by LIME. A limitation of the method concerns the fact that it needs to know the gradient of the learning model. Furthermore, the authors emphasize the importance of building techniques that can be used with different black box models. Finally, the authors recognize the need to expand user interaction to meet their restrictions and preferences.

3.1. Differences between the methods

In this section, we conduct a qualitative analysis of CSSE, comparing it to related work previously described. Although all are methods related to counterfactual explanations, they propose different approaches. Wachter et al. (2017) focus on the contributions of counterfactuals explanations to GDPR. Dhurandhar et al. (2018) exploit the generation of counterfactuals with pertinent negatives. Guidotti et al. (2019) propose an outcome based on rules factual and counterfactuals. Rathi (2019) highlights the counterfactuals generated with the SHAP approach and the generation of counterfactuals by changes only to features that adversely impact the classification task. The novelty of the proposal by Gomez et al. (2020) is the visual presentation of the explanations. Mothilal et al. (2020) prioritize the diversity of counterfactuals.

On the other hand, CSSE emphasizes the generation of explanations aimed at the end user, including non-ML specialists. This approach aligns with the work presented by Miller (2019), which focused on social science guidelines for greater acceptance, trust, and understanding of users regarding explanations (see more details in Section 4). Our work prioritizes identifying and including resources that make the explanations understandable, relevant, and contextualized for each user and application.

Other factors differentiate us from the approaches presented in this section, as shown in Table 1. CSSE allows the generation of multiple counterfactuals, which is impossible in WACH, CEM, CFSHAP, and ViCE. This limitation restricts the user's choice possibilities. Furthermore, by being agnostic, CSSE becomes widely applicable. This characteristic is not present in WACH, CEM, and DICE.

Table 1
Comparison of resources present in counterfactual methods.

Method	Multiple	Agnostic	Actionability
WACH* (Wachter et al., 2017)			
CEM (Dhurandhar et al., 2018)			
LORE (Guidotti et al., 2019)	✓	✓	✓
CFSHAP* (Rathi, 2019)		✓	
ViCE (Gomez et al., 2020)		✓	✓
DICE (Mothilal et al., 2020)	✓		✓
CSSE (proposed method)	✓	✓	✓

*We used the same names found in Guidotti (2022)'s article since we did not find the name of the methods in the original works.

Another resource of CSSE that aims to contribute to the selectivity and socialization of the explanations is the possibility for the user to indicate features that should not be included in the explanations. This functionality is absent in the WACH, CEM, and CFSHAP. As with CSS, LORE, ViCE, and DICE allow you to restrict the features that can be changed (actionable). However, in our work, the focus on Miller (2019) guidelines leads us to expand the notion of actionability. In addition to blocking immutable features, we suggest using it to limit the explanation features to those of interest (or expertise) to a given user. A more detailed discussion of this resource can be found in Section 4.3.

Therefore, considering the methods described in this section, only LORE (Guidotti et al., 2019) has the resources present in CSSE. However, in generating multiple counterfactuals, our method allows the user to define the number of explanations he wants to visualize. On the other hand, in LORE, as it is an approach based on rules extracted from a decision tree, the quantity of non-prolix counterfactuals is limited to the number of generated rules.

Thus, since LORE is the closest method to CSSE, we performed some experiments to compare them quantitatively. We include WACH in this comparison because it was one of the first works to propose counterfactual explanations and is probably the most famous. We present these experiments in Section 5.

4. CSSE for generating counterfactual examples

This section details the proposed CSSE. The solution is based on the counterfactual approach, for which we present the following definition:

Definition 1. *Consider the classification model p and an original instance x, where $p(x) = y$. We say that c is a counterfactual of x if c is as close as possible to x and $p(c) = y'$, to $y' \neq y$. We emphasize that c is a synthetic instance, i.e., not necessarily present in the original dataset. Figure 2 illustrates the notion of counterfactual, where c is the shortest distance counterfactual (d) for the original instance x, considering features f1 and f2.*

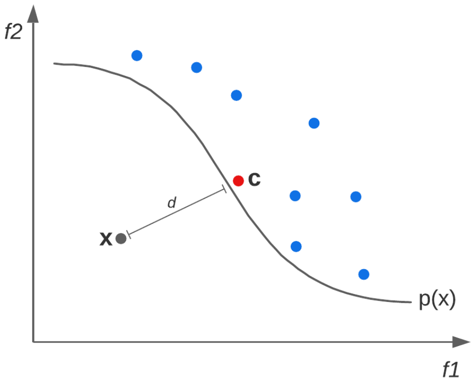
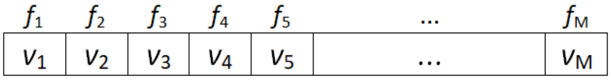


Figure 2: Counterfactual illustration.

Our problem consists of generating a set of explanations $e_1, e_2, ..., e_k$ for the decision $p(x) = y$, where k corresponds to the number of user-defined explanations. In this sense, each explanation e_i (where $i \leq k$) is generated through the modified W_i features in x to generate c_i. More specifically, e_i consists of the set of W_i triple $(f_j, v_j, v_j^{'})$, where v_j and $v_j^{'}$ correspond to the values of feature f_j in x and c_i, respectively, and $v_j \neq v_j^{'}$. In other words, each explanation e_i consists of the set of changes in the feature values of x that would make it migrate from class y to y'.

CSSE uses an ad-hoc GA to find the counterfactuals. In Figure 3, we present an example of an individual (chromosome) used in the GA. This individual consists of an array of M numeric variables, where M is the number of input features from the dataset.



f_1	f_2	f_3	f_4	f_5	...	f_M
v_1	v_2	v_3	v_4	v_5	...	v_M

Figure 3: Chromosome design.

Figure 4 presents an overview of CSSE. The proposed method requires the user to provide:

1. p: the trained classification model.
2. x: instance of interest/original.
3. ds: with the dataset used in the classification model, we can identify the possible values that each feature can assume in the explanation. It is a way to obtain the feature domain without the need for user interventions.
4. k: the number of counterfactual examples, and consequently of explanations, that the user wants to receive. In this way, the user can evaluate the example that interests him most.
5. $static\_list$: optionally, the user can define a list of features that cannot be modified and, therefore, will not appear in the counterfactual explanation. Section 4.3 discusses possible usage perspectives for $static\_list$.

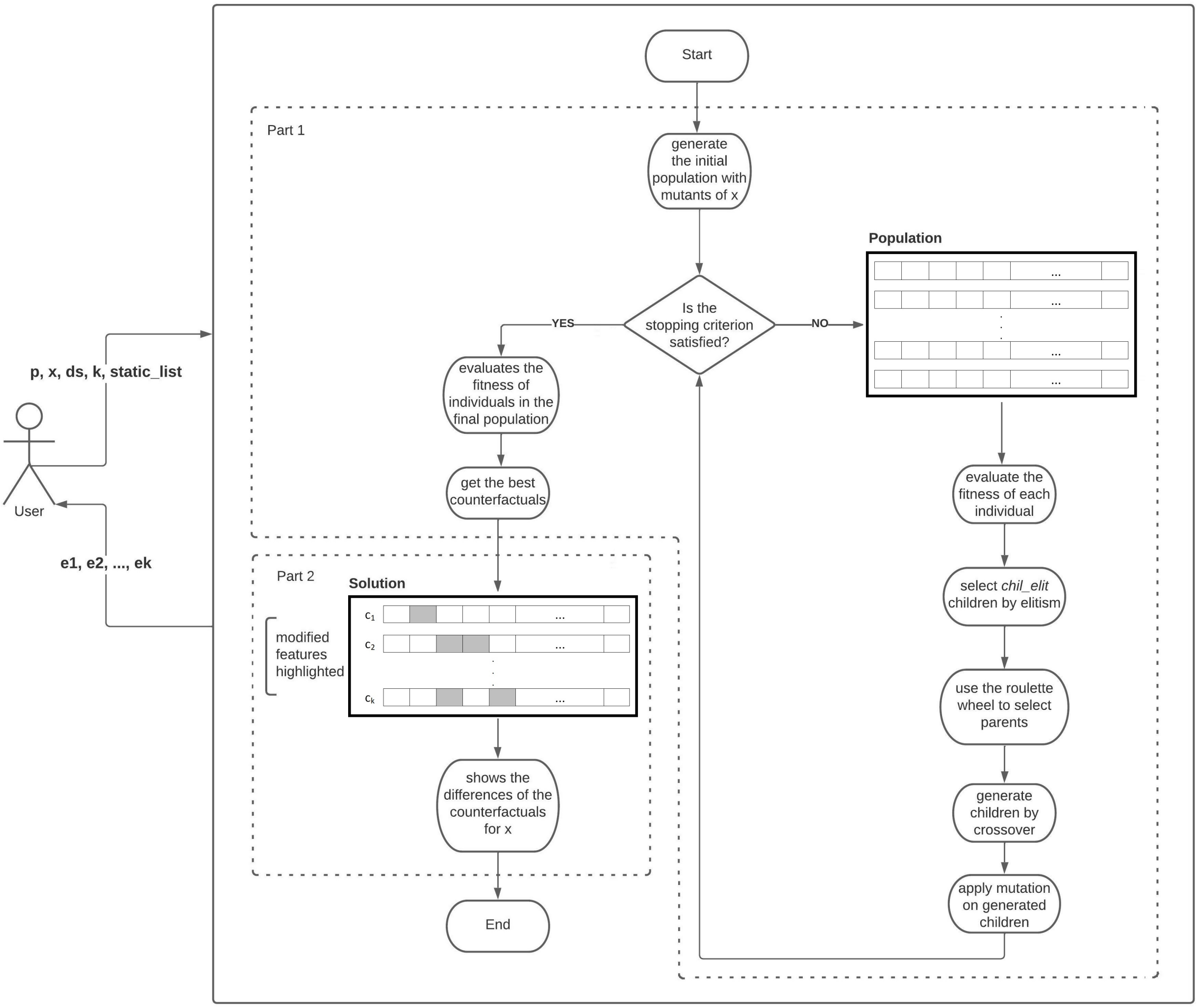

Start
Part 1
generate the initial population with mutants of x
Population
...
Is the stopping criterion satisfied?
YES
NO
p, x, ds, k, static_list
evaluates the fitness of individuals in the final population
User
get the best counterfactuals
evaluate the fitness of each individual
e1, e2, ..., ek
Part 2
Solution
select chil_elit children by elitism
c_1
c_2
c_k
modified features highlighted
use the roulette wheel to select parents
shows the differences of the counterfactuals for x
generate children by crossover
apply mutation on generated children
End

Figure 4: Overview of CSSE.

CSSE includes two main parts. *Part 1* consists of the GA itself. The strategy is that, at each generation, the GA seeks counterfactuals close to the desired class (y'), with minimal changes from the original instance (x). We detail the objective function in Section 4.1. Then, using the best counterfactuals found by GA, the method generates the explanations that are presented to the user (*Part 2*). We present the pseudocode of the proposed method in section 4.2.

CSSE includes elements whose purpose is to approximate the output generated by the method to the guidelines highlighted by Miller (2019) about how people best receive explanations. We present such resources in Section 4.3.

4.1. The GA objective function

Through GA, the proposed approach seeks to optimize the sparsity/similarity of counterfactuals, considering the restriction of these belonging to the desired class. For that, we use the objective function described in Equation 1.

$$C(x) = arg \min_{c} \lambda_1 \; x\_dist(c, x) + \; \lambda_2 \; x\_nchg(c, x) + \; y\_dist(p(c), y') \tag{1}$$

where $x\_dist(c, x)$ is the distance function between the original instance x and the counterfactual c, $x\_nchg(c, x)$ is the number of features modified in x to generate c, and $y\_dist(p(c), y')$ calculates the distance of c for the desired class

y'. Thus, x_dist e x_nchg keeps the counterfactual c close to the original instance, and y_dist aims to drive c towards the desired class.

The parameters λ_1 and λ_2 assign weights to the function parts that measure the distance to the original instance and the number of changes required in the original instance, respectively. The user can adjust the parameters for each problem. We discuss the setting of these parameters in Section 5.3.

To assess the minimality of each counterfactual for x, the GA uses a distance function (Euclidean, for example) and the number of modified features. It is necessary to encode the categorical features to numerical ones to calculate the distance between the counterfactuals and the original instance. We consider it more appropriate for the user to make the necessary conversions to represent the values of the features better.

The calculation of $y\_dist(p(c), y')$ consists of the difference $\left|p(c) - y'\right|$, where $p(c)$ corresponds to the *output value* predicted by the classification model for the candidate a counterfactual c, and y' corresponds to the boundary of classes. However, the distance is calculated only for counterfactual candidates that have yet to reverse the classification. We assign zero distance ($y\_dist = 0$) for valid counterfactuals, i.e., those that exceed the p's threshold. Thus, we avoid encouraging significant changes between c and the original instance.

We apply the normalization process before calculating the distances between the original instance and each counterfactual. Another need for normalization refers to the metrics that make up the objective function (Equation 1) since its values refer to metrics of different natures and meanings. In both cases, the process causes the values to vary [0, 1].

4.2. CSSE pseudocode

Algorithm 1 details the operations performed by the CSSE. Thus, the proposed method consists of the following implementation:

Algorithm 1: CSSE Pseudocode

```
   Input: $p$, $x$, $ds$, $k$, $static\_list$
   Output: $C(x_i)$, $i = 1, 2, \cdots$ ,k
1  $t \leftarrow 0$;
2  $Pop(0) \leftarrow InitialPopulation(x, ds, static\_list)$;
3  while not Number of Generations do
4      $Fitness(Pop(t), x, p)$;
5      $Pop(t+1) \leftarrow Elitism(Pop(t))$;
6      $parents \leftarrow Selection(Pop(t))$;
7      $children \leftarrow Crossover(parents)$;
8      $children \leftarrow Mutation(children, ds, static\_list)$;
9      $Pop(t+1) \leftarrow Pop(t+1) + children$;
10     $t \leftarrow t + 1$;
11 end while
12 $C \leftarrow GetBest(Pop(t), k)$;
13 $ShowChanges(C, x)$;
```

1. *InitialPopulation(x, ds, static_list)* (Line 2): the GA population is initialized with mutants of the original instance x. pop_size mutants are created by modifying one feature drawn at random (features in the $static\_list$ are not modified). These are the parents of the first generation.

2. $Fitness(Pop(t), x, p)$ (Line 4): the function evaluates the sparsity/similarity of each individual of the population Pop with the original instance x and their proximity to the desired class through the classification model p. We detail the objective function in Section 4.1.

3. $Elitism(Pop(t))$ (Line 5): then $chil\_elit$ children are selected by elitism ($chil\_elit = pop\_size * per\_elit$). The remaining ($pop\_size - chil\_elit$) children are generated by genetic operations.

4. $Selection(Pop(t))$ (Line 6): we use the roulette wheel to select parents to privilege individuals with the highest evaluation function without neglecting those with the lowest evaluation function.

5. $Crossover(parents)$ (Line 7): we use a one-point crossover operator (according to crossover probability).

6. $Mutation(children, ds, static\_list)$ (Line 8): mutants are generated by the same strategy presented in the initial population (according to mutation probability).

7. $GetBest(Pop(t), k)$ (Line 12): when the stop condition is reached, the final population is ordered considering the evaluation function. Then, as in “Output”, the k best counterfactuals that reach the desired class are returned.

8. $ShowChanges(C, x)$ (Line 13): the k counterfactual examples $c_1, c_2, ..., c_k$ obtained are compared to the original instance x to identify which features differ. The differences between c_i and x constitute one of the possible sets of changes necessary for the chosen entry to change to the desired class. Sections 5.1 and 6.4 include examples of the output generated by the method.

Table 2 shows GA parameters. We developed the method implementation using Python, which is available at `https://github.com/marcelobalbino/CSSE`.

Table 2
Genetic Algorithm control parameters.

Parameter	Description	Default value
num_gen	number of generations	30
pop_size	population size	100
λ_1, λ_2	objective function weights	1, 1
per_elit	percentage of elitism	0.1
$cros\_proba$	crossover probability	0.8
mut_proba	mutation probability	0.1

4.3. User-oriented explanations

We propose to provide explanations according to the guidelines highlighted by Miller (2019), according to which contrastive, selected, and social explanations are more understandable and valuable to the end user. Since the proposed method is based on contrastive explanations, our challenge is to include elements that contribute to the principles of selectivity and socialization of explanations. This means including resources that allow the user to interact with the method and consider their context and intentions, thus making the explanation relevant and knowledge-generating.

In this sense, the method allows the user to determine k of explanations to be received. Thus, it is possible to select the explanations that matter, whether to understand the model’s decision or perform an action. Regarding generating multiple counterfactuals, it is necessary to consider that single-objective GAs are often used to find a single solution. Therefore, it is common to find close solutions when searching for multiple solutions, resulting in counterfactual explanations with little diversity and thus limiting the user’s choice. For this reason, the method presents the best solution, and the others $k - 1$ shown are selected, following the order, so they contain a different set of features. Thus, we offer alternatives for the user with diversity and without prolixity.

Another resource available in CSSE is the $static\_list$, through which the user can restrict the possible features for the explanations. According to the needs of each user, $static\_list$ can bring different perspectives of use for the method. Among them, we can highlight the following:

1. Point out features that cannot be modified in practice, such as a person’s gender, for example. In this case, the user is interested in the practical application of the explanations.

2. Suppose the user is interested in explanations that support short-term decision-making. In this case, it may be necessary to remove features that the user believes cannot be changed immediately, such as raising the level of education or changing a person’s social class.

3. Restrict the explanation features to those of interest to a given user, making the explanation close to a specific context and matching their capabilities and area of expertise. For example, suppose a problem that involves multidisciplinary attributes. For a doctor, an explanation that is restricted to features related to his area may be more useful so that the explanation makes sense to him and includes elements on which he can act.

4. Suppose the user does not use $static\_list$. Then, if there are no restrictions, it is possible to view the features on the border of decisions, which helps their understanding. In addition, other aspects can be highlighted. Hypothetically, if in a bank credit approval model, the explanation shows that changing the individual's gender or race would change the prediction, this may indicate a discriminatory decision bias.

By definition, counterfactual examples must be generated through minimal changes to the original instance. As a result, we have more understandable explanations and are more likely to be used in practice by the user. However, this notion of minimality can vary according to the application. We have two possibilities: minimize the number of features to be changed (*sparsity*) or the distance between the counterfactual and the original instance (*similarity*) (Guidotti, 2022). Given this, the proposed method allows the user to define the notion of minimality that he wants to apply. With the parameters λ_1 and λ_2 presented in Section 4.1, it is possible to adjust the method to indicate the notion of minimality to be used, prioritizing sparsity or similarity or even seeking explanations that include both concepts.

Thus, both the GA parameterization options and the inputs provided to the method allow the user to communicate aspects relevant to their context. Adding these resources to the communication capacity inherent in counterfactual explanations, we believe that the method generates explanations that increase the user's understanding and confidence about ML models.

5. Experimental results

To evaluate the performance of CSSE, we present the experiments applying the method in four different contexts: 1) *qualitative analysis*, obtaining the counterfactual of an individual instance (Section 5.1); 2) *quantitative analysis*, evaluating the method's ability to generate counterfactuals through minimal changes in the original instance (Section 5.2). For this, we compare the performance obtained with CSSE and LORE (Guidotti et al., 2019), and WACH (Wachter et al., 2017); 3) Finally, we perform a sensitivity analysis to measure the influence of the objective function parameters on the results produced by the method (Section 5.3).

To carry out these experiments, we used the same datasets used by Guidotti et al. (2019):

1. The Compas[3] dataset contains criminal history, jail and prison time, demographics data, and three *Compas scores* (Risk of Recidivism, Risk of Violence, and Risk of Failure to Appear) of 7214 Broward County defendants from 2013 and 2014. Compas scores for each defendant ranged from 1 to 10, with ten being the highest risk. In the experiments, scores of 1 to 6 were labeled as "Medium-Low" risk, and 7 to 10 were labeled as "High" risk. Considering the Risk of Recidivism, the base has 5219 defendants classified as "Medium-Low" and 1995 as "High". Guidotti et al. (2019) performed dataset transformations that resulted in 11 features in addition to the class.

2. The German Credit[4] dataset includes 1000 loan applicants. Each candidate is described by a set of 20 features in addition to the class. Each candidate is assigned a credit risk rating of "good" (0) or "bad" (1). In the dataset, 700 instances belong to the class of good credit candidates and 300 to the bad candidates.

For the quantitative comparison to be fair, we reproduce the scenario used by Guidotti et al. (2019): we use the same features, transformations, and base separation criterion (70% of the dataset to train the black box model and 30% as instances to be explained). In addition, as performed by the authors, we use the Random Forest algorithm using the *scikit-learn* library in Python with default parameters for both datasets.

[3] Available at https://github.com/propublica/compas-analysis. We describe the dataset features in Table 10 in Supplementary Materials.

[4] Available at https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data). We describe the dataset features in Table 11 in Supplementary Materials.

For the German credit dataset, the mean values and standard deviation of the *Precision*[5], *Recall*[6], and *F-measure*[7] are, respectively: 0.75 (0.07±), 0.69 (0.23±), and 0.70 (0.15±). For the Compas dataset, the mean values and standard deviation of the Precision, Recall, and F-measure are, respectively: 0.77 (0,07±), 0.71 (0.21±), and 0.73 (0.15±).

5.1. Qualitative analysis - obtaining the counterfactual for an individual instance

To exemplify the application of CSSE and analyze it qualitatively, we used the 'german' and chose an instance of credit risk "bad"(1). In addition, we decided to return three counterfactuals ($k = 3$), and we do not restrict any feature to be included in the explanation (empty *static_list*). Finally, we kept the default parameters shown in Table 2 for GA.

As presented in Table 3, the CSSE indicated that the customer would be classified as "good" for credit risk if he: 1) were 30 years old instead of 21 years old; or 2) increased the savings account/bonds; or 3) reduced the credit amount.

Table 3
Application result of CSSE with german dataset considering all features.

	Modified features		
	age	**savings**	**credit_amount**
Original instance: **class**: 1 (Bad)	21	1 (... <100 DM)	15653
Counterfactuals **class**: 0 (Good)	30	_	_
	_	2 (100 <= ... <500 DM)	_
	_	_	9157
No static feature			

The counterfactual explanation can be used to understand the model's local behavior, eventually followed by some decision-making. In this case, changing the customer's age, presented in Table 3, to reverse the credit rating to "good" would not be immediately feasible. Hypothetically, we could repeat the execution of the method, including feature *age* in *static_list*, so it would not be considered in the counterfactual search. Thus, we exemplify one of the method's resources. Table 4 presents the new explanation.

Table 4
Applying the method on the german dataset including "age" in *static_list*.

	Modified features		
	savings	**credit_amount**	**duration (months)**
Original instance: **class**: 1 (Bad)	1 (... <100 DM)	15653	60
Counterfactuals **class**: 0 (Good)	2 (100 <= ... <500 DM)	_	_
	_	9277	_
	_	_	33
Static feature: age			

Given the stochastic nature of GAs, we could have obtained different explanations, but the features *savings* and *credit amount* remained. The new execution included an explanation that indicates that reducing *duration in months* would change the credit applicant's risk rating to "good".

Tables 3 and 4 show some resources present in CSSE. We highlight the generation of three explanations in each execution by user choice ($k = 3$). We also notice that the counterfactual is generated in each case using a different feature (multiple, diverse, and non-prolix explanations). In the second case (Table 4), we utilized the *static_list* to prevent the use of the age, considering it a non-actionable feature. Regarding the minimality treatment, since we used

[5] $\text{Precision} = \frac{\text{TP}}{\text{TP+FP}}$

[6] $\text{Recall} = \frac{\text{TP}}{\text{TP+FN}}$

[7] $\text{F} - \text{Measure} = \frac{2\times\text{Precision}\times\text{Recall}}{\text{Precision+Recall}}$

the default parameters, we executed the method equally important for sparsity and similarity. Thus, CSSE searched for counterfactuals that minimize the number of changed features and the distance to the original instance with equal importance. However, it is clear by observing the explanations presented only sparsity since the method generated each counterfactual through changes in only one feature. Finally, we highlight the multiple explanations, with diversity, without prolixity, and the treatments of minimality and actionability present in the method as essential resources for our end-user approach.

5.2. Quantitative analysis - Comparison between methods CSSE, LORE and WACH

This section presents the comparison result between the CSSE, LORE (Guidotti et al., 2019), and WACH (Wachter et al., 2017) methods. For this, we use two performance indicators presented by Guidotti et al. (2019): the number of falsified conditions in counterfactual (*nf*), and the rate of agreement of black box and counterfactual decision for counterfactual instance (*c-hit*).

To understand how to obtain the value of *nf* and *c-hit*, it is essential to define the total number of possible counterfactuals for a given set of instances to be explained. Thus, given a set of N instances X_i to be explained, in which the user wants to obtain k counterfactuals for each instance, ideally, we would have $N \times k$ counterfactuals. However, one must consider that the method may not be able to generate the required k counterfactuals. Thus, we say that, for each X_i instance, NC_i counterfactuals are obtained, where $NC_i \leq k$, resulting in a total of counterfactuals TC (Equation 2):

$$TC = \sum_{i=1}^{N} NC_i \tag{2}$$

Thus, given that each counterfactual is generated through CH features changed in the original instance, we have that the *nf* is the mean of changes considering the TC counterfactuals. In this way, *nf* is associated with *sparsity*, which aims to find counterfactuals with minimal changes (lowest *nf* value), generating simpler explanations and contributing to the explanation's understanding and application. More specifically, *nf* (Equation 5) consists of the mean of changes (*CH_mean*) (Equation 3) and its standard deviation (*SD*) (Equation 4).

$$CH\_mean = \frac{\sum_{i=1}^{TC} CH_i}{TC} \tag{3}$$

$$SD = \sqrt{\frac{\sum_{i=1}^{TC}(CH_i - CH\_mean)^2}{TC - 1}} \tag{4}$$

$$nf = \begin{cases} CH\_mean \\ SD \end{cases} \tag{5}$$

The *c-hit* is obtained as follows: given a classifier p and an original instance x, where $p(x) = y$ and an assumed counterfactual c, the counterfactual is valid ($hit = 1$) if $p(c) = y'$ (to $y' \neq y$), and $hit = 0$, otherwise. By evaluating the counterfactuals of the entire test set, we have the percentage of valid counterfactuals that can be expressed in Equation 6.

$$c\text{-}hit = \frac{\sum_{i=1}^{TC} hit_i}{TC} \tag{6}$$

Thus, with these metrics defined, Table 5 shows the performance of the LORE and WACH methods published by Guidotti et al. (2019) and the results found in the experiments with the proposed method. Regarding *nf*, using the t-test, we conclude that, with 95% confidence, the LORE and CSSE methods present equivalent behavior on the German dataset (t = 0.451689941, *p-value* = 0.651986745 (*p-value* > 0.05), and *Critical Value* = 1.972017478). As for the COMPAS dataset, the test showed that CSSE is superior (t = 7.876682721, *p-value* = 2,17815E-13 (*p-value* < 0.05), and *Critical Value* = 1.972017478). On the other hand, the WACH method obtained worse results for the two bases evaluated. Although LORE aims to generate rule-based explanations and CSSE addresses explanations by examples, the comparison of *nf* is adequate since Guidotti et al. (2019) consider the counterfactuals generated from the rules for calculating the metrics.

Table 5
Performance comparison of LORE, WACH and CSSE counterfactual methods.

Dataset	Method	*nf*	*c-hit*
german	LORE	1.52 ± 1.18	0.78 ± 0.38
	WACH	14.80 ± 1.59	0.31 ± 0.47
	CSSE	1.46 ± 0.61	1.00 ± 0.0
compas	LORE	1.84 ± 0.78	0.87 ± 0.37
	WACH	6.24 ± 1.45	0.80 ± 0.38
	CSSE	1.16 ± 0.37	1.00 ± 0.0

Concerning the *c-hit* indicator, CSSE outperforms[8] LORE and WACH in both datasets since CSSE presents the user with only examples that belong to the counterfactual class (*c-hit* = 1.00). However, we recognize that the difference between the approaches favors CSSE over *c-hit* since LORE works with rules generated from a decision tree and naturally cannot guarantee that all rules are correct. On the other hand, in CSSE, we can use the model itself to validate the counterfactuals. Therefore, we understand that the *nf* is the most appropriate metric for comparing the methods.

Thus, the CSSE performed satisfactorily in the scenarios presented, as it obtained sparse and valid counterfactuals. The essential point for a good performance is the high capacity of the GA to explore the search space, added to the adequacy of the strategies applied in the developed solution.

5.3. Sensitivity analysis

As mentioned earlier, a counterfactual explanation aims to reverse the class of an original instance from minimal changes (minimality) in its features. We can associate minimality with similarity (minimizing the distance) or sparsity (minimum number of changes). These concepts compose the objective function proposed for GA and are presented in Equation 1, page 8.

$$C(x) = arg \min_{c} \lambda_1 \; x\_dist(c, x) + \; \lambda_2 \; x\_nchg(c, x) + \; y\_dist(p(c), y')$$

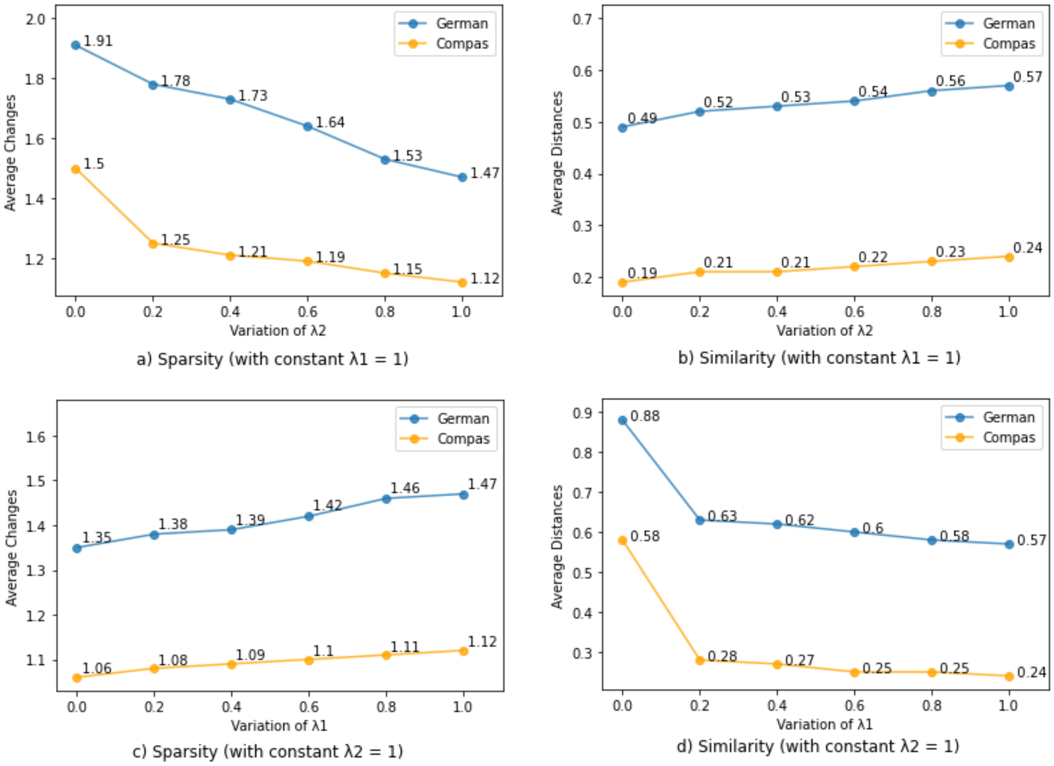
The first term of the equation, *x_dist(c, x)*, is related to *similarity*, and the second, x_nchg(c, x), to sparsity. The function also includes a third term, *y_dist(p(c), y')*, which leads the counterfactual to the target class.

The notion of minimality to be employed may vary depending on the user's needs or the nature of the application. For this reason, we include the parameters λ_1 and λ_2 in the objective function that assign weights, respectively, to similarity and sparsity. In the proposed method, the user can set float values in the interval [0, 1] to these parameters. This way, the user can express how much he prioritizes similarity or sparsity or indicate that he wants to assign the same importance.

Thus, from the models developed for the German and Compas datasets, we performed a sensitivity analysis of these parameters to demonstrate the effect of adjusting them on the counterfactuals generated by CSSE. To do this, we set the value of $\lambda_1 = 1$ and vary λ_2 in the interval [0, 1] using increments of 0.2 as a step. We apply the same strategy, setting $\lambda_2 = 1$, varying λ_1. We kept the default values for all other GA parameters. In each run, we collected the average quantity of changed features (*CH_mean*) and the average distance of the counterfactuals to the original instance to evaluate sparsity and similarity, respectively.

[8] t = -5.789473684, *p-value* = 2.7332E-08 for the German dataset, and t = -3.513513514, *p-value* = 0.00054789 for the Compas dataset.

Figures 5(a) and 5(b) show the results for sparsity and similarity, respectively, in the cases where we prioritize similarity ($\lambda_1 >= \lambda_2$). Figures 5(c) and 5(d) report these same measures when prioritizing sparsity ($\lambda_2 >= \lambda_1$).



a) Sparsity (with constant λ1 = 1)

b) Similarity (with constant λ1 = 1)

c) Sparsity (with constant λ2 = 1)

d) Similarity (with constant λ2 = 1)

Figure 5: Sparsity and similarity by varying the λ_1 and λ_2 parameters.

We achieved similar results in experiments using the German and Compas datasets. Evaluating Figures 5(a) and 5(b), we can observe that, in general terms, the more significant the difference between the parameter weights in favor of λ_1, we have counterfactuals that are more similar to the original instance, even if generated at from a larger number of changes. Thus, the best result for similarity (0.49 for German, 0.19 for Compas) and the worst for sparsity (1.91 for German, 1.50 for Compas) occur when $\lambda_1 = 1$ and $\lambda_2 = 0$.

On the other hand, from Figures 5(c) and 5(d), we notice the opposite effect when we prioritize λ_2. The more significant the difference between the weights in favor of λ_2, we have counterfactuals generated on average from a smaller quantity of changed features, even though the distance to the original instance is relatively greater. In this case, the best result for sparsity (1.35 for German, 1.06 for Compas) and the worst for similarity (0.88 for German, 0.58 for Compas) occur when $\lambda_1 = 0$ and $\lambda_2 = 1$.

The default values for executing are $\lambda_1 = 1$ and $\lambda_2 = 1$. In the experiment performed with this configuration, we obtained intermediate results for sparsity (1.47 for German, 1.12 for Compas) and similarity (0.57 for German, 0.24 for Compas), as we considered equal importance for both metrics.

We also highlight the cases in which we assign zero weight to one of the parameters. When we set $\lambda_1 = 0$, we have the worst result for similarity because the method ignores the distance to the original instance, prioritizing the number of changes exclusively. We have the same effect when $\lambda_2 = 0$, significantly impairing sparsity. Given these results, even if one wants to prioritize one of the metrics, we recommend assigning a minimum weight to the parameter accompanying the other metric. Thus, for example, given two counterfactuals generated from the same number of changes, the method will choose the more similar one.

Finally, the results found in the experiments on the different variations of λ_1 and λ_2 match the expected behavior for the method with respect to the generated counterfactuals. We noticed a less significant difference between the average distance values as we varied the parameters. We understand that it is an effect of data normalization for method execution.

6. CSSE application in the context of Attention Deficit Hyperactivity Disorder

The application of counterfactual explanations is even more consonant with problems in which one of the classes refers to something undesirable, as with health problems, for example. In (Balbino, Jandre, de Miranda, & Nobre, 2022), the authors present a work that predicts the academic performance of the arithmetic[9] of children and adolescents with Attention Deficit Hyperactivity Disorder (ADHD). This is a great context to exemplify the use and adequacy of CSSE, as we can help understand models and even indicate how to reverse cases of inadequate academic performance.

In section 6.1, we briefly describe the ADHD problem and the academic performance of these individuals. In Section 6.2, we describe the dataset and pre-processing performed. Section 6.3 brings the performance of machine learning algorithms in predicting arithmetic performance, and Section 6.4 details the application of CSSE in this context.

6.1. Contextualization

The learning of arithmetic, writing, and reading is part of the first stage of Basic Education provided in the International Standard Classification of Education (ISCED), enabling students to develop fundamental skills in these three disciplines, as well as the formation of a solid base to learn and understand the areas main of knowledge and personal and social development (UNESCO, 2012). Furthermore, it is understood that literate and knowledgeable individuals in mathematical operations become more independent and better appreciate their tasks since simple daily actions, such as elaborating a shopping list or checking the value of change, become more difficult or even unrealizable without the basic notion of these areas.

Thus, the analysis of this problem is essential because about 40% of people with ADHD have difficulties in academic activities, especially in learning and application of knowledge in these three disciplines. In addition, frustration in the educational scenario, either interaction or performance, can lead to depression and anxiety, which become associated problems in more than half of ADHD cases (Cortez & Pinheiro, 2018; Loe & Feldman, 2007; Mattos, 2015; Moreira & Barreto, 2017).

These school adversities become more impactful when the individual is in childhood and adolescence, as parents and teachers usually do not know how to act and what measures should be taken in the face of their difficulties, causing damage that generally reflects on different aspects of their lives (Mattos, 2015). It is known that the earlier the diagnosis of ADHD and the implementation of therapeutic measures, the lower the negative impact the disorder can achieve on people with it and their surroundings (Moreira & Barreto, 2017; Muzetti & de Luca Vinhas, 2017).

Thus, knowing the pattern that reduces or expands the academic difficulties of children and adolescents with ADHD enables the creation of strategies that soften dissatisfaction and trauma that may arise in the lives of these individuals.

6.2. Dataset description and preprocessing

This research partnered with the Department of Pediatrics and the Federal University of Minas Gerais Faculty of Medicine, which provided the database of 266 children and adolescents aged between 6 and 18 years, being 196 with ADHD and 70 without the disorder.

All of these patients were diagnosed by the Impulsivity and Attention Research Center (NITIDA - Núcleo de Investigação da Impulsividade e Atenção) at the University Hospital of Clinics of this University, where children with suspected ADHD, by indication from schools or health centers, are carefully diagnosed by a multidisciplinary team involving the areas of child psychiatry, pediatrics, neuropsychology, among other professionals. In the case of an ADHD diagnosis, parents/guardians are informed, and the most appropriate treatment is initiated according to the needs of each patient.

Each patient has 225 features, including personal, family, gestational health, medical, socioeconomic, parental care, education, and scores for each individual's arithmetic, writing, and reading tests. It is noteworthy that we consider this work only the performance in arithmetic.

Regarding the pre-processing performed in the dataset, we adopt the same steps adopted by Balbino et al. (2022): 1) transformation of the classification feature, which was numerical, to nominal (lower and higher grade); 2) exclusion of duplicate features (for example, age in months and years) or irrelevant to predict school performance; for example,

[9] In (Balbino et al., 2022), the authors exploit other disciplines, as well as arithmetic.

name and phone number); 3) binarization of non-ordinal nominal features (One-Hot Encoding[10]); for example, the father's marital status and the student's disorders; 4) imputation of missing data; 5) class balancing, as the class with superior performance is the minority (imbalance in the ratio of 1:2.7); 6) A genetic algorithm was applied for feature selection. Of the 225 features, the AG selected 29 (including the class), obtaining better results than the complete dataset. A detailed description of these steps can be found in (Balbino et al., 2022). In Table 12, in supplementary materials, we present a complete list of all features considered, along with their category and domain.

About the division of the base for training/validation and testing, we randomly separated 15% of the instances of each class to perform the test step. This division resulted in a balanced set of 118 cases for training/validation and 40 examples for testing (with 11 instances of the Higher class and 29 of the Lower class). We use the test set also as an instance set to explain the prediction using CSSE.

6.3. Performance of algorithms in the prediction of academic performance

Seeking to obtain a better predictive capacity, we developed models based on four ML algorithms: Decision Tree, Neural Networks, SVM, and Random Forest. The models were implemented in Python using the Scikit-learn library. We present the parameters applied to the prediction models in Table 6. The other parameters not present in the table follow the default values of each model.

Table 6
Control parameters applied to learning algorithms.

Algorithm	Control parameters	Values
Decision Tree	max_depth	8
	criterion	entropy
Random Forest	n_estimators	120
	criterion	entropy
	max_features	log2
Neural Networks	hidden_layers_sizes	10
	learning_rate_init	0.05
	activation	identity
SVM	kernel	linear
	probability	true
	degree	1

We used the metrics of Precision, Recall, and F-measure to evaluate the quality of the models. All classifiers were built and validated using the k-fold cross-validation process, with $k = 10$. The proposal is to apply CSSE to the model with better predictive performance, as the model's quality strongly impacts the explanation's quality.

Figure 6 presents the test results of the predictive models for arithmetic. Table 13, in supplementary materials, shows some statistical information about the models, including confidence interval (C.I.), with a 95% confidence level, regardless of the classification performance, in which the sample values come from 10-fold cross-validation. From this, we can conclude that, for the Random Forest algorithm, with 95% confidence, the calculated confidence intervals contain the population means of precision, recall, and F-measure for each class. For example, for the F-measure metric, each class's confidence interval, mean, and Standard Deviation (SD) are *Lower*: C.I.: (0.66, 0.89), *Higher*: C.I: (0.67, 0.88), Mean: 0.78, and SD: 0.15. For more information, see Table 13.

Furthermore, we applied a t-Student hypothesis test, adopting the F-Measure metric since it represents the harmonic mean between Precision and Recall, to obtain the mean error difference between the four investigated models. Thus, our comparison used the class averages, h_0: $\overline{x_1} = \overline{x_2}$, with a 95% confidence level. The significance test results showed a difference between the evaluated models and that Random Forest is better, with $p\text{-}value = 0.0212$ ($p\text{-}value < 0.05$). All the other models are equivalent.

[10] Despite recognizing the problems of one-hot coding (Hancock & Khoshgoftaar, 2020) (Pargent, Pfisterer, Thomas, & Bischl, 2022), we use it due to the low cardinality of these features. We recommend investigating more efficient encodings to increase the quality of learning models.

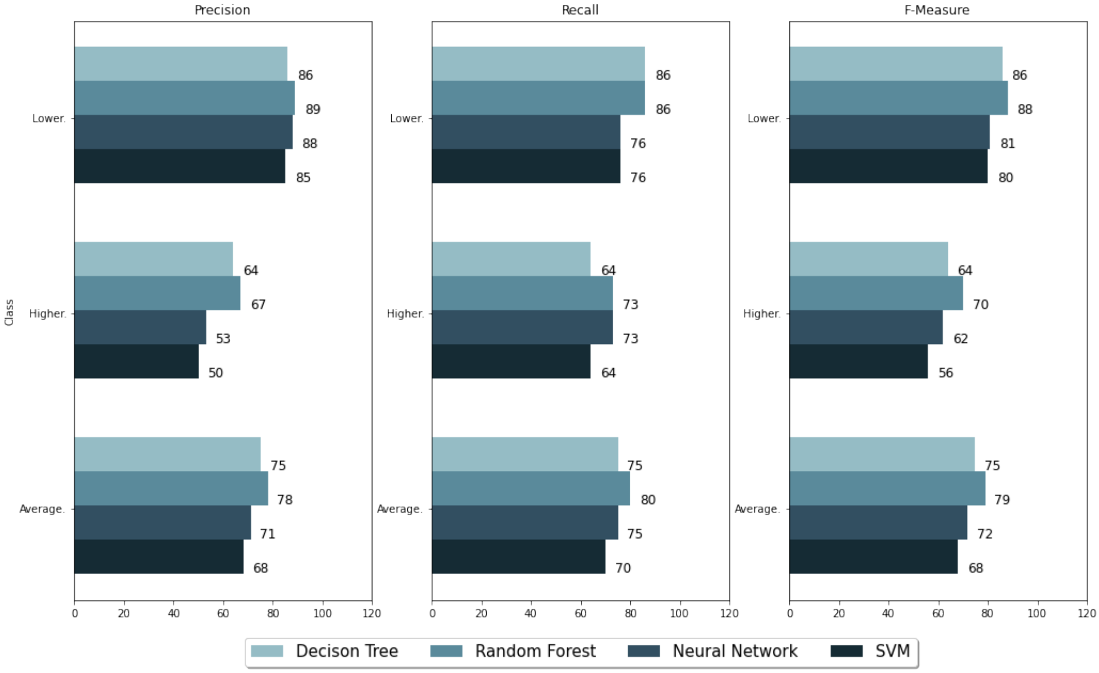


Figure 6: Models performance evaluation in the context of ADHD.

These results show that the Random Forest has an F-measure value of 88% and 70% for the 'lower' and 'higher' classes, respectively. This better behavior in the lower class also appears in the precision and recall metrics for Random Forest and all other algorithms. Investments in different balancing algorithms and other encodings may improve these models' false positive and negative rates, thus improving the precision and recall metrics, respectively.

6.4. Application of CSSE in the context of ADHD

From the academic performance prediction model in arithmetic with Random Forest, we apply CSSE to generate explanations for the predictions of the 40 instances selected as a test/explanation set. CSSE sought three counterfactual explanations for every 40 instances considering the default parameters. We do not use static_list so that the method can use all features in the explanation.

Table 7 contains the list of features mentioned in the description of the experiments and the numerical transformation of their values. Tables 8 and 9 exemplify counterfactual explanations generated for two instances representing individuals with ADHD and classified as having lower performance in arithmetic. As it is a local explanation method, we consider that the user will direct the application of the method to previously defined instances of interest.

In the first case (Table 8), we present three explanations of how to reverse the class from low to high performance: 1) if the mother's schooling were incomplete graduation instead of high school; or 2) if the mother's inattention score were 3.0 instead of 12.0; or 3) if the student's social class were high instead of vulnerable.

In the second case (Table 9), the instance of the lower class would reverse its prediction in the following situations: 1) if the test score $Raven\_Z$[11] had been 0.550 instead of -0.3725; or 2) if the test score $Raven\_Z$ had been 0.120 instead of -0.3725 and if the student studied in a private school; or 3) if the test score $Raven\_Z$ had been 0.120 instead of -0.3725 and if the student were of high social class.

Specifically, in the cases presented, the presence in the explanations of the variables mother's education, mother's inattention, type of school, and social class demonstrate the need to support individuals with ADHD to improve their academic performance. These results corroborate the findings of Balbino et al. (2022); Jandre, Balbino, de Miranda, Zárate, and Nobre (2023); Jandre et al. (2021), highlighting the influence of social factors and the importance of maternal monitoring in the academic performance of students with ADHD. Additionally, Raven_Z, a variable related to the student's IQ, also appears prominently in the work of Balbino et al. (2022).

[11] The Raven_Z refers to the average standard deviation of the value obtained in the Raven's Progressive Matrices Test.

Table 7
Features highlighted in the explanations.

Feature	Domain
Raven_Z	-3.58...2.41
School_type	Public school (0) or Private school(1)
Social_class	Poor(0), Vulnerable(1), Middle class(2) or High class(3)
Mother_inattention	1.0...31.0
Mother_schooling	1 to 4 years(0), 5 to 8 years(1), Incomplete HS(2), Complete HS(3), Incomplete graduation(4), or Complete graduation(5)

Table 8
CSSE application result - Instance 1 (Arithmetic).

	Modified features		
	Mother_Schooling	**Mother_inattention**	**Social_class**
Original instance: **class**: Lower	3 (Complete HS)	12.0	1 (Vulnerable)
Counterfactuals **class**: Higher	4 (Incomplete graduation)	_	_
	_	3.0	_
	_	_	3 (High class)
No static feature			

Table 9
CSSE application result - Instance 2 (Arithmetic)

	Modified features		
	Raven_Z	**School_type**	**Social_class**
Original instance: **class**: Lower	-0.3725	0 (Public)	0 (Poor)
Counterfactuals **class**: Higher	0.550	_	_
	0.120	1 (Private)	_
	0.120	_	3 (High class)
No static feature			

We emphasize the treatment of diversity and prolixity performed by the method, which generated each explanation using a different set of features. In particular, in the second case (Table 9), we highlight a relevant quality of this treatment: notice that the three explanations used the *Raven_Z*. However, the method only keeps the second and third explanations because these perform a minor modification on the *Raven_Z* than the first. Thus, if the user wants to act based on the explanations, he has two options: modify only one feature with a more significant variation in value or two features with the *Raven_Z* closer to the original value.

Thus, besides providing an opportunity for experimentation for CSSE, this study can provide subsidies for parents, educators, and other professionals (psychologists, psychiatrists, and neurologists) to direct their actions in the search for better results for students with the disorder.

7. Discussion and Conclusion

This work presents the CSSE method for the explainability of black box models. Our motivation for this is that the literature reveals the need for improvements in current interpretability methods in communicating with end users, especially for users who are not specialists in machine learning.

In this sense, Miller (2019) indicates that the solution is to expand interpretability methods beyond computational issues and a better understanding of how people receive explanations. The author suggests that we also consider the studies carried out in the social sciences, an area with extensive and valuable research on the subject. The author lists three aspects of an end-user-oriented approach: contrastive, selected, and social explanations.

Given this, CSSE is a method based on GA to generate counterfactual explanations in line with the guidelines pointed out by Miller (2019). Therefore, we identify and implement resources that aim to meet these guidelines. CSSE presented satisfactory results in the experiments, outperforming other methods highlighted in the literature. Furthermore, in the case study developed in the context of the academic performance of children and adolescents with ADHD, the application of the method was valuable and promising.

Main contributions. The main novelty of the work is developing a principle-based method that includes computational and social science aspects to generate explanations aimed at the end user, including non-ML specialists. The work of (Miller, 2019) is the primary reference for developing our approach. As the essential characteristics and resources of the method, we can highlight the counterfactual explanations, the generation of multiple explanations, with diversity and without prolixity, the possibility for the user to choose the notion of minimality that he wants to adopt, and to allow the user to indicate the list of immutable features. In addition, some adjustments of a more technical nature (for example, GA parameters) contribute to adapting the method to each application. Another advantage of CSSE is that it is agnostic and applicable to any classification model. The qualitative and quantitative comparisons show our proposal's advances concerning relevant methods in the literature.

Experiments performed. We performed experiments comparing CSSE with LORE (Guidotti et al., 2019) and WACH (Wachter et al., 2017) using the German and Compas datasets. In the German data set, using the student's t-test, the CSSE is shown as equivalent to LORE and superior to WACH. In the Compas dataset, the CSSE outperformed the other methods (with *p-value* = 2.17815E-13). Thus, we highlight the robustness and adequacy of the GA in the search for counterfactual solutions. Finally, we also performed experiments to analyze the sensitivity of GA parameters.

Case study. Our work also involved a case study on predicting academic performance in children and adolescents with ADHD. We investigated four learning algorithms (Decision Tree, Random Forest, Neural Network, and SVM) to identify these individuals' superior and inferior academic performance. We performed the student's t-test based on a difference between sample means obtained from the cross-validation folds of the four investigated learning algorithms. The Random Forest proved to be the most efficient for solving this problem from this test. Using the Random Forest, we have 70% and 88% of F-measure for the higher and lower classes, respectively. These values follow the confidence intervals calculated with 95% confidence (see Table 13 in Supplementary materials). This study also includes the analysis of the interpretability, using CSSE, of the model obtained by Random Forest. The CSSE indicated the following characteristics as relevant for predicting the academic performance of individuals: Raven_Z, type of school, social class, and mother's features. Thus, the results constitute a support tool for family members, health professionals, and other people close to the children and adolescents searching for better academic performance for students with the disorder.

Therefore, the proposed CSSE method presents advances in interpretability compared to current practices, as it generates user-oriented explanations that increase user confidence and understanding of model decisions.

7.1. Limitations and future work

In the current version of CSSE, we limit its application to classification models with binary output. However, the method can be extended to multiclass models by allowing the user to choose the counterfactual target class. Another present limitation is that our approach only works with tabular data. We expect to support text or images as input in future work.

In counterfactuals generation, when one feature is changed, another one could be required to be updated. For example, suppose that the interest rate is related to the number of installments in a loan evaluation system. In this case, if the number of installments is changed when generating a counterfactual, the interest rate should be changed accordingly. In the context of XAI, this treatment is commonly called *causality*. However, few methods observe causality relationships (Guidotti, 2022). In a future version of CSSE, we intend to treat causal relationships between input features.

One of the significant advantages of GA is the ability to deal well with optimization problems with many variables. Thus, we expect that CSSE will perform even better when compared to other methods on datasets with a more significant number of input features or scenarios that require a larger quantity of changes to the original instance to find the counterfactual. Therefore, we plan to perform other experiments to confirm this hypothesis in future work. Furthermore, extending the tests with the method to different scenarios is essential to consolidate the results found and make the necessary adjustments.

Finally, we also plan to conduct studies with developers, data scientists, and non-expert ML users to understand how they evaluate the available resources and format of the output generated by CSSE. It is important to note how multiple explanations, diversity, prolixity, similarity, and sparsity affect users' satisfaction with explanations.

8. Acknowledgements

The authors thank the National Council for Scientific and Technological Development of Brazil (CNPq), the Coordination for the Improvement of Higher Education Personnel - Brasil (CAPES) - Finance Code 001, the Foundation for Research Support of Minas Gerais State (FAPEMIG), the Federal Center for Technological Education of Minas Gerais (CEFET-MG), and the Pontifical Catholic University of Minas Gerais (PUC Minas).

9. Supplementary materials

Tables 10 and 11 show the characteristics of the Compas and German databases, respectively.

Table 10
Features of the Compas dataset.

Feature	Domain
Age	mean = 34.82, std = 11.89, min = 18, max = 96, p25 = 25, p50 = 31, p75 = 42
Age category	Less than 25(0), 25 - 45(1), Greater than 45(2)
Gender	Female (0), Male(1)
Race	African-American(0), Asian(1), Caucasian(2), Hispanic(3), Native American(4), Other(5)
Priors count	mean = 3.47, std = 4.88, min = 0, max = 38, p25 = 0, p50 = 2, p75 = 5
Days between screening and arrest	mean = 17.38, std = 72.19, min = 0, max = 1057, p25 = 1, p50 = 1, p75 = 1
Charge degree	F (0), M(1)
Is recid	0,1
Is violent recid	0,1
Two year recid	0,1
Length of stay	mean = 49.62, std = 81.23, min = 0, max = 806, p25 = 1, p50 = 16, p75 = 60
Risk of Recidivism (class)	Medium-Low, High

Table 12 presents the features of the ADHD database, organized into 15 categories: 1) Personal information; 2) School Information; 3) General health; 4) Sleep information; 5) K-SADS interview; 6) Mother information; 7) Father information; 8) Family disorders; 9) Adult Self-Report Scale (ASRS); 10) State-Trait Anxiety Inventory (STAI); 11) Beck Depression Inventory (BDI-II); 12) Gestational; 13) Nativity; e 14) Tests to measure IQ and school performance (Raven Test).

Table 13 shows some statistical measures of the investigated learning algorithms, such as the models' performance metrics separated by class, the general mean with the standard deviation, and the confidence interval obtained from the 10-fold cross-validation.

Table 11
Features of the German dataset.

Feature	Domain
Age	mean = 35.55, std = 11.38, min = 19, max = 75, p25 = 27, p50 = 33, p75 = 42
Personal status/gender	male: divorced/separated(1), female: divorced/separated/married(2), male: single(3), male: married/widowed(4), female: single(5)
Present residence since	mean = 2.85, std = 1.10, min = 1, max = 4, p25 = 1, p50 = 2, p75 = 3
People under maintenance	mean = 1.16, std = 0.36, min = 1, max = 2, p25 = 1, p50 = 1, p75 = 1
Installment as income percentage	mean = 2.97, std = 1.12, min = 1, max = 4, p25 = 2, p50 = 3, p75 = 4
Other debtors / guarantors	none(1), co-applicant(2), guarantor(3)
Savings account/bonds	... < 100 DM(1), 100 <= ... < 500 DM(2), 500 <= ... < 1000 DM(3), .. >= 1000 DM(4), unknown/no savings account(5)
Property	real estate(1), if not (1): building society savings agreement/life insurance(2), if not (1)/(2): car or other, not in feature "Savings account"(3), unknown/no property(4)
Housing	rent(1), own(2), for free(3)
Telephone	none(1), yes, registered under the customers name (2)
Job	unemployed/unskilled - non-resident(1), unskilled - resident(2), skilled employee/official(3), management/self-employed(4), highly qualified employee/officer(5)
Present employment since	unemployed(1), ... < 1 year(2), 1 <= ... < 4 years(3), 4 <= ... < 7 years(4), .. >= 7 years(5)
Foreign worker	yes(1), no(2)
Account check status	... < 0 DM(1), 0 <= ... < 200 DM(2), ... >= 200 DM(3), salary assignments for at least 1 year(4), no checking account(5)
Credit history	no credits taken/all credits paid back duly(0), all credits at this bank paid back duly(1), existing credits paid back duly till now(2), delay in paying off in the past(3), critical account/ other credits existing (not at this bank)(4)
Number of existing credits at this bank	mean = 1.41, std = 0.58, min = 1, max = 4, p25 = 1, p50 = 1, p75 = 2
Duration in month	mean = 20.90, std = 12.06, min = 4, max = 72, p25 = 12, p50 = 18, p75 = 24
Credit amount	mean = 3271.26, std = 2822.74, min = 250.00, max = 18424.00, p25 = 1365.50, p50 = 2319.50, p75 = 3972.25
Purpose	car (new)(0), car (used)(1), furniture/equipment(2), radio/television(3), domestic appliances(4), repairs(5), education(6), (vacation - does not exist?)(7), retraining(8), business(9), others(10)
Other installment plans	bank(1), stores(2), none(3)
Credit risk rating (class)	good(0), bad(1).

Table 12
Features of ADHD dataset, organized by category.

Category	Feature	Domain
Personal information	Age	mean = 9.83, std = 2.19, min = 6, max = 18, p25 = 8.00, p50 = 9.50, p75 = 11.00
	Catholic (if the patient is catholic)	Yes(1) or No(0)
	Height	mean = 1.43, std = 0.15, min = 1.04, max = 1.72, p25 = 1.33, p50 = 1.41, p75 = 1.52
	Weight	mean = 39.42, std = 13.82, min = 19.00, max = 87.00, p25 = 31.25, p50 = 36.58, p75 = 41.31
School Information	School_type	Public school(0) or Private school(1)
	School_Year	Early childhood education(0), First year(1), Second year(2), Third year(3), Fourth year(4), Fifth year(5), Sixth year(6), Seventh year(7), Eighth year(8), Ninth year(9) or High school(10)
General health	Accident	Yes(1) or No(0)
	Food allergy	Yes(1) or No(0)
	Neurologist	Yes(1) or No(0)
Sleep information	sleep hours	mean = 9.13, std = 1.11, min = 5.00, max = 13.00, p25 = 9.00, p50 = 9.00, p75 = 9.50
K-SADS interview	ADHD (if the patient has ADHD)	Yes(1) or No(0)
Mother information	Mother age	mean = 38.07, std = 6.37, min = 25.00, max = 67.00, p25 = 35.00, p50 = 38.00, p75 = 42.00
	Mother schooling	1 to 4 years(0), 5 to 8 years(1), Incomplete HS(2), Complete HS(3), Incomplete graduation(4), or Complete graduation(5)
Father information	Father age	mean = 41.28, std = 6.16, min = 26.00, max = 58.00, p25 = 38.00, p50 = 41.00, p75 = 44.00
	Married father	Yes(1) or No(0)
	Father schooling	1 to 4 years(0), 5 to 8 years(1), Incomplete HS(2), Complete HS(3), Incomplete graduation(4), or Complete graduation(5)
Family disorders	If the patient's mother has some disorder	Yes(1) or No(0)
ASRS[a]	Father inattention	mean = 9.11, std = 1.23, min = 6.00, max = 20.00, p25 = 9.00, p50 = 9.00, p75 = 9.00
	Mother functional impairment (if the manifestation of the symptoms present in the questionnaire cause disturbances that negatively influence the patient's mother's life)	Yes(1) or No(0)
	Mother hyperactivity	mean = 11.04, std = 5.07, min = 0.00, max = 31.00, p25 = 10.00, p50 = 11.00, p75 = 11.75
	Mother inattention	mean = 12.27, std = 4.97, min = 1.00, max = 31.00, p25 = 11.25, p50 = 12.00, p75 = 13.00
STAI[b]	Mother state (patient's mother's score in the analysis of the degree of anxiety in a transient reaction directly related to a situation of adversity that appears at a given moment)	Low anxiety(0), Medium anxiety(1) or High anxiety(2)
BDI-II[c]	BDI mother (total detection score of depressive symptoms in patient's mother)	Mild depression(0), Moderate depression (1) or Severe depression(2)
	Social_class	Poor(0), Vulnerable(1), Middle class(2) or High class(3)
Gestational	Contraceptive (If the patient's mother became pregnant, even making use of contraceptives)	Yes(1) or No(0)
Nativity	Birth	Normal(0) or Caesarean(1)
	Birth weight	mean = 3.11, std = 0.55, min = 1.10, max = 4.26, p25 = 2.90, p50 = 3.20, p75 = 3.35
Raven Test	Raven_Z (mean of the standard deviation of the value obtained in the Raven Progressive Matrix Test)	mean = -0.07, std = 1.31, min = -3.58, max = 2.41, p25 = -0.99, p50 = -0.11, p75 = 1.01
Class	Performance	Lower, Higher

[a] ASRS - Adult Self-Report Scale. [b] STAI - State-Trait Anxiety Inventory. [c] BDI-II - Beck Depression Inventory

Table 13
Description of the statistical measures, confidence interval, mean and standard deviation, and evaluation metrics of the proposed models

Algorithm	Recall			Precision			F-measure		
	Lower	Higher	Mean/SD(±)	Lower	Higher	Mean/SD(±)	Lower	Higher	Mean/SD(±)
Decison Tree	(0.55, 0.82)	(0.44, 0.86)	0,67/0.24	(0.50, 0.86)	(0.55, 0.77)	0,67/0.20	(0.52, 0.82)	(0.48, 0.79)	0,65/0.21
Random Forest	(0.66, 0.85)	(0.66, 0.97)	0.79/0.18	(0.67, 0.98)	(0.67, 0.88)	0.80/0.18	(0.66, 0.89)	(0.67, 0.88)	0.78/0.15
Neural Network	(0.47, 0.81)	(0.52, 0.83)	0.66/0.22	(0.49, 0.86)	(0.51, 0.86)	0.68/0.24	(0.50, 0.75)	(0.53, 0.75)	0.63/0.16
SVM	(0.44, 0.76)	(0.65, 0.88)	0.68/0.20	(0.54, 0.90)	(0.53, 0.82)	0.70/0.22	(0.49, 0.79)	(0.58, 0.83)	0.67/0.18

SD - Standard Deviation

References

Abdul, A., Vermeulen, J., Wang, D., Lim, B. Y., & Kankanhalli, M. (2018). Trends and trajectories for explainable, accountable and intelligible systems: An hci research agenda. In *Proceedings of the 2018 chi conference on human factors in computing systems* (pp. 1–18).

Adadi, A., & Berrada, M. (2018). Peeking inside the black-box: A survey on explainable artificial intelligence (xai). *IEEE Access*, *6*, 52138–52160.

Balbino, M., Jandre, C., de Miranda, D., & Nobre, C. (2022, June). Predictions of academic performance of children and adolescents with ADHD using the SHAP approach. *Stud Health Technol Inform*, *290*, 655–659.

Carvalho, D. V., Pereira, E. M., & Cardoso, J. S. (2019). Machine learning interpretability: A survey on methods and metrics. *Electronics*, *8*(8), 832.

Chen, Z.-Y., Lin, W.-C., Ke, S.-W., & Tsai, C.-F. (2015). Evolutionary feature and instance selection for traffic sign recognition. *Computers in Industry*, *74*, 201–211.

Cortez, M. T., & Pinheiro, Â. M. V. (2018). TDAH e escola: incompatibilidade? *Paidéia*, *13*(19).

Derrac, J., García, S., & Herrera, F. (2012). A survey on evolutionary instance selection and generation. In *Modeling, analysis, and applications in metaheuristic computing: Advancements and trends* (pp. 233–266). IGI Global. Retrieved from https://doi.org/10.4018/jamc.2010102604

Dhurandhar, A., Chen, P.-Y., Luss, R., Tu, C.-C., Ting, P., Shanmugam, K., & Das, P. (2018). Explanations based on the missing: Towards contrastive explanations with pertinent negatives. *Advances in neural information processing systems*, *31*.

Du, M., Liu, N., & Hu, X. (2019). Techniques for interpretable machine learning. *Communications of the ACM*, *63*(1), 68–77.

El Shawi, R., Sherif, Y., Al-Mallah, M., & Sakr, S. (2019). Interpretability in healthcare a comparative study of local machine learning interpretability techniques. In *2019 ieee 32nd international symposium on computer-based medical systems (cbms)* (p. 275-280). IEEE. Retrieved from https://ieeexplore.ieee.org/document/8787506

Ghorbani, H., Wood, D. A., Moghadasi, J., Choubineh, A., Abdizadeh, P., & Mohamadian, N. (2019). Predicting liquid flow-rate performance through wellhead chokes with genetic and solver optimizers: an oil field case study. *Journal of Petroleum Exploration and Production Technology*, *9*, 1355–1373.

Gomez, O., Holter, S., Yuan, J., & Bertini, E. (2020). Vice: Visual counterfactual explanations for machine learning models. In *Proceedings of the 25th international conference on intelligent user interfaces* (pp. 531–535).

Guidotti, R. (2022). Counterfactual explanations and how to find them: literature review and benchmarking. *Data Mining and Knowledge Discovery*, 1–55.

Guidotti, R., Monreale, A., Giannotti, F., Pedreschi, D., Ruggieri, S., & Turini, F. (2019). Factual and counterfactual explanations for black box decision making. *IEEE Intelligent Systems*, *34*(6), 14–23. doi: 10.1109/MIS.2019.2957223

Hamdia, K. M., Zhuang, X., & Rabczuk, T. (2021). An efficient optimization approach for designing machine learning models based on genetic algorithm. *Neural Computing and Applications*, *33*, 1923–1933.

Hancock, J. T., & Khoshgoftaar, T. M. (2020, Apr 10). Survey on categorical data for neural networks. *Journal of Big Data*, *7*(1), 28. Retrieved from https://doi.org/10.1186/s40537-020-00305-w doi: 10.1186/s40537-020-00305-w

Jandre, C., Balbino, M., de Miranda, D., Zárate, L., & Nobre, C. (2023). Towards interpretable machine learning models to aid the academic performance of children and adolescents with attention-deficit/hyperactivity disorder. In *Biomedical engineering systems and technologies: 14th international joint conference, biostec 2021, virtual event, february 11–13, 2021, revised selected papers* (pp. 180–201).

Jandre, C., Santos, B. C., Balbino, M., de Miranda, D., Zárate, L. E., & Nobre, C. (2021). Analysis of school performance of children and adolescents with attention-deficit/hyperactivity disorder: A dimensionality reduction approach. In *Healthinf* (pp. 155–165). Retrieved from https://www.scitepress.org/Papers/2021/102404/102404.pdf

Karim, A., Mishra, A., Newton, M. H., & Sattar, A. (2018). Machine learning interpretability: A science rather than a tool. Retrieved from https://arxiv.org/abs/1807.06722 doi: 10.48550/ARXIV.1807.06722

Keane, M. T., & Smyth, B. (2020). Good counterfactuals and where to find them: A case-based technique for generating counterfactuals for explainable ai (xai). In *Case-based reasoning research and development: 28th international conference, iccbr 2020, salamanca, spain, june 8–12, 2020, proceedings 28* (pp. 163–178).

Kim, K.-j. (2006). Artificial neural networks with evolutionary instance selection for financial forecasting. *Expert Systems with Applications*, *30*(3), 519–526.

Loe, I. M., & Feldman, H. M. (2007). Academic and educational outcomes of children with adhd. *Journal of pediatric psychology*, *32*(6), 643–654.

Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. In *Advances in neural information processing systems* (pp. 4765–4774).

Mattos, P. (2015). *No mundo da lua: perguntas e respostas sobre transtorno do défict de atenção com hiperatividade em crianças, adolescentes e adultos*. E-BOOK: Associação Brasileira do Défict de Atenção.

Miller, T. (2019). Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, *267*, 1–38.

Mittelstadt, B., Russell, C., & Wachter, S. (2019). Explaining explanations in ai. In *Proceedings of the conference on fairness, accountability, and transparency* (p. 279–288). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3287560.3287574

Mokhtari, K. E., Higdon, B. P., & Başar, A. (2019). Interpreting financial time series with SHAP values. In *Proceedings of the 29th annual international conference on computer science and software engineering* (pp. 166–172).

Molnar, C. (2020). *Interpretable machine learning*. Lulu. com.

Moreira, S. C., & Barreto, M. A. M. (2017). Transtorno de déficit de atenção e hiperatividade: conhecendo para intervir. *Revista Práxis*, *1*(2).

Mothilal, R. K., Sharma, A., & Tan, C. (2020). Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the 2020 conference on fairness, accountability, and transparency* (pp. 607–617).

Muzetti, C. M. G., & de Luca Vinhas, M. C. Z. (2017). Influência do déficit de atenção e hiperatividade na aprendizagem em escolares. *Psicologia argumento*, *29*(65).

Pargent, F., Pfisterer, F., Thomas, J., & Bischl, B. (2022, Nov 01). Regularized target encoding outperforms traditional methods in supervised machine learning with high cardinality features. *Computational Statistics*, *37*(5), 2671-2692. Retrieved from https://doi.org/10.1007/s00180-022-01207-6 doi: 10.1007/s00180-022-01207-6

Rathi, S. (2019). Generating counterfactual and contrastive explanations using SHAP. *ArXiv*, *abs/1906.09293*.

Ribeiro, M. T., Singh, S., & 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* (pp. 1135–1144).

Shahab, M., Azizi, F., Sanjoyo, B., Irawan, M., Hidayat, N., & Rukmi, A. (2021). A genetic algorithm for solving large scale global optimization problems. In *Journal of physics: Conference series* (Vol. 1821, p. 012055).

Stepin, I., Alonso, J. M., Catala, A., & Pereira-Fariña, M. (2021). A survey of contrastive and counterfactual explanation generation methods for explainable artificial intelligence. *IEEE Access*, *9*, 11974–12001.

Tjoa, E., & Guan, C. (2020). A survey on explainable artificial intelligence (xai): Toward medical xai. *IEEE Transactions on Neural Networks and Learning Systems*, 1-21. doi: 10.1109/TNNLS.2020.3027314

Tsai, C.-F., Eberle, W., & Chu, C.-Y. (2013). Genetic algorithms in feature and instance selection. *Knowledge-Based Systems*, *39*, 240–247.

UNESCO, I. f. S. (2012). *International standard classification of education: Isced 2011*. Quebec: UNESCO Institute for Statistics Montreal.

Van Looveren, A., & Klaise, J. (2021). Interpretable counterfactual explanations guided by prototypes. In *Machine learning and knowledge discovery in databases. research track: European conference, ecml pkdd 2021, bilbao, spain, september 13–17, 2021, proceedings, part ii 21* (pp. 650–665).

Verma, S., Boonsanong, V., Hoang, M., Hines, K. E., Dickerson, J. P., & Shah, C. (2020). Counterfactual explanations and algorithmic recourses for machine learning: A review. *arXiv preprint arXiv:2010.10596*.

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

Xue, Y., Zhu, H., Liang, J., & Słowik, A. (2021). Adaptive crossover operator based multi-objective binary genetic algorithm for feature selection in classification. *Knowledge-Based Systems*, *227*, 107218.

Zeebaree, D. Q., Haron, H., Abdulazeez, A. M., & Zeebaree, S. (2017). Combination of k-means clustering with genetic algorithm: A review. *International Journal of Applied Engineering Research*, *12*(24), 14238–14245.