H.F.R.I call "Basic Research Funding (Horizontal Support for all Sciences)"
National Recovery and Resilience Plan
(Greece 2.0)

Project Name: Counterfactuals for Clustering: Explainability, Fairness and Quality

Acronym: FairXCluster

Project No: 15940

Deliverable D1.1: State-of-the-art on Counterfactual Explanations

The research project is implemented in the framework of H.F.R.I call "Basic Research Funding (Horizontal Support for all Sciences)" under the National Recovery and Resilience Plan "Greece2.0", funded by the European Union – NextGenerationEU (H.F.R.I project Number: 15940)

State-of-the-art on Counterfactual Explanations

1 Machine Learning: Advancements and Model Complexity

Over the past few decades machine learning (ML) methods have emerged as a powerful techniques revolutionizing the way we analyze and interpret massive amounts of data. By facilitating the development of algorithms capable of recognizing intricate patterns in complex datasets, ML allows us to extract meaningful insights, make accurate predictions, and discover data patterns that otherwise could remain hidden [1, 2, 3, 4]. Recent ML problems are becoming increasingly challenging and require more advance and complex models to solve them [5]. Unfortunately, such ML models often sacrifice explainability due to several reasons such as:

- **Model complexity:** As ML models become more complex they incorporate numerous layers, parameters, and interactions. This complexity increases their predictive power, but makes it difficult to understand how they arrive at certain decisions.
- **Non-Linear Transformations:** Complex models use non-linear transformations, such as deep neural networks. These transformations allow them to capture intricate patterns in the data. However, the trade-off is that the interpretation of these non-linear functions becomes convoluted.
- **Feature Engineering:** Complex models automatically learn features from raw data, bypassing manual feature engineering. While this is beneficial, it obscures the direct relationship between input features and model predictions.
- **Black box nature:** Some advanced models function as "black boxes". They make accurate predictions but lack transparency. Understanding the inner workings of these models becomes elusive, hindering explainability.
- **Trade-offs:** Model complexity often involves tradeoffs. While simpler models (e.g., linear regression) are interpretable, they may sacrifice predictive performance. Complex models strike a balance, but at the expense of interpretability.

2 Explainability in ML

2.1 The demand of explainable AI

It is evident that in several ML applications explainability in decision making is a requirement. To establish that statement let's consider the following scenario [6]: Alice applies for a loan at a bank. A machine learning classifier is used and the decision to analyzes Alice's characteristics such as income, credit score, education, and age. Despite her application, Alice is denied the loan, leaving her with two key questions:

1. Why was the loan denied?
2. What steps can she take to secure approval in the future?

The first question could be answered with explanations such as "CreditScore is too low" and is similar to most traditional explanations. The latter question forms the basis of Counterfactual explanations: what (possibly small) changes could be made to Alice's feature vector in order to end up on the other side of the classifier's decision boundary?

2.2 Types of explainability methods

In general, explainability methods in ML can be categorized into three main types. These are the self-explainable vs post-hoc explainable, global vs local explanations, and model-dependent vs model-agnostic explanation methods.

Self-explanatory models are generally simple models, such as decision trees and linear models, which by definition are inherently transparent. Let's elaborate on Alice example, and suppose that the ML system responsible for the loans relies on the following linear model $f(x) = 0.7 \times \text{credit} + 0.95 \times \text{salary} + 0.1 \times \text{education} + 0.2 \times \text{age}$. It is evident that this model heavily prioritizes the features of credit score and the salary. On the other hand, post-hoc explanations, are generally applicable to complex models after their training phase, such as neural networks, due to their black-box nature.

Global explanations provide a general understanding of the behavior of the model. Such methods are useful for identifying patterns or biases in the model. In contrast, local explanations are responsible for explaining individual predictions, providing a more precise insight into why a particular instance x received a particular output $\hat{y}$ from the model f_θ ($f_\theta(x) = \hat{y}$). Prominent local explanation techniques are Local Interpretable Model-agnostic Explanations (LIME) [7] and SHapley Addictive exPlanations (SHAP) [8].

The final major distinction of explainability methods is that between model-dependent and model-agnostic explanations. The former category, as their name implies, are capable of explaining the output of a particular model or family of models. Such methods utilize the internal properties and structure of the model to provide insights into its decision-making process. Naturally, the explanations provided by such methods tend to be more faithful and accurate. On the other

hand model-agnostic methods are not tied to any specific model and can be universally applied.

3 Counterfactual Explanation: Definition and Properties

3.1 Definition

Counterfactual explanations (CFEs) constitute a relatively new type of explanation methods [6]. In general, CFEs are applied to complicated models that are typically black-box in nature. Specifically, CFEs are categorized as post hoc and local explanability methods, and were first introduced in 2017 in [9]. This means that CFEs assume a trained model $f_{\theta^\star}(x) = y$, where $\theta^\star$ are the trained parameters, x is the input vector, and y is the model output to be applied. To simplify the notation, we symbolize the trained model as $f(x)$.

Before we dive into the mathematical formulation, let's use Alice's example to better understand the CFE idea. Suppose there are two possible model outcomes: $y =$ `'negative'`, the applicant **does not qualify** for the loan, and $y' =$ `'positive'`, the applicant **does qualify** for the loan. Alice did not get the loan she applied for, since $f(x) = y$, meaning that the input vector x corresponding to Alice didn't produce the desired output y'. An explanation for such a decision is needed to help Alice get the loan in the future: what is the minimum required change that Alice should make (in terms of income, education, etc.) in order to qualify for the loan.

More generally, this is the kind of explanation CFEs give to a model decision: *What is the minimum change that can be applied to x (producing x') so that the model output changes from $f(x) = y$ to $f(x') = y'$?* The definition of CFE is the following:

Definition 1. *Given a classifier f that outputs the decision $y = f(x)$ for an instance x, a counterfactual explanation consists of an instance x' such that the decision for f on x' is different from y, i.e., $f(x') \neq y$, and such that the difference between x and x' is minimal [10].*

In term of mathematical formulation, this question naturally formulates an optimization objective:

$$\arg\min_{x'} d(x, x') \tag{1}$$

$$\text{s.t. } f(x') = y' \tag{2}$$

where d is a function that measures the distance between data points $x, x' \in \mathcal{X}$, and $\mathcal{X}$ is the input domain. Such an objective is not generally easy to optimize due to the nonlinear nature of the constraints. However, using the penalty approach, this objective function can be transformed into a differentiable unconstrained optimization problem as follows [9]:

$$\arg\min_{x'} \max_{\lambda} \{\lambda(f(x') - y')^2 + d(x, x')\} \tag{3}$$

3.2 CFEs desirable properties

Note that eq. 3 formulates the minimum requirements optimization problem for a datapoint x' to be *validly* characterized as a CFE. At the same time, it provides a solid foundation for further development. Of course, there are other requirements, such as considering only mutable features, such as income, and not immutable ones, such as race. Thus, the formulation can be further generalized to take into account a set $\mathcal{A}$ of *actionable* features as follows:

$$\arg\min_{x' \in \mathcal{A}} \{\max_{\lambda} \lambda(f(x') - y')^2 + d(x, x')\} \tag{4}$$

Another requirement is that a CFE should modify the fewest possible features most effectively. Thus the objective can be enriched with **sparsity** constraints as follows:

$$\arg\min_{x' \in \mathcal{A}} \max_{\lambda} \{\lambda(f(x') - y')^2 + d(x, x') + g(x' - x)\}, \tag{5}$$

where $g(\cdot)$ can be for example the L_0 or L_1 norma to enforce sparsity.

Another notable requirement is that the CFEs should result in a combination of features that are realistic, in a sense that they are already observed in the training data. For such a requirement, the CFEs should be computed to be relatively *close to the data manifold*. In particular, an appropriate penalty loss $l(x'; \mathcal{X})$ can be introduced, which results in the next CFE formulation:

$$\arg\min_{x' \in \mathcal{A}} \max_{\lambda} \{\lambda(f(x') - y')^2 + d(x, x') + g(x' - x) + l(x'; \mathcal{X})\} \tag{6}$$

It should be noted that features in a dataset are rarely independent; thus, altering one feature probably impacts others. Therefore, *causality* is another property, which is associated with actionability and plausibility, as a counterfactual generated through causation ensures these two properties of an action by preserving the causal connection between elements.

Another desirable property is *diversity*. In this case, instead of a single CFE for instance x, we wish to produce a set $C = \{x'_1, ..., x'_k\}$ of counterfactuals. The counterfactual explanation set C should be formed by diverse counterfactuals, i.e., in addition to counterfactual $x'_i \in C$ being minimal and similar to x, their difference should also be maximized. Diversity suggests different ways of changing the outcome class. We can encode diversity by forcing that the pairwise distance between counterfactual explanations is greater than a given value [11]. Another approach for satisfying diversity, proposed in [12] is by building on determinant point processes (DPP). Of course, there are additional requirements that may be application or domain specific, see [6] for more details on this topic.

An additional categorization of CFE methods is *endogenous* vs *exogenous* explorers [10]. Endogenous methods propose CFEs that exist in the dataset X, thus such methods can guarantee *plausibility*. On the other hand, exogenous methods generate (synthetic) CFEs that do not necessarily belong to the dataset X. This sometimes may be be problematic because the suggested changes may be difficult or even impossible to make. Referring back to the Alice's example,

Table 1: Taxonomy of CFEs methods.

Name	Model	Strategy	Data Type	Code
WACH [9]	Gradients	Optimization	Tabular	✓
NICE [14]	Agnostic	Instance Based	Tabular	✓
CEM [15]	Gradients	Optimization	✓	✓
CEML [16]	Agnostic	Optimization	✓	✓
TREPAN [17]	Agnostic	Decision Trees	Tabular	✗
FACE [18]	Agnostic	Instance Based	✓	✗
GRACON [19]	Gradients	Heuristic Search	✓	✗
MUCH [20]	SVDD	Sampling	Tabular	✓
GRACE [21]	Gradients	Optimization	Tabular	✓
GSG [22]	Agnostic	Heuristic Search	✓	✓
DICE [12]	Gradients	Optimization	Tabular	✓
FT [23]	Tree-base Ensembles	Decision Trees	Tabular	✓
FOCUS [24]	Tree-base Ensembles	Optimization	Tabular	✓
CEGP [25]	Agnostic	Optimization	✓	✓
ARES [26]	Agnostic	Optimization	Tabular	✗
DACE [27]	Linear/Tree-base Ensembles	Optimization	Tabular	✓
CEODT [28]	Tree	Optimization	✓	✗
CEML [29]	Agnostic	Optimization	Tabular	✓
CET [30]	Agnostic	Optimization	Tabular	✓
FACTS [31]	Agnostic	Frequent Itemset	Tabular	✓
GLOBE-CE [32]	Agnostic	Optimization	Tabular	✓

it is impossible to for Alice to change her age or gender in order to get her loan accepted [13]. However, it should be noted that the majority of methods belong to the exogenous category [10].

4 CFE Methods for Individual Instances

In this section, we present a variety of CFE methods that can generally be classified as either model agnostic or model based. It should be noted that the vast majority of methods focuses on the classification problem.

Table 1 presents a taxonomy of CFE methods. The *Name* column refers to the name of the method. The *Model* column refers to the type of access the method requires from the model to compute CFEs. The *Strategy* column indicates the procedure in which the CFE is computed. The *Data Type* column indicates the type of data to which the method can be applied.

4.1 Model Agnostic Methods

Model agnostic methods refer to CFE techniques that are generally applicable to different algorithms and are not restricted to a specific model. This flexibility

allows them to be used across different machine learning models, making them highly versatile. As a result, these methods provide consistent and comparable explanations regardless of the underlying model used in the analysis.

NICE. Nearest Instance Counterfactual Explanations (NICE) is a model agnostic method that treats the model $f(x)$ as a black-box [14]. The NICE method can only be applied to tabular data and can also handle categorical features. First, it finds the nearest unlike neighbor x'' of datapoint x, where $f(x'') \neq f(x)$. Note that by definition x'' is itself a counterfactual. Additionally, the method identifies the non-overlapping features of x'' and x. This procedure identifies a set of features that can be gradually changed to make x resemble the counterfactual x''. If the change results in a positive reward, it is kept, otherwise it is not. Finally, the examined feature is removed from the feature list.

FACE. It should be noted that the majority of methods proposed in the literature generate CFEs that are not necessarily representative of the underlying data distribution. Thus the generated CFEs may even suggest unachievable goals to the user. In [18] the concept of the "feasible path" between the current state x and the target state x' is introduced. Feasible and Actionable Counterfactuals Explanations (FACE) is a method that computes actionable CFEs through the discovery of feasible path.

A feasible path is modeled as the shortest path defined on a data density-weighted matrix. FACE consists of three main options of computing the density weights of the matrix as follows:

$$w_{ij} = f_{\hat{p}}(\frac{x_i + x_j}{2})||x_i - x_j|| \quad \text{(KDE)} \tag{7}$$

$$w_{ij} = \tilde{f}(\frac{r}{||x_i - x_j||})||x_i - x_j||, \quad \text{where} \quad r = \frac{k}{N n_d} \quad \text{(k-NN)} \tag{8}$$

$$w_{ij} = \tilde{f}(\frac{\epsilon^d}{||x_i - x_j||})||x_i - x_j||, \quad \text{when} \quad ||x_i - x_j|| \leq \epsilon \quad (\epsilon\text{-graph}) \tag{9}$$

In the formulas above, $f(\cdot)$ is a positive scalar function, n_d is the volume of a sphere of unit radius in $\mathbb{R}^d$. Of course, if the conditions are not satisfied, then $w_{ij} = 0$. After the matrix is computed, the shortest path algorithm (Dijkstra's algorithm) [33] is run over all candidate targets to find all data points that satisfy the conditions.

TREPAN. Decision trees (DTs) [34, 35] are well known in ML for their exceptional interpretability due to their internal transparency. DTs use axis parallel hyperplanes as decision boundaries, while their inherent structure provides a transparent framework for understanding decisions. Thus, the idea of using them for interpretability is actually an old one. TREPAN [17] serves as

an introductory post-hoc explanation method that uses a DT to provide an explanation for a complex classifier such as a neural network. The core idea of TREPAN is to use the DT to approximate the inference of a (trained) complex model in order to reveal its internal logic of decision making on a local or global scale. In addition, the method uses the structure of the DT to enforce feature constraints. It should be noted that TREPAN is not a proposed method for computing CFEs, however with minimal adjustments it can provide CFEs as noted in [10]. It is important to note that decision trees have their own set of limitations, including susceptibility to the curse of dimensionality and suitability primarily for tabular datasets.

GSG. The Growing Spheres Generation (GSG) [22] is another model and data agnostic method. Its strategy relies on a generative approach that grows a sphere around the data point x to find the CFE x'. Specifically, GSG method heuristically optimizes the following cost function:

$$x' = \arg\min ||x - x'||_2 + \gamma||x - x'||_0 \tag{10}$$

$$\text{s.t. } f(x') = y' \tag{11}$$

where $||\cdot||_0$ the l_0 norm defined as the number of non-zero elements and γ a hyperparameter weighting the two terms.

The heuristic optimization approach is designed in two steps, the generation step and the feature selection step. The generation step constructs samples uniformly at random in a sphere. If the generated samples do not contain CFEs, the sphere grows. If the initial sphere contains CFEs, the sphere shrinks to find the CFEs with the smallest distance. In the feature selection step, a different heuristic strategy is followed to create sparse CFEs. Specifically, the method ignores the small feature changes as the less locally relevant with respect to the classifier decision boundary. Thus, the algorithm aims to adjust as many feature exclusions as possible, as long as the predicted class of x' does not change.

4.2 Model-based Methods

Model-based methods are developed to explain the results of a particular model or family of models. More specifically, model-based methods use the internal properties and structure of the model to provide insight into its decision-making process. Explanations derived from model-based techniques tend to be more accurate and faithful because they exploit the internal workings of the model.

WACH. As discussed earlier, in [9] a well-defined formulation for CFE computation has been introduced. In particular, the optimization objective presented in eq. 3, consists of two terms, $\lambda(f(x') - y')^2$ and $d(x, x')$. Except for the typical L2 norm, the first term aims to compute the CFE x'. By definition, x' should belong to the category y'; thus the term $\lambda(f(x') - y')^2$ formulates a natural objective. At the same time, the second term $d(x, x')$ is targeted towards discovering a counterfactual x' that is close to the datapoint x. The value λ is a

regularization parameter responsible for balancing the the contribution of the first term against the second term.

As noted in [9] the choice of the distance function d is a crucial characteristic for the computation of the CFE. The paper suggests the use of L_1 norm, or Manhattan distance, weighted by the inverse Median Absolute Deviation (MAD). The median absolute deviation for the feature k, over the set of datapoint X is defined as:

$$MAD_k = \text{median}_{j \in X}(x_{j,k} - \text{median}_{l \in X}(x_{l,k})). \tag{12}$$

Finally, the proposed distance function d is formulated as:

$$d(x_i, x') = \sum_{k \in F} \frac{|x_{i,k} - x'_k|}{MAD_k} \tag{13}$$

Of course, the authors suggest that the distance function d should be refined based on the application. Finally, the CFE can be computed by the optimization of the objective function presented in Eq. 3.

CEM. The Contrastive Explanation Method (CEM) requires access to the gradients of the model f [15]. Therefore, the model must be differentiable. The authors define the CFE as $x' = x + \delta$, where δ is a perturbation applied to x s.t. $f(x') \neq f(x)$. In addition, the *Pertinent Negative* and *Pertinent Positive* analyses are defined. The main goal of such a formulation is to generate explanations that not only discover the minimum change required to be applied to x, but also identify contrastive features that should be minimally absent for x to maintain its current class. In particular, the methods aim to answer the following question: "An input x is classified in class y because features $f_i, \ldots, f_k$ are present *and* because $f_m, \ldots, f_p$ are absent". The CEM is based on two optimization objectives, whose minimization finds δ^{pos} and δ^{neg} corresponding to the positive and negative explanations, respectively. The CEM can be applied to neural networks and to data from different domains (tabular, images, etc.).

GRACON. In [19] the GRAdual CONstruction CFEs (GRACON) method is proposed for deep neural networks without softmax activation. GRACON models the CFE x' as follows:

$$x' = (1 - M) \circ x + M \circ C \tag{14}$$

where $M = \{0, 1\}^d$ is a binary mask, C is a composition matrix and $\circ$ denotes the element-wise multiplication. GRACON consists of two main steps, the masking step and the composition step. The goal of the masking step is to select an important feature to change the original decision from y to the target category y'. Then the composition step optimizes C for the selected feature to improve the output score of the target class.

Specifically, the masking step selects the most influential feature as follows:

$$i^\star = \max(|\nabla f_{y'}(x)|)_i, \tag{15}$$

where $f_{y'}$ is the output of the model corresponding to class y'. After the masking step, the composition step optimizes the feature value to ensure that the deep network assigns the perturbed data x' (eq. 14) to class y'. The following objective function is proposed for the specification of C:

$$\arg\min_{C} \left| \sum_{k=1}^{K} \left(f'_k(x') - \frac{1}{N} \sum_{i=1}^{N} f'_k(x_{i,y'}) \right) \right| + \lambda |x' - x| \tag{16}$$

where K is the number of classes, f'_k represents the logit score for a class k and $X_{i,y'}$ denotes the i_{th} datapoint that is classified to class y'. The GRACON method iterates between the masking step and the composition step and returns the CFEs x'.

MUCH. In [20] the MUltiCounterfactual via Halton sampling (MUCH) method is proposed for determining multiple counterfactual explanations based on Support Vector Data Description [36] in a mutli-class framework. The core idea of MUCH is to utilize Halton sampling to generate multiple CFEs.

TGT. In [37] a new type of explanation called a Global Counterfactual Explanation (GCE) is introduced along with an algorithm called Transitive Global Translations (TGT) for computing GCEs. GCEs aim to identify the most important differences between groups of points in a low-dimensional representation of data.

The authors assume $r : \mathbb{R}^d \to \mathbb{R}^m$ a differentiable function that maps the points in the feature space to a lower-dimensional representation space. The authors also assume two regions of interest in the feature space X_{initial} and X_{target} and also in the representation space R_{initial} and R_{target}. The goal of GCE is to compute a transformation that takes the points in X_{initial} and maps them to R_{target} using r. Hence the goal is to find a transformation $t : \mathbb{R}^d \to \mathbb{R}^d$ such that:

$$r(t(x)) \in R_{\text{target}} \forall x \in X_{\text{initial}} \tag{17}$$

where $t(x) = x + \delta$ is the explanation. In order to find δ, the TGT algorithm minimizes the following objective function:

$$loss(\delta) = ||r(\overline{x}_{\text{initial}} + \delta) - \overline{r}_{\text{target}}||_2^2 + \lambda ||\delta||_1, \tag{18}$$

where $\overline{x}_{\text{initial}}$ and $\overline{r}_{\text{target}}$ are mean values in feature space and representation space, respectively.

GRACE. In [21] a CFE method is proposed for explaining neural networks that are trained on tabular datasets. Given the neural network model $f(x)$,

GRACE solves the following optimization problem:

$$\begin{aligned} &\min_{x'} \ dist(x', x) \\ &\text{s.t. } \arg\max(f(x)) \neq \arg\max(f(x')) \\ &\quad |S| \leq K \\ &\quad SU(X^i, X^j) \leq \gamma \quad \forall i, j \in S \\ &\quad x' \in dom(X) \end{aligned} \tag{19}$$

where S is the feature set of x that are perturbed to generate x', K is the allowed number of feature to change, $SU(\cdot)$ is Symmetrical Uncertainty function (a normalized form of mutual information), γ is an upper bound (hyperparameter), and *dom* is the set of actionable features. The GRACE algorithm first initializes $x' = x$. Then it ranks all features according to their predictive power with respect to the prediction $f(x)$, resulting in the ordered list $\mathcal{U}$. This ordering can be based on gradients with respect to the nearest contrasting class v. Then it computes the new ordered list $\mathcal{U}^\star$ from $\mathcal{U}$, by iteratively adding each feature (from the most to least predictive) such that the SU value between any pair of features in $\mathcal{U}^\star$ is within the upper bound γ. Finally, GRACE generates the CFE sample by perturbing x' towards the contrasting class (using the gradients) until it changes category.

DICE. In [12] a framework is proposed for generating and evaluating a diverse set of actionable counterfactuals which should also satisfy two properties: *feasibility* and *diversity*. Specifically, they extend the work of Wachter [9] and construct a loss function with the following formulation:

$$C(x) = \arg\min_{c_1,\ldots,c_k} \frac{1}{k}\sum_{i=1}^{k} \text{yloss}(f(c_i), y) + \lambda_1 \frac{1}{k}\sum_{i=1}^{k} \text{dist}(c_i, x) - \lambda_2 \text{dpp\_diversity}(c_1, \ldots, c_k) \tag{20}$$

where c_i represents a CFE, k is the total number of CFEs to be generated, $f(.)$ denotes the ML model, $yloss(.)$ is a metric that quantifies the error between $f(.)$'s prediction for c_i and the target outcome y, d represents the total number of input features, x is the original instance, $dpp\_diversity(\cdot)$ is the diversity metric and λ_1 and λ_2 are hyperparameters that adjust the balance among the three components of the loss function. The first part ($yloss$) pushes the counterfactual c towards a different prediction than the original instance x, the second term enforces the proximity property. i.e. CFE should be close to the original input in order to be more useful to the user and the third term captures diversity by building on determinantal point processes (DPP). The method handles categorical features by utilizing one-hot encoding and introduces a regularization component with a substantial penalty for each categorical feature. In addition, this work defines measures for validity, proximity and diversity which are used to evaluate the set of CFEs generated from any method. Also, a secondary model is used (1-NN classifier), using both the generated CFE set and the original

input to assess the performance of CFEs. This evaluation involves examining how accurately the secondary model can replicate the predictions of any new input of the original machine learning model.

CEGP. Looveren et al [25] propose an approach for finding interpretable counterfactual explanations of classifier predictions by using *class prototypes*. Let x_0 be the original instance and $x_{cf} = x_0 + \delta$ be the counterfactual instance. CEGP considers the following objective function to be minimized with respect to δ:

$$L = c \cdot L_{\text{pred}} + \beta \cdot L_1 + L_2 + L_{\text{AE}} + L_{\text{proto}} \tag{21}$$

where L_{pred} encourages the predicted class i of the perturbed instance x_{cf} to be different than the predicted class of the original instance x_0 and c is a scaling parameter. The loss term $\beta \cdot L_1 + L_2$ represents the distance between the x_0 and the x_{cf} and enforces the property of sparsity. CEGP includes the loss term L_{AE} to generate plausible counterfactual instances by using an autoencoder (AE) which is fit on the training data. CEGP also adopts the loss term L_{proto} based on *prototypes* in order to guide the perturbations δ towards an interpretable counterfactual x_{cf}. For each class, CEGP, establishes a prototype through the encoder part of the autoencoder. This prototype is defined as the average encoding of the k nearest instances with the same class label in the latent space. When given an input x, CEGP initially identifies the nearest prototype in the latent space and efficiently solves the optimization problem. If the training encoder does not exist, they build a k-d tree to represent each class in order to find the nearest prototype. Another contribution of this paper is the approach for handling categorical features, achieved by using pairwise distance measures to create embeddings of categorical features within a numerical space.

CEML. In Artelt et al [29], it is explored how *prototype-based* models, like *Learning Vector Quantization* (LVQ), can be used to compute counterfactual explanations by exploiting its specific structure. Specifically, in order to compute a counterfactual x' of a given input x assigned to class y, they solve the following optimization problem for each prototype p_i of class $y' \neq y$ and select the counterfactual x' that minimizes a loss (distance) function $\theta(x', x)$:

$$\arg\min_{x' \in R^d} \theta(x', x) \tag{22}$$

$$s.t \quad d(x', p_i) + \epsilon \leq d(x', p_j) \quad \forall p_j \in P(y') \tag{23}$$

where p_i is the i-th prototype, $P(y')$ denotes the set of all prototypes not labeled as y' and $\epsilon > 0$ is a small value preventing the counterfactual from lying exactly on the decision boundary. Additionally, they demonstrate how to integrate plausibility constraints into their framework to guarantee plausible counterfactual explanations via an optimization problem structured as follows:

$$\arg\min_{x' \in R^d} \theta(x', x) \tag{24}$$

$$s.t. \quad h(x') = y' \quad p(y') \geq \delta \tag{25}$$

where h is the prediction function of the LVQ model, $p_y(.)$ denotes a class dependent density and δ denotes a minimum density value for which they assume a data point to be plausible. Also, they propose a *counterfactual metric* for explaining the change of distance matrix based models when faced with new data. Instead of altering a specific data point to achieve a desired prediction, they propose making minimal adjustments to the distance matrix of the model to obtain the desired prediction for the given data point. Let $h : \mathbb{R}^d \rightarrow Y$ be a prediction function that depends on a distance matrix $\Omega \in S_+^d$. They define a counterfactual metric $\Omega' \in S_+^d$ as the solution to the following optimization problem:

$$\arg \min_{\Omega' \in S_+^d} \theta(\Omega, \Omega') \tag{26}$$

$$s.t. \quad h_{\Omega'}(x) = y' \tag{27}$$

where (x, y') constitutes (or a set of) labeled and currently misclassified sample (samples), S_+^d denotes the set of $d \times d$ symmetric positive semi-definite matrices and $\theta(\Omega', \Omega)$ quantifies the difference between the counterfactual distance matrix Ω' and the original distance matrix Ω. This objective defines the minimum required change of a given classifier such that a new training sample, is classified differently as compared to the current status. In addition, they investigate how to solve efficiently this optimization problem for various types LVQ models, like the generalized matrix learning vector quantization (GMLVQ) and the localized generalized matrix learning vector quantization (LGMLVQ).

4.3 CFEs for Tree-based Models

Considerable research work has focused on the computation of CFEs for tree-based classifiers (decision trees and random forests). Some of the proposed methods are summarized below.

FT. Tolomei et al. [23] introduces a method based on actionable Feature Tweaking (FT) aimed at determining which adjustable features of a given instance x should be transformed in order to influence the prediction of a tree-based ensemble. The FT method can only be applied on tabular data and it can also handle both continuous and categorical features. FT focus on $\hat{f}$ represented as an ensemble of K tree-based classifiers, $\hat{f} = \phi(h_1, ..., h_k)$. Each $h_k : X \rightarrow Y$ is a base estimate, and ϕ is the function responsible for combining the outputs of all the individual base classifiers into a single prediction. In any tree-based ensemble classifier, each h_k is encoded by a decision tree T_k, and the ensemble is represented as a forest $T = \{T_1, ..., T_k\}$. FT tweaks the original input vector x in order to turn the predicted output of the ensemble from negative (-1) to positive (+1). FT skips the set of trees with positive output and focuses on each tree T_k with negative output. It considers the set of all positive paths of

each negative tree T_k. Within each negative tree, the algorithm identifies paths leading to positive outcomes. For each such path, it associates an instance from the vector space that satisfies the boolean conditions along that path, reaching a positive outcome. Among these instances it selects those having slightly adjusted feature values, allowing for slight changes within a specified tolerance of at most ϵ. For any small fixed $\epsilon > 0$, they build a feature vector $x'(\epsilon)$ as follows:

$$x'_i(\epsilon) = \begin{cases} \theta_i - \epsilon & \text{if } x_i \leq \theta_i \\ \theta_i + \epsilon & \text{if } x_i > \theta_i \end{cases} \tag{28}$$

where θ_i is the threshold on the i-th feature value. Nevertheless, it should be taken into account that each such transformation may have an impact on other trees of the forest. In other words, by changing x into another instance x', it is only guaranteed that the prediction of the base classifier T_k is correctly fixed, i.e. from $h_k(x) = -1$ to $h_k(x') = 1$.

CEODT. In [28], the authors focus on counterfactual explanations using classification trees, both axis-aligned trained with *CART* and oblique trees trained with the Tree Alternating Optimization (TAO) algorithm [38, 39]. Oblique decision trees recursively divide the feature space by using splits based on linear combinations of features. In contrast to their univariate equivalents which utilize only one feature for each split, oblique decision trees are frequently more compact and accurate. Given an input instance x that has been classified by the tree as belonging to class y ($T(x) = y$), the original optimization problem is to find the nearest instance x' that is classified as another class $y' \neq y$ (the target class). Solving this problem involves minimizing the following objective function subject to several constraints:

$$\min_x E(x'; x) \text{ s.t. } T(x) = y, c(x) = 0, d(x) \geq 0 \tag{29}$$

where $E(x'; x)$ is the cost of changing features of x, and $c(x)$ and $d(x)$ are equality and inequality constraints in vector form. Although the tree function $T(x)$ is non-convex and non-differentiable, the authors provide a strategy for simplifying the original optimization problem. The main idea is decomposing the problem into smaller sub-problems corresponding to individual leaves of a tree, making the discovery of counterfactuals easier. Therefore, the focus shifts to solving the optimization problem *within a single leaf* $i \in L$ that satisfies the desired label condition $y_i = y$. The objective is to minimize the function $E(x'; x)$ subject to the constraints specific to the region R_i of leaf i. The optimization problem over a single leaf $i \in L$ is represented as follows:

$$\min_{i \in L} \min_{x \in R_i} E(x'; x) \text{ s.t. } h_i(x) \geq 0, c(x) = 0, d(x) \geq 0 \tag{30}$$

where $h_i(x)$ represents constraints specific to leaf i, $c(x)$ represents equality constraints, and $d(x)$ represents inequality constraints.

FOCUS. Flexible Optimizable Counterfactual Explanations (FOCUS) is a model-specific method for tree-ensembles [24]. The FOCUS method can only be applied on tabular data and it can handle both continuous and categorical features. Building on Wachter et al [9], FOCUS faces the challenge of identifying counterfactual explanations by suggesting integrating differentiable approximations of non-differentiable models into the gradient-based optimization framework. More specifically, for acquiring the differentiable approximation $\tilde{f}$ of f, FOCUS constructs a probabilistic approximation of the original tree ensemble f and replaces each split in each tree with a sigmoid function.

Robx. In Sanghamitra et al. [40], a method called RobX is presented for generating counterfactuals for tree-based ensembles that are not only valid but also robust. A new metric is proposed termed *counterfactual stability* to measure how robust a counterfactual is going to be. Let $M(\cdot) : \mathbb{R}^d \to [0, 1]$ be the tree-based ensemble model that takes an input vector and outputs the positive class probability. Counterfactual stability of a counterfactual x' is defined as follows:

$$R_{K,\sigma^2}(x', M) = \frac{1}{K} \sum_{x'' \in N_{x'}} M(x'') - \sqrt{\frac{1}{K} \sum_{x'' \in N_{x'}} \left(M(x'') - \frac{1}{K} \sum_{x'' \in N_{x'}} M(x'') \right)^2} \tag{31}$$

where $N_{x'}$ is a set of K points in $\mathbb{R}^d$ drawn from the distribution $\mathcal{N}(x', \sigma^2 I_d)$ with I_d being the identity matrix.

Given a data point $x \in X$ such as $M(x) \leq 0.5$, the goal is to find a valid counterfactual x' with $M(x') > 0.5$ that belongs to the data manifold of x, but is also robust. The first step of the method is to generate a counterfactual x' for an instance x using any existing method for tree-based ensembles, such as Feature Tweaking (FT) or FOCUS. The second step is to check if the generated counterfactual satisfies the counterfactual *stability test*. The counterfactual *stability test* is satisfied when $R_{K,\sigma^2}(x', M) \geq \tau$ where τ is a threshold. If the test criterion is not met, the algorithm produces c conservative counterfactuals, which are the c nearest neighbors of x' in the dataset that pass the stability test. Then, it iteratively approaches each of them until a stable counterfactual is identified for all c cases. At the end, it picks the stable counterfactual with the lowest L_p distance from x, for $p = 1$ or $p = 2$.

DACE. The Distribution-Aware Counterfactual Explanation (DACE) method [27] is based on mixed-integer linear optimization. The contribution of this approach is a new cost function that builds on Mahalanobis distance and on the Local Outlier Factor (LOF) in order to enforce the plausibility of the generated counterfactuals. For two vectors x, $x' \in R^d$ and a positive semi-definite matrix $M \in \mathbb{R}^{d \times d}$, Mahalanobis distance between x and x' is defined as

$$d_M(x, x'|M) := \sqrt{(x' - x)^T M (x' - x)} \tag{32}$$

LOF is a outlier score that measures how unusual a given instance is by using k-nearest neighbors (k-NN) computation. DACE focus on additive classifiers, such as Linear models and Tree ensemble models and the aim is to find the perturbation vector a that minimizes the cost $C_{DACE}(x|a)$. Given a positive semi definite matrix M, a set X of N instances, a positive integer k, and $\lambda \geq 0$, DACE defines the objective function C_{DACE} with respect to an input instance x as:

$$C_{DACE}(a|x) = d_M^2(x, x+a|M) + \lambda \cdot q_k(x+a|X) \tag{33}$$

where $d_M^2(x, x+a)$ is the squared MD between the input instance x and its modified instance $x+a$, $q_k(x+a|X)$ is the k-LOF of $x+a$, and $\lambda \geq 0$ is a trade-off parameter between d_M^2 and q_k. Moreover, DACE manages categorical features using one-hot encoding and mitigates implausibility through the LOF score.

5 CFE Methods for Groups of Instances

Except for computing CFEs given a specific instance, it may also be desirable to compute CFEs given a set (group) of instances. These are called *group counterfactuals*.

In [41] the most general problem is addressed of finding a group of counterfactual explanations for a group of instances. The authors present several mathematical optimization models to illustrate each potential allocation rule between counterfactuals and instances. First, the authors define the single-instance single-counterfactual case that has the following bi-objective optimization formulation:

$$\min_x \left(C(x_0, x), -P(x)\right) \tag{34}$$

where $C(x_0, x)$ is the cost to perturb x_0 to x and $P(x)$ is the probability x of being classified as positive that must be high.

Then they define the *group counterfactual* problem in which given a group of instances $\mathbf{x_0}'$ they seek to find a group of R counterfactual instances $\mathbf{x}' = x_1, ..., x_R$

$$\min_{\mathbf{x}} \left(\mathcal{C}(\mathbf{x_0'}, \mathbf{x}), -\mathcal{P}(\mathbf{x})\right), \tag{35}$$

The third optimization problem is an alternative approach of Eq. 34 where they add a hard constraint, i.e. the P must be above a threshold value $\nu \in [0, 1]$ and has the following form:

$$\min_{\mathbf{x}} \quad \mathcal{C}(\mathbf{x}_0, \mathbf{x}) \tag{36}$$

$$\text{s.t.} \quad \mathcal{P}(\mathbf{x}) \geq \nu. \tag{37}$$

Furthermore, they outline the various components necessary to formulate mathematical optimization problems in counterfactual analysis. The first component is the *ambient space*, which is the domain from which counterfactuals are drawn.

(a) One-for-one (b) Many-for-one (c) One-for-all (d) One-for-many

Figure 1: Allocation rules between instances (squares) and their counterfactual explanations (circles) in group counterfactual analysis.

They divide counterfactuals into endogenous and exogenous. Endogenous counterfactuals have the advantage of being real and are retrieved by solving combinatorial problems, such as the p-median problem. In contrast, exogenous counterfactuals, which is the most popular approach, are constructed by solving continuous or mixed-integer optimization problems, such as the minimum-sum-of-squared-distances problem. The second component is the allocation rules, which dictates how counterfactual explanations are assigned to instances. The allocation rules are defined as follows:

- One-for-one: Exactly one counterfactual for each instance.
- Many-for-one: Many counterfactuals for a exactly one instance. This allocation rule provides some type of diversity.
- One-for-all: All the instances share the same counterfactual.
- One-for many: After partitioning the instances into subsets, one counterfactual is computed for each subset.

Figure 1 graphically presents the above allocation rules. Next, they examine the various constraints required for counterfactual explanations. These constraints ensure the explanations are practical, plausible, and useful. The first major constraint is related to the interactions between instances and counterfactuals such as restrictions on the number of changes in order to prevent unrealistic modifications, typically using a distance measure $d(x_0, x_r) \leq \tau$. The second constraint is to fulfill the closeness to historical data i.e., counterfactuals should remain near historical data points or within the convex hull of the original dataset. In addition, counterfactual explanations should presernve fixed features that must remain unchanged One more constraint is the maintenance of statistical distributions, ensuring diversity and similarity within clusters. Other essential components needed to define the mathematical optimization problems include the utilization of score-based classifiers and the manner in which the probabilities of different counterfactuals to belong to the positive class are aggregated. Another critical aspect is the cost criterion, which differs for endogenous and exogenous counterfactuals. This criterion evaluates the difficulty of perturbing instances to generate their counterfactuals. All optimization problems have been solved using a general optimization package, namely the Gurobi optimization solver.

AReS. In Rawal et al [26], a model-agnostic method termed Actionable Recourse Summaries (AReS) is proposed to generate global counterfactuals. These explanations aim to offer an interpretable and accurate summary of recourses for the entire population with special attention given to specific subgroups of interest. For example, these subgroups can be characterized by features, such as race or gender. The goal of this framework is to capture the differences in recourse among various subgroups. AReS proposes a two-level recourse set, denoted as R, which is a structured model organized hierarchically. It comprises several recourse sets, each of which is enclosed within an outer if-then framework. The outer if-then rules can be seen as descriptors for subgroups, representing various subpopulations within the data. The inner if-then rules are recourses for the corresponding subgroups. A two-level recourse set is a set of triples of following form:

$$R = (q_1, c_{11}, c'_{11}), (q_1, c_{12}, c'_{12})(q_2, c_{21}, c'_{21}) \tag{38}$$

where q_i corresponds to the subgroup descriptor and (c_{ij}, c'_{ij}) together represent the inner if-then recourse rules with c_{ij} denoting the *if-condition* and c_{ij} denoting the *recourse*. A two-level recourse set offers recourse to an instance x as follows: if x meets only one rule i (i.e., x meets $q_i \wedge c_i$), its recourse is c'_i. If x doesn't meet any rule in R, no recourse is provided. If x meets multiple rules in R, the recourse is determined by the rule with the highest probability of providing a correct recourse, computed directly from the data. In addition, AReS quantifies key aspects of the explanations in order to construct correct recourses and simultaneously provide recourses for many affected individuals, while reducing costs and ensuring interpretability. These are formalized as follows:

- Recourse Correctness: the number of instances in X_{aff} for which acting upon the prescribed recourse by R does not lead to the desired prediction. X_{aff} is the set of affected individuals who received unfavorable outcomes.
- Recourse Coverage: measures how many individuals in X_{aff} are provided with recourses.
- Recourse Costs: Each of the M features is associated with a cost that reflects the difficulty of changing its value, which means that some features are more *actionable* than the others. Thus, this includes $\textit{featurecost}(R) = \sum_{i=1}^{M} \text{cost}(c_i)$ and $\textit{featurechange}(R) = \sum_{i=1}^{M} \text{magnitude}(c_i, c'_i)$, accounting for the difficulty and magnitude of changes in feature values.
- Interpretability Metrics: The size of R (the number of triples), the maximum number of predicates in conjunctions, and the number subgroups.

CET. Counterfactual Explanation Tree (CET) [30] is a framework which assigns actions (from a set of actions A) to multiple instances with a decision tree. A CET must satisfy two requirements: *transparency*, i.e., providing a reason for an assigned action in the form of a rule, and *consistency*, ensuring that these

reasons do not conflict with each other. CET reduces a multi-class classification problem to a binary classification between the target class and other classes. Initially, CET models the problem of assigning an effective, single action a for a set X of N instances where $f(x) \neq 1$ for any $x \in X$ in order to alter the prediction result. However, because some of the instances in X are close to the decision boundary while others are far away, this results in the required cost to change the instances being high, and we do not achieve an effective action. Therefore, the invalid score $i(a \mid x)$ is introduced to evaluate the effectiveness of an action with respect to an instance x. Then, CET finds an action $a^* \in A$ by solving the following optimization problem:

$$\text{minimize}_{a \in A}\ g(a \mid X) = \sum_{x \in X} i(a \mid x) \tag{39}$$

CET aims to achieve interpretability of the entire procedure of action assignment and to balance the trade-off between effectiveness and interpretability. Thus, for a set of feasible actions A, a CET is a decision tree that assigns actions to input instances. It uses a set of if-then-else rules structured as a binary tree. Each instance x is assigned an action by following the tree's branching rules from the root to a leaf, determined by conditions on the features. The tree partitions the input space into subspaces, each associated with a specific action and rule. This structure ensures transparency and consistency in the action assignments. The problem of learning a CET h is defined as follows: Given a set X of N instances such that $\forall x \in X,\ f(x) \neq +1$, find an optimal CET h^* such that:

$$\text{minimize}_{h}\ o(h \mid X) = \frac{1}{N} \sum_{x \in X} i(h(x) \mid x) + \lambda \cdot |L(h)| \tag{40}$$

The first term in $o(h \mid X)$ evaluates the average invalidity $i(a \mid x)$ of the actions $a = h(x)$ assigned to $x \in X$, considering a cost $c(a \mid x)$ and the validity loss. The second term $|L(h)|$ represents the total number of leaves, i.e., actions, in h. By adjusting the parameter λ, we can balance the trade-off between the effectiveness of the actions assigned by a CET h and the interpretability of h. CET can be applied to any classifier f and cost function c used in existing methods.

GLOBE-CE. Global and Efficient Counterfactual Explanations (GLOBE-CE) [32] is a flexible framework which provides global counterfactuals explanations (GCEs) to a group of input instances, while facing reliability and scalability issues on high dimensional datasets and in the presence of continuous features. For reliability, GLOBE-CE generates GCEs that lead to accurate conclusions about the model's behavior, ensuring maximum coverage and minimum cost. GLOBE-CE measures efficiency based on the average CPU time required to compute GCEs. For each input x belonging to a particular subgroup, a translation δ with a scalar k is applied so that $x_{CF} = x + k\delta$ becomes a valid counterfactual. For each x the method calculates the minimum value of k necessary for recourse. GLOBE-CE first computes directions δ and then through

scaling it efficiently captures the variation within the set of local instances and their proximity to the decision boundary. Therefore, the main contribution of this work is the notion of *scaling the magnitudes* of translations. Given a group of instances, the GLOBE-CE algorithm firs computes the set of GCE directions $\delta_1, \delta_2, ..., \delta_n$. Next, each GCE δ_i is scaled across a range of m scalars $k_1, k_2, \ldots, k_m$ to provide the counterfactuals of the group. Also, an approach is presented for handling the translation of the categorical data by expressing them in the form of if/then rules.

6 Evaluating CFE Quality

Evaluating CFE quality is not a trivial task and usually involves user inspection. A number of measures have been proposed for CFE quality assessment that are presented below.

Validity. Validity assesses the ratio of generated counterfactuals that indeed have the desired class label compared to the total count of generated counterfactuals. Higher validity is desirable.

Instability. Measures how closely the counterfactuals (set C) obtained for a given instance x align with those (set $\overline{C}$) obtained for its nearest instance $\overline{x}$ within the dataset X, where $\overline{x}$ receives the same black-box decision as x. The underlying principle is that close instances x and $\overline{x}$ should yield comparable explanations. The lower the better

$$\text{inst}(x, \overline{x}) = \frac{1}{1 + d(x, \overline{x})} \cdot \frac{1}{|C||\overline{C}|} \sum_{x' \in C} \sum_{x'' \in \overline{C}} d(x', x'') \tag{41}$$

Dissimilarity. It measures the proximity between instance x and its counterfactuals (set C). We measure it in two ways: dis_{dist}, which calculates the average distance between x and the counterfactuals using various distance functions, and dis_{count}, which measures the average number of features that differ between x and a counterfactual x'.

$$dis_{dist} = \frac{1}{|C|} \sum_{x' \in C} d(x, x') \tag{42}$$

$$dis_{count} = \frac{1}{|C|m} \sum_{x' \in C} \sum_{i=1}^{m} \mathbf{1}(x'_i \neq x_i) \tag{43}$$

where m is the number of features.

Diversity. It applies in the case where multiple counterfactuals are generated. Diversity is promoted by maximizing the distance between multiple counterfactuals [12], incorporated either as an optimization objective term or as a strict constraint [42] or by minimizing the mutual information among every pair of altered features [21].

Running time. The execution time required to generate the explanation is an important evaluation measure.

IM1 and IM2. For the evaluation of interpretability, the authors in [25] propose two measures tailored to algorithmic methods that use autoencoders (AE). Let x_{cf} be the generated counterfactual, i be the class of counterfactual and t_0 be the original class. AE_i is the autoencoder trained on the training instances of the class i and AE_{t0} is the autoencoder trained on the training instances of the class t_0. The IM1 measures the ratio between the reconstruction errors of $x_{cf} = x_0 + \delta$ using AE_i and AE_{t0} and has the following form:

$$IM1(AE_i, AE_{t0}, x_{cf}) := \frac{\|x_0 + \delta - AE_i(x_0 + \delta)\|_2^2}{\|x_0 + \delta - AE_{t0}(x_0 + \delta)\|_2^2 + \epsilon} \tag{44}$$

A smaller $IM1$ value suggests that the counterfactual x_{cf} can be more accurately reconstructed by the autoencoder trained on the counterfactual class i compared to the autoencoder trained on the original class t_0. This indicates that x_{cf} is positioned nearer to the data manifold of counterfactual class i than t_0, which is regarded as more interpretable.

$IM2$ evaluates the resemblance between the reconstructed counterfactual instances generated by AE_i and those produced by an autoencoder AE trained on all classes and is formulated as follows:

$$IM2(AE_i, AE, x_{cf}) := \frac{\|AE_i(x_0 + \delta) - AE(x_0 + \delta)\|_2^2}{\|x_0 + \delta\|_1 + \epsilon} \tag{45}$$

A low value of IM2 means that the reconstructed instances of x_{cf} are very similar when using either AE_i or AE.

7 Counterfactuals and Fairness

Counterfactual explanations can be used to assess fairness of decisions. In [31, 43], the concept of *burden* is introduced, a form of group fairness that is easier to understand and explain than the typical group-fairness metrics. The burden encapsulates the idea that the challenge for individuals or groups to achieve recourses (i.e. to execute the necessary actions to alter their features for a favorable outcome) should be comparable across sensitive groups. In other words the burden for a group G is computed by averaging the distance between the original input feature x_i and the counterfactual feature $x_i{}'$ for all members of

group G. This average distance indicates the level of changes required to assign the group members to the desired class and has the following form:

$$\text{Burden(G)} = \frac{1}{|G|} \sum_{i \in G} distance(x_i, x_i') \tag{46}$$

Recourses provide explainability and actionability to an affected individual. Assume a dataset D, a binary classifier $h : \mathcal{X} \to \{-1, 1\}$ and a set of possible actions A which, when applied to an individual x, results in a counterfactual $x' = a(x)$. Assume also a predicate p which defines a subpopulation group $G_p \subseteq D$ and distinguishes protected groups. The following recourse fairness measures and constraints have been proposed:

- Effectiveness (eff): The proportion of individuals from G to achieve recourse through a specific action a:

$$\text{eff}(a,\text{G}) = \frac{1}{|G|} \left|\{x \in G \,|\, h(a(x)) = 1\}\right|$$

- Aggregate Effectiveness (aeff): How recourse is achieved for the group G through a set of possible actions A. There are two ways to measure. The first way adopts the micro viewpoint, where individuals in a group act independently, choosing the action that benefits them the most. The micro-effectiveness of a set of actions A for group G is defined as the proportion of individuals in G that can achieve recourse through some action in A:

$$\text{aeff}_{\mu}(A, G) = \frac{1}{|G|} \left|\{x \in G \mid \exists a \in A,\ \text{eff}(a, x) = 1\}\right| \tag{47}$$

  The second way adopts the macro viewpoint, the group is treated as a single entity, with one action applied to all its members. Specifically, the macro-effectiveness of a set of actions A for group G is defined as the highest proportion of individuals in G who can achieve recourse through the same action in A:

$$\text{aeff}_{M}(A, G) = \max_{a \in A} \frac{1}{|G|} \left|\{x \in G \mid \text{eff}(a, x) = 1\}\right| \tag{48}$$

- Equal Effectiveness constraint: The proportion of individuals in the protected G_0 and in unprotected G_1 group that can achieve recourse should be the same:

$$\text{aeff}(A, G_0) = \text{aeff}(A, G_1)$$

- Equal Choice for Recourse constraint: Both groups should have an equal choice of 'sufficiently effective' actions for achieving recourse. Sufficiently effective actions are those that are effective for a proportion of the subgroup members greater than ϕ ($\phi \ \in [0, 1]$):

$$\left|\{a \in A \,|\, \text{eff}(a, G_0) \geq \phi\}\right| = \left|\{a \in A \,|\, \text{eff}(a, G_1) \geq \phi\}\right|$$

- Equal Effectiveness within Budget constraint: The proportion of individuals that achieve recourse with a cost at most c should be the same for both groups:
$$ecd(c; A, G_0) = ecd(c; A, G_1)$$

- Equal Cost of Effectiveness constraint: The minimum cost to achieve aggregate effectiveness of $\phi \in [0, 1]$ in both groups should be equal:
$$ecd^{-1}(\phi; A, G_0) = ecd^{-1}(\phi; A, G_1)$$

- Fair Effectiveness-Cost Trade-off: Both groups should have the same effectiveness-cost distribution, or, conversely, their aggregate effectiveness should be equal for every cost budget c:
$$\max_c |ecd(c; A, G_0) - ecd(c; A, G_1)| = 0$$

FACTS. In [31] an efficient, interpretable, model-agnostic framework is presented for examining subgroup fairness through recourses. More specifically, a *recourse cost* function of an individual instance x is defined as the minimum cost effective action a which, when applied to x provides a counterfactual instance belonging to the desired output. They also examine how many individuals from a group G achieve recourse through an action a, i.e, they compute the effectiveness $eff(a, G)$. Given the subgroups of interest, several fairness measures based on effectiveness are defined and computed as described previously. FACTS assesses each of the aforementioned definitions across all subgroups, generating an unfairness score for each definition and each subgroup. The outcome of FACTS is a ranked list of the subgroup counterfactuals in decreasing order of their unfairness score.

8 Summary

In this literature review, we emphasize the significance of explainable AI and the challenges it faces as models grow increasingly powerful and complex. We begin by briefly introducing prominent types of model explanations and then we particularly focus on the counterfactual explanations (CFE) approach for expalining classification decisions. Following an overview of CFEs and their desirable properties, we provide a survey of the relevant literature highlighting several proposed methods for computing CFEs. Our discussion includes both model-agnostic and model-specific CFE approaches, as well as group CFE techniques that compute counterfactuals for groups of instances. Finally, we present evaluation metrics for CFEs and discuss how counterfactual explanations could be used to assess the fairness of classification decisions.

References

[1] C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and machine learning*, vol. 4. Springer, 2006.

[2] K. P. Murphy, *Machine learning: a probabilistic perspective*. MIT press, 2012.

[3] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[4] S. J. Prince, *Understanding Deep Learning*. MIT Press, 2023.

[5] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

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

[7] M. T. Ribeiro, S. Singh, and C. Guestrin, "" 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, 2016.

[8] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *Advances in neural information processing systems*, vol. 30, 2017.

[9] S. Wachter, B. Mittelstadt, and C. Russell, "Counterfactual explanations without opening the black box: Automated decisions and the gdpr," *Harv. JL & Tech.*, vol. 31, p. 841, 2017.

[10] R. Guidotti, "Counterfactual explanations and how to find them: literature review and benchmarking," *Data Mining and Knowledge Discovery*, pp. 1–55, 2022.

[11] A.-H. Karimi, G. Barthe, B. Balle, and I. Valera, "Model-agnostic counterfactual explanations for consequential decisions," in *International conference on artificial intelligence and statistics*, pp. 895–905, PMLR, 2020.

[12] R. K. Mothilal, A. Sharma, and C. Tan, "Explaining machine learning classifiers through diverse counterfactual explanations," in *Proceedings of the 2020 conference on fairness, accountability, and transparency*, pp. 607–617, 2020.

[13] M. T. Keane and B. Smyth, "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, Springer, 2020.

[14] D. Brughmans, P. Leyman, and D. Martens, "Nice: an algorithm for nearest instance counterfactual explanations," *Data mining and knowledge discovery*, pp. 1–39, 2023.

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

[16] A. Artelt, "Ceml: Counterfactuals for explaining machine learning models - a python toolbox." `https://www.github.com/andreArtelt/ceml`, 2019 - 2023.

[17] M. Craven and J. Shavlik, "Extracting tree-structured representations of trained networks," *Advances in neural information processing systems*, vol. 8, 1995.

[18] R. Poyiadzi, K. Sokol, R. Santos-Rodriguez, T. De Bie, and P. Flach, "Face: feasible and actionable counterfactual explanations," in *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pp. 344–350, 2020.

[19] H.-G. Jung, S.-H. Kang, H.-D. Kim, D.-O. Won, and S.-W. Lee, "Counterfactual explanation based on gradual construction for deep networks," *Pattern Recognition*, vol. 132, p. 108958, 2022.

[20] A. Carlevaro, M. Lenatti, A. Paglialonga, and M. Mongelli, "Multi-class counterfactual explanations using support vector data description," *IEEE Transactions on Artificial Intelligence*, 2023.

[21] T. Le, S. Wang, and D. Lee, "Grace: Generating concise and informative contrastive sample to explain neural network model's prediction," in *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 238–248, 2020.

[22] T. Laugel, M.-J. Lesot, C. Marsala, X. Renard, and M. Detyniecki, "Comparison-based inverse classification for interpretability in machine learning," in *Information Processing and Management of Uncertainty in Knowledge-Based Systems. Theory and Foundations: 17th International Conference, IPMU 2018, Cádiz, Spain, June 11-15, 2018, Proceedings, Part I 17*, pp. 100–111, Springer, 2018.

[23] G. Tolomei, F. Silvestri, A. Haines, and M. Lalmas, "Interpretable predictions of tree-based ensembles via actionable feature tweaking," in *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 465–474, 2017.

[24] A. Lucic, H. Oosterhuis, H. Haned, and M. de Rijke, "Focus: Flexible optimizable counterfactual explanations for tree ensembles," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 36, pp. 5313–5322, 2022.

[25] A. Van Looveren and J. Klaise, "Interpretable counterfactual explanations guided by prototypes," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 650–665, Springer, 2021.

[26] K. Rawal and H. Lakkaraju, "Beyond individualized recourse: Interpretable and interactive summaries of actionable recourses," *Advances in Neural Information Processing Systems*, vol. 33, pp. 12187–12198, 2020.

[27] K. Kanamori, T. Takagi, K. Kobayashi, and H. Arimura, "Dace: Distribution-aware counterfactual explanation by mixed-integer linear optimization.," in *IJCAI*, pp. 2855–2862, 2020.

[28] M. Á. Carreira-Perpiñán and S. S. Hada, "Counterfactual explanations for oblique decision trees: Exact, efficient algorithms," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, pp. 6903–6911, 2021.

[29] A. Artelt and B. Hammer, "Efficient computation of counterfactual explanations and counterfactual metrics of prototype-based classifiers," *Neurocomputing*, vol. 470, pp. 304–317, 2022.

[30] K. Kanamori, T. Takagi, K. Kobayashi, and Y. Ike, "Counterfactual explanation trees: Transparent and consistent actionable recourse with decision trees," in *International Conference on Artificial Intelligence and Statistics*, pp. 1846–1870, PMLR, 2022.

[31] L. Kavouras, K. Tsopelas, G. Giannopoulos, D. Sacharidis, E. Psaroudaki, N. Theologitis, D. Rontogiannis, D. Fotakis, and I. Emiris, "Fairness aware counterfactuals for subgroups," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[32] D. Ley, S. Mishra, and D. Magazzeni, "Globe-ce: a translation based approach for global counterfactual explanations," in *International Conference on Machine Learning*, pp. 19315–19342, PMLR, 2023.

[33] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to algorithms*. MIT press, 2022.

[34] W.-Y. Loh, "Classification and regression trees," *Wiley interdisciplinary reviews: data mining and knowledge discovery*, vol. 1, no. 1, pp. 14–23, 2011.

[35] J. R. Quinlan, "Induction of decision trees," *Machine learning*, vol. 1, pp. 81–106, 1986.

[36] D. M. Tax and R. P. Duin, "Support vector data description," *Machine learning*, vol. 54, pp. 45–66, 2004.

[37] G. Plumb, J. Terhorst, S. Sankararaman, and A. Talwalkar, "Explaining groups of points in low-dimensional representations," in *International Conference on Machine Learning*, pp. 7762–7771, PMLR, 2020.

[38] M. A. Carreira-Perpinán and P. Tavallali, "Alternating optimization of decision trees, with application to learning sparse oblique trees," *Advances in neural information processing systems*, vol. 31, 2018.

[39] M. A. Carreira-Perpinán, "The tree alternating optimization (tao) algorithm: A new way to learn decision trees and tree-based models," *arXiv*, 2021.

[40] S. Dutta, J. Long, S. Mishra, C. Tilli, and D. Magazzeni, "Robust counterfactual explanations for tree-based ensembles," in *International conference on machine learning*, pp. 5742–5756, PMLR, 2022.

[41] E. Carrizosa, J. Ramírez-Ayerbe, and D. R. Morales, "Mathematical optimization modelling for group counterfactual explanations," *European Journal of Operational Research*, 2024.

[42] B. Ustun, A. Spangher, and Y. Liu, "Actionable recourse in linear classification," in *Proceedings of the conference on fairness, accountability, and transparency*, pp. 10–19, 2019.

[43] S. Sharma, J. Henderson, and J. Ghosh, "Certifai: Counterfactual explanations for robustness, transparency, interpretability, and fairness of artificial intelligence models," *arXiv preprint arXiv:1905.07857*, 2019.