MÁSTER EN TRATAMIENTO ESTADÍSTICO
COMPUTACIONAL DE LA INFORMACIÓN

TRABAJO FIN DE MASTER

# Multicriteria Model-Agnostic Counterfactual Explainability

**Ignacio Fernández Sánchez-Pascuala**


UNIVERSIDAD
COMPLUTENSE
MADRID


POLITÉCNICA

UNIVERSIDAD COMPLUTENSE DE MADRID
FACULTAD DE CIENCIAS MATEMÁTICAS
&
UNIVERSIDAD POLITÉCNICA DE MADRID
ETSI TELECOMUNICACIÓN

---

Tutor: Pedro José Zufiria Zatarain

Madrid, September 2024

# Agradecimientos

Este Trabajo de Fin de Máster (TFM) pone el broche a mi trayectoria académica hasta ahora. Han sido años duros de gran esfuerzo con muchos altibajos, pero a la vez muy gratificantes. Es un orgullo ver cómo se han cumplido cada uno de los objetivos académicos que me he propuesto, gracias en parte a aquellos que me han acompañado durante el camino. Por ello, quiero dedicarles unas palabras de agradecimiento:

*Quiero agradecer a mi tutor, Pedro J. Zufiria, por motivarme, proporcionar los recursos necesarios para completar este trabajo, y ayudarme a publicar mi primer artículo de investigación junto a él y Cristian R. Rojas en una conferencia. Él me introdujo en esta emocionante área dentro del aprendizaje automático, la cual desconocía previamente.*

*Estoy también muy agradecido a mis padres por su constante apoyo y por apostar en mi desarrollo académico e intelectual. Su presencia ha sido uno de los pilares de mi crecimiento.*

*A mi novia María y mis amigos, gracias por estar ahí emocionalmente y animarme durante este año lleno de trabajo duro y experiencias increíbles. Su apoyo y confianza han significado mucho para mí.*

*Finalmente, quiero agradecer a mis compañeros de máster y a todos los profesores por crear un ambiente de motivación y trabajo en equipo. Aprender unos de otros ha sido una parte fantástica de este viaje.*

Tras este trabajo, mi camino en este máster acaba, pero no mis ganas de seguir formándome y progresando para conseguir un mundo mejor.

# Abstract

In recent years, the necessity for understanding and interpreting the decisions made by machine learning models has become increasingly critical. This need arises from the importance of ensuring model transparency, increasing user confidence, identifying and reducing biases, and complying with regulatory requirements.

This master's thesis focuses on a local, model-agnostic, post-hoc technique known as Counterfactual Explainability (CE). CE aims to identify the minimal changes needed in an input instance to alter the model's prediction to a desired outcome.

The proposed multi-criteria framework addresses the varying importance of input variables, providing a flexible approach to managing the relative and subjective importance of different potential counterfactuals. Multi-criteria optimization is applied to generate realistic and actionable counterfactuals, taking into account various objectives like validity, proximity, simplicity, and actionability.

Practical applications in healthcare (predicting heart failure events) and finance (bank loan approvals) demonstrate the effectiveness of the methodology.

This research contributes in the field of machine learning explainability, offering a robust solution for generating meaningful counterfactual explanations.

**Keywords:**

Counterfactual Explainability, Multi-Criteria Optimization, Machine Learning, Transparency, Interpretability

# Resumen

En los últimos años, la necesidad de comprender e interpretar las decisiones tomadas por modelos de aprendizaje automático se ha vuelto cada vez más crítica. Esta necesidad surge de la importancia de asegurar la transparencia del modelo, aumentar la confianza del usuario, identificar y reducir sesgos, y cumplir con las normativas.

Este Trabajo de Fin de Máster se centra en una técnica local, independiente del modelo, y posterior al entrenamiento, conocida como Explicabilidad Contrafactual (CE). La CE tiene como objetivo identificar los cambios mínimos necesarios en una instancia de entrada para alterar la predicción del modelo hacia un resultado deseado.

El marco de trabajo propuesto, basado en un enfoque multicriterio, aborda la importancia variable de los inputs de entrada, proporcionando un enfoque flexible para gestionar la importancia relativa de diferentes soluciones contrafactuales potenciales. La optimización multi-criterio se aplica para generar soluciones realistas, teniendo en cuenta varias propiedades como la validez, proximidad, simplicidad y capacidad de realización del cambio.

Las aplicaciones mostradas en salud (predicción de individuos con insuficiencia cardíaca) y en finanzas (aprobaciones de préstamos bancarios) demuestran la eficacia de la metodología.

Esta investigación proporciona una nueva contribución en el campo de la explicabilidad del aprendizaje automático, ofreciendo una solución robusta para generar explicaciones contrafactuales significativas.

**Palabras clave:**

Explicabilidad Contrafactual, Optimización Multi-Criterio, Aprendizaje Automático, Transparencia, Interpretabilidad

# Contents

# 1 Introduction

In recent years, the need to understand and interpret the predictions or decisions made by machine learning models has become increasingly critical. This necessity stems from the imperative to ensure model transparency, increase user confidence, identify and mitigate biases, and comply with regulatory requirements. Machine learning models are now integral to critical applications in healthcare, finance, or criminal justice, where the implications of their decisions can be profound [1, 2, 3, 4].

To address these concerns, various techniques have been developed to explain the decisions of machine learning models. These techniques can be categorized based on different criteria. Depending on the explanation scope, we distinguish between global explanations, which attempt to elucidate the overall behavior of a classifier, and local explanations, which focus on explaining specific predictions or decisions made by the model. Furthermore, model-specific techniques are tailored for particular types of models, while model-agnostic techniques are applicable across various models, offering a more general understanding of predictions [5]. Additionally, some methods incorporate interpretability during the training process of classifiers, whereas post-hoc methods provide explanations after the model has been trained, without altering its internal architecture or parameters [6].

In this master's thesis, building on the work of [7], from which some parts have been directly taken, we focus on a local, model-agnostic, post-hoc technique known as Counterfactual Explainability (CE) [1, 6, 8, 9]. CE aims to identify the minimal changes needed in an input instance to alter the prediction of the model to a desired outcome. For example, in a loan application scenario, CE would determine the smallest changes in an applicant's financial profile that would change the loan application result from rejection to approval. This method provides actionable insights for users, helping them understand what changes could lead to different outcomes.

A critical aspect of CE is quantifying the modifications in the input space. This is typically done using distance metrics that reflect the relative importance of each input variable. However, defining such distances can be challenging, especially for categorical variables [10]. This thesis proposes a multi-criteria framework to handle the varying importance of input variables, providing a flexible approach to managing the relative and subjective importance of different potential counterfactuals.

Multi-criteria optimization has been applied in various areas of machine learning, such as robust model training [11, 12], designing model ensembles [13, 14], and enhancing explainability [15]. Notably, previous work has used the Gower distance [16] to balance multiple objectives in CE [17]. However, this approach imposes a fixed relative importance among features. Our framework extends beyond a single distance metric by considering separate distances for various features in the input space.

We formalize the CE problem within this multi-criteria framework and

provide insights into how the resulting CE Pareto set is influenced by the number of actionable variables and the structure of the classifiers. Furthermore, we extend this problem to scenarios where the classifier is not directly accessible, using only a dataset of classified instances, and also to regression problems. We propose a procedure that utilizes multiple approximating classification models, combining their outputs to improve the approximation of the CE Pareto set.

This thesis also demonstrates the practical application of our methodology through real-world examples in healthcare (predicting heart failure events) and finance (bank loan approvals). These examples illustrate the effectiveness of our approach in generating realistic and actionable counterfactual explanations (CEs).

## 1.1 Structure of the Thesis

The structure of this thesis is as follows:

- **Section 1: Introduction:** Provides context, outlines the objectives and contributions, and presents the structure of the thesis.
- **Section 2: Machine Learning Explainability:** Reviews the various methods and techniques used to explain machine learning models, categorizing them according to different criteria.
- **Section 3: Counterfactual Explanations:** Details the concept of counterfactual explanations (CEs), their importance, and the desirable properties for effective CEs.
- **Section 4: Proposed Framework:** Introduces our multi-criteria framework for CE, outlining the problem definition, multi-criteria distance formulations, and selection of actionable input variables.
- **Section 5: CE When Only Samples Are Available:** Discusses the scenario where only sample data is available and outlines our approach for generating CEs in such cases.
- **Section 6: Proposed Algorithm:** Describes the algorithm used to implement our framework, including the genetic algorithm NSGA-III and distance metrics for different variable types.
- **Section 7: Examples of Application:** Demonstrates the application of our methodology through practical examples in healthcare and finance.
- **Section 8: Conclusions:** Summarizes the findings and contributions of the thesis.
- **Section 9: Future Work:** Suggests potential areas for future research to further advance the field of machine learning explainability.

# 2 Machine Learning Explainability

Machine Learning Explainability (MLX) refers to the methods and techniques used to make the behavior and predictions of machine learning models understandable to humans. The importance of explainability in machine learning has grown significantly as these models are increasingly used in critical applications such as healthcare, finance, and criminal justice.

Understanding the decision-making process of models helps in building trust, ensuring transparency, detecting biases, and complying with regulatory requirements, which is crucial for ensuring that these systems operate fairly and ethically [1, 8].

Approaches to explainability in machine learning can be categorized according to different criteria, and these categories are not mutually exclusive. Here are the main types of explainability:

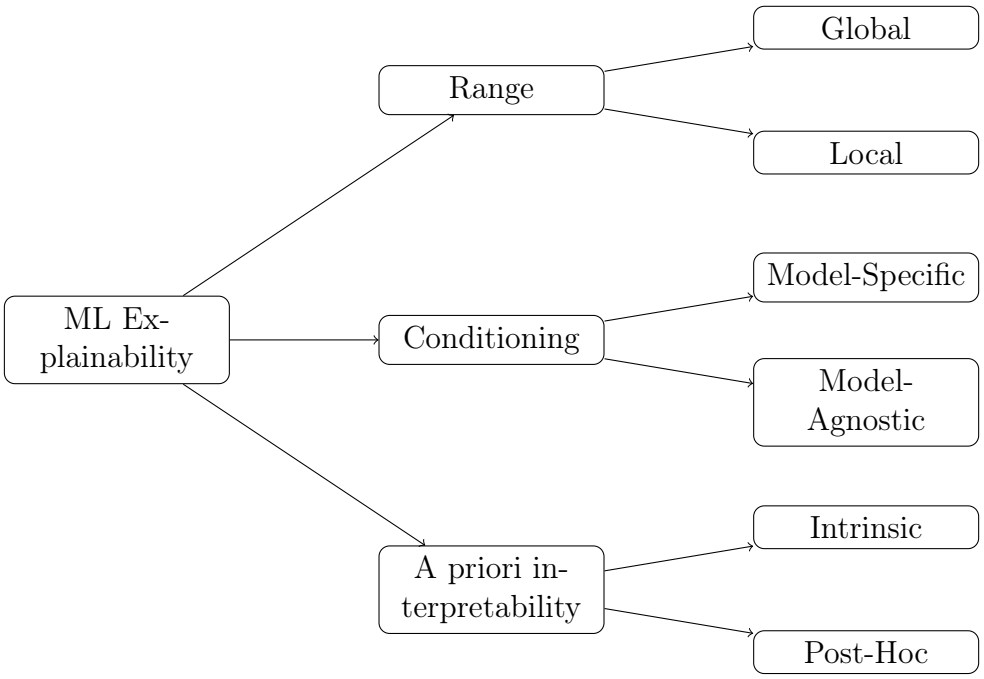


Figure 1: Criteria for classifying Machine Learning Explainability

## 2.1 Global vs. Local

Global range explanations aim to provide insights into the overall behavior of a model, offering a comprehensive view of how different features affect the predictions. Techniques like feature importance, partial dependence plots, and global surrogate models are commonly used for this purpose.

On the other hand, local ones focus on individual predictions, explaining why a specific decision was made for a particular instance. Methods such as LIME (Local Interpretable Model-agnostic Explanations), SHAP (SHapley Additive exPlanations), and Counterfactual Explanations (CEs) fall under

this category [8, 18].

## 2.2 Model-Specific vs. Model-Agnostic

Model-specific methods are tailored to specific types of machine learning models. For example, interpreting decision trees by visualizing the tree structure, or using gradient-based methods for neural networks. These methods leverage the internal structure of the model to provide explanations.

Model-agnostic methods, however, can be applied to any machine learning model. These techniques do not require access to the model internal workings and they can be used with a variety of models, offering greater flexibility. Examples include permutation feature importance, partial dependence plots, and local surrogate models [8].

## 2.3 Intrinsic vs. Post-Hoc

Intrinsic interpretability methods involves using inherently interpretable models such as linear regression, decision trees, and rule-based systems, which are designed to be understandable by default.

Post-hoc interpretability refers to methods applied after a model has been trained, aiming to explain the model behavior without modifying its structure. Techniques such as feature importance scoring, visualizations, and CEs are examples of post-hoc interpretability methods [8].

The choice of explainability technique depends on the specific needs of the application, the type of model used, and the level of detail required for the explanations.

# 3 Counterfactual Explanations

Counterfactual Explanations (CEs), classified as a local, model-agnostic, and post-hoc technique, have emerged as a crucial approach in response to the growing need to interpret and understand the decisions made by machine learning models. The primary objective of CEs is not only to identify the reasons behind a particular decision but also to provide concrete ways to change that decision in the future.

The core idea is to determine what small changes in an individual's input features could alter the outcome of a classification model, shifting an individual from a negative to a positive class. This methodology not only offers an interpretation of the current decision but also provides practical guidance for affected users, enabling them to understand and act on the factors that can improve their future outcomes [9].

As an example, consider a scenario where a person applies for a loan and the application is rejected. CEs would identify the smallest changes in the applicant financial situation of the applicant that would result in the loan being approved.

Figure 2 illustrates this concept by showing the current situation of the applicant and the potential changes needed to move from the rejection region to the acceptance region of the model's decision boundary. It is important to note that this image is illustrative of how an individual can move to different counterfactual solutions, but it does not represent the actual dimension and proximity of these changes.

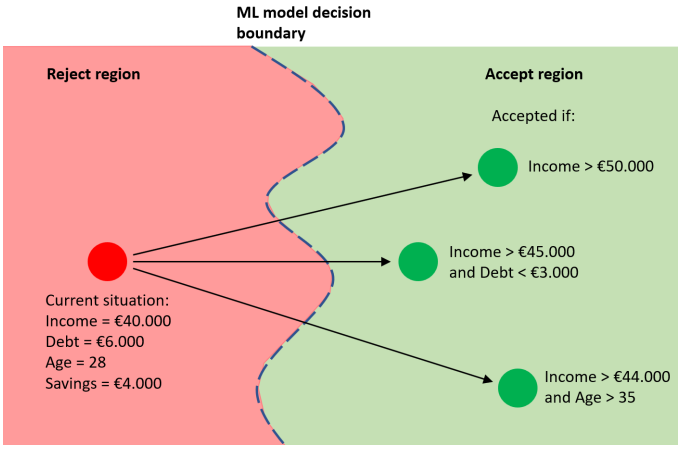


Figure 2: Example of CEs in a loan application scenario. This image is sourced from `https://vis.win.tue.nl/masterprojects/100/`.

## 3.1 History, Evolution, and Importance

The concept of counterfactual explanations dates back to the early developments in artificial intelligence and machine learning interpretability. Initially, these explanations were rooted in the need to understand and justify

decisions in expert systems and rule-based models. Over time, as machine learning models grew more complex and opaque, the necessity for transparent and interpretable AI systems became paramount.

CEs gained prominence as a means to provide clear and actionable insights into model decisions, especially in high-stakes domains such as finance, healthcare, and legal frameworks [1]. They play a pivotal role in enhancing the transparency of machine learning models, increasing user trust, identifying biases, and complying with regulatory requirements.

By offering a clear path to transform a negative outcome into a positive one, CE provide users with a powerful tool to understand and improve their personal situation. This capability is particularly significant in areas where algorithmic decisions can have profound impacts on the lives of individuals [2].

## 3.2 Desired Properties

For a counterfactual explanation to be effective, it must satisfy several desirable properties. These properties can be categorized into those that pertain directly to the quality of the counterfactual solutions and those that describe the characteristics of the algorithms generating these solutions:

**Properties Related to Optimization Objective**

- **Validity**: The explanation must ensure that the proposed change will indeed alter the decision made by the model. The objective is to minimize the distance between the counterfactual instance and the original data point while ensuring the model output changes to the desired class. However, some frameworks, especially those using approximate methods, do not guarantee a strict change in decision but instead find solutions that are very close to the decision boundary.

- **Proximity**: The suggested changes should be minimal, making them easier to implement. This is achieved by minimizing the distance between the original and counterfactual instances. The notion of proximity in dimensions greater than one can be relative and subjective to the user, depending on their context. Different metrics can be used to measure this distance, such as L1/L2 norms, quadratic distances, or more user-centric measures that account for the perceived cost of feature changes.

- **Simplicity**: Explanations should be comprehensible, involving the least number of feature changes. This is often referred to as sparsity, where the goal is to alter as few features as possible to achieve the desired outcome. Simplicity enhances user understanding and trust in the explanation.

- **Actionability**: Recommendations should be realistic and feasible within the user context. This includes ensuring that changes to immutable features (such as race or country of origin) are not suggested. Actionability also considers user preferences and practical constraints, ensuring that the proposed changes are not only possible but also meaningful and relevant to the user situation.

- **Diversity**: Providing multiple CEs allows users to choose from various viable options. Diversity can accommodate different user preferences and constraints, enhancing the overall utility and satisfaction with the explanations provided.

- **Data Manifold Closeness**: The proposed modifications should be plausible and aligned with the distribution of the training data. Counterfactuals that are too far from the training data may be unrealistic or implausible. Ensuring proximity to the data manifold helps maintain the credibility and feasibility of the suggested changes.

  Figure 3 illustrates this concept with two possible paths for a datapoint (shown in blue), originally classified in the negative class, to cross the decision boundary. The endpoints of both paths (shown in red and green) are valid counterfactuals for the original point. Note that the red path is the shortest, whereas the green path adheres closely to the manifold of the training data but is longer.

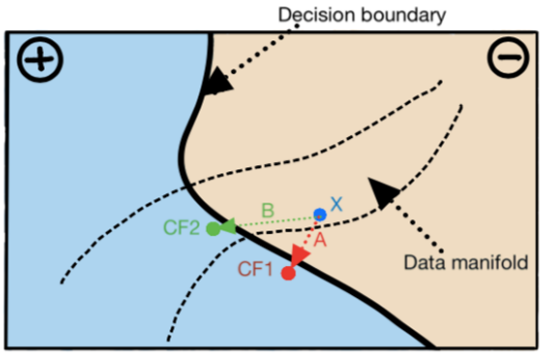


Figure 3: Illustration of Data Manifold Closeness in CEs [9].

- **Causality**: Features in a dataset are rarely independent; hence, changes to one feature can affect others. A realistic counterfactual should maintain known causal relationships between features. For example, obtaining a new educational degree typically involves an increase in age. Incorporating causality ensures that the proposed changes are realistic and coherent.

## Properties Related to Algorithms Generating Counterfactuals

- **Amortized Inference**: Generating counterfactuals can be computationally expensive. Techniques like those proposed by Mahajan [19]

and Verma [20] aim to pre-train models to quickly generate counterfactuals without solving an optimization problem for each new instance. This makes the process more efficient and scalable.

- **Model Agnosticity**: An approach that is model-agnostic can work with various types of machine learning models, making it more versatile and widely applicable. This property ensures that the counterfactual generation method is not limited to specific model types.
- **Black-box Access**: The ability to generate counterfactuals with only access to the model's prediction function, without needing full transparency of the model, is crucial in scenarios where models are proprietary or sensitive. Black-box access implies model agnosticity, as it allows working with different types of models without requiring detailed knowledge of their internal workings. Methods such as genetic algorithms and reinforcement learning-based approaches facilitate this kind of access, enhancing the versatility of CEs.

The development of CE continues to evolve, incorporating increasingly complex constraints to ensure they are actionable and helpful. Research in this field aims to balance these properties in a way that is generalizable across algorithms and computationally efficient. We will explore how our framework achieves positive outcomes by fulfilling most, if not all, of these properties.

## 3.3 State of the Art

In recent years, there has been significant progress in developing methods of Counterfactual Explainability in Machine Learning. A comprehensive review by Verma [9] outlines numerous approaches to generating CEs, highlighting their strengths and weaknesses in different contexts. Below, we stand out some of these approaches.

Wachter [1] proposed a method that formulates the CEs generation as an optimization problem, aiming to minimize the distance between the original and the counterfactual instances subject to the constraint that the counterfactual changes the prediction of the model. This method is widely used due to its simplicity and general applicability.

DiCE (Diverse Counterfactual Explanations) provides multiple diverse CEs for a single instance. It uses a gradient-based search technique to find counterfactuals that not only change the prediction of the model but are also diverse from each other, giving the user various options to change their outcome [21].

Karimi [22] introduced a model-agnostic method for generating CEs that emphasizes the consequential nature of changes. This approach leverages *satisfiability modulo theories* (SMT)[1] to find counterfactuals efficiently, ensuring

[1]SMT is the problem of determining whether a mathematical formula is satisfiable

that the proposed changes lead to meaningful and actionable differences in the outcomes.

Sharma [23] proposed a method using a genetic programming approach to generate counterfactuals that not only change the prediction of the model but are also realistic and aligned with observed data distributions.

Mahajan [19] presented a method to generate CEs that preserve causal constraints. This approach addresses the feasibility of counterfactuals by ensuring that the proposed changes adhere to the causal relationships between features, thereby generating more realistic and actionable explanations.

Dandl [17] introduced Multi-Objective Counterfactual Explanations (MOC), formulating the generation of counterfactuals as a multi-objective optimization problem. This approach explicitly balances multiple objectives, such as minimizing feature changes, maintaining proximity to observed data, and achieving the desired outcome, using the NSGA-II algorithm to provide a diverse set of solutions. However, Dandl's method does not apply the multi-objective approach specifically to the features themselves, as our method does.

Naumann and Ntoutsi [24] proposed a multi-objective sequential approach to counterfactual generation, recognizing that changes in features often need to occur in a specific order due to causal relationships or practical constraints. Their method uses a genetic algorithm to find optimal sequences of actions.

GeCo, introduced by Schleich [25], presents a real-time system for generating CEs that balances plausibility and feasibility constraints. By leveraging a custom genetic algorithm, GeCo achieves high-quality explanations efficiently, making it suitable for interactive applications.

These advancements illustrate the diversity of approaches in the field of CE, each addressing different aspects of the problem and offering unique benefits. Our proposed method builds on these ideas by introducing a multi-objective framework that incorporates all features. Using the NSGA-III algorithm, we handle both categorical and continuous variables, incorporate various constraints, and ensure realistic and causality-preserving counterfactuals, thereby advancing the state of the art in multi-objective CEs.

## 3.4 Other Applications of CE

As mentioned above, the primary application of CE is to determine the changes needed in an individual's input features to alter the outcome of a classification model, shifting an individual from a negative to a positive class. However, CE is also being utilized in various other domains. These additional applications demonstrate the versatility and potential of CE in addressing different challenges [9]:

**Anomaly and Data-Drift Detection:** Counterfactual explanations can be used to detect anomalies and data drift in time-series datasets. They help in explaining the anomalies by identifying minimal changes needed to align the data point with the expected behavior [26, 27].

**Training Dataset Debugging:** CEs assist in debugging training datasets by diagnosing the behavior of models and using synthetic data to alter decision boundaries. This helps in identifying mislabeled data and detecting bugs in financial models [28].

**Data Augmentation:** CEs are used to augment training data, which is particularly useful in addressing class imbalance problems and enhancing robustness. They generate synthetic data points that can improve the performance and fairness of machine learning models [29].

**Drug Designing:** In the field of drug design, CEs help in identifying modifications to drug and protein molecules that increase their affinity, thus aiding in the development of more effective pharmaceuticals [30].

**Bias Detection in Machine Learning Models:** CEs are employed to detect biases in machine learning models, ensuring that the models are fair and do not discriminate against certain groups [31, 32].

# 4 Proposed Framework

A crucial aspect of Counterfactual Explainability (CE) is the quantification of the sizes of modifications needed to change a classifier decision. The usual approach involves defining a distance in the input space to reflect the greater or lesser relative importance of each input variable. This method can present problems related to the relative scales of different variables, especially categorical ones.

To address these issues, we employ a multi-criteria formulation on the input variables proposed in [7]. This formulation allows for flexible management of the relative and subjective importance of different attainable counterfactuals. By incorporating multiple criteria, our method can better handle the varying scales and types of input variables.

Our proposal is designed to be model-agnostic and black-box accessible. This means that it only requires access to the prediction function of the classifier, without needing to understand or modify its internal workings. As a result, our approach can be applied to all types of models, offering a versatile solution for generating CEs.

## 4.1 Problem Definition

Let us consider a binary classifier such that

$$Y = C(x),\ x \in X, \tag{1}$$

where $C\colon X \to \{-1, 1\}$ is defined over a feature space $X$.

Given a single input value $\hat{x}$, provide the *smallest* modifications (in a multicriteria wide sense) of such input that would lead to a different classification output. The multicriteria formulation will allow to address in a general framework all favorable joint modifications of the values of several components of $\hat{x}$ (i.e., several input variables).

We formalize multicriteria CEs via the following problem: given a reference input value $\hat{x}$, our first goal is to find the input value $\tilde{x} \in X$ which is *closest* or *most similar* to $\hat{x}$ such that $C(\tilde{x}) = -C(\hat{x})$.

**Remark 1:** *The framework to be proposed in this work can be easily generalized to multiclass classifiers. We shall confine ourselves to binary classifiers for ease of exposition.*

Assuming that the *closeness* between $\hat{x}$ and $\tilde{x}$ is described via a dissimilarity function $d_X\colon X \times X \to \mathbb{R}^m$ which satisfies:

- $(d_X)_i\colon X \times X \to \mathbb{R}$ for each $i = 1, \ldots, m$ is a pseudometric, and
- if $d_X(x, x') = 0$ then $x = x'$ (distinguishable),

the problem can be formulated as:

$$\tilde{x} = \arg\min_{x \in X} d_X(\hat{x}, x), \quad \text{subject to} \quad C(x) = -C(\hat{x}) \tag{2}$$

The definition of this problem is grounded on the concepts of *dominance* and *Pareto efficiency* [33].

## 4.2 Dominance and Pareto Efficiency

To understand the solution of problem (2), we first need to define the concepts of dominance and Pareto efficiency:

**Weak Dominance:** Given $x, x' \in X$, $x'$ is said to *weakly dominate* $x$ iff:

$$(d_X)_i(\hat{x}, x') \leq (d_X)_i(\hat{x}, x) \quad \text{for all} \quad i = 1, \ldots, m.$$

**Dominance:** Furthermore, $x'$ is said to *dominate* $x$ if $x'$ *weakly dominates* $x$ and:

$$d_X(\hat{x}, x') \neq d_X(\hat{x}, x).$$

**Pareto Efficiency:** A point $x \in X$ is said to be *Pareto Efficient* iff there is no $x' \in X$ such that $x'$ *dominates* $x$.

**Strict Pareto Efficiency (SPE):** A point $x \in X$ is said to be *Strict Pareto Efficient* (SPE) iff there is no $x' \in X$, $x' \neq x$, such that $x'$ *weakly dominates* $x$.

**Pareto Set and Pareto Front:** The set $P \subset X$ of Pareto Efficient points in (2) is denoted as the *Pareto set*, and the set $F = d_X(\hat{x}, P) \subset \mathbb{R}^m$ is denoted as the *Pareto front*.

In the CE framework, we will refer to:

- $P_C(\hat{x})$ as the *CE Pareto set* associated with $\hat{x}$ for the classifier $C$.
- $SP_C(\hat{x})$ as the *CE Strict Pareto set* associated with $\hat{x}$ for the classifier $C$.

Figure 4 shows a visual representation of the Pareto front in a two-objective optimization problem. The shaded region represents feasible solutions, while the Pareto front consists of non-dominated solutions. This visualization helps to understand the trade-offs in our multi-criteria formulation, enabling us to identify the smallest modifications needed in input variables to achieve different classification outcomes while balancing multiple criteria.

Therefore, the multi-objective search framework provides as a result a whole family of input configurations which are Pareto Efficient. This way,

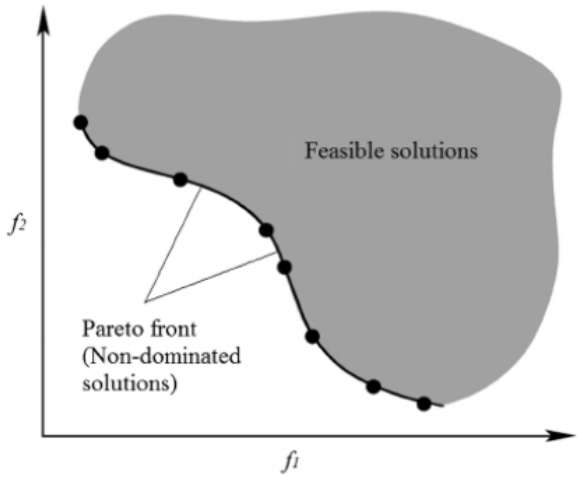


Figure 4: Pareto front in the context of multi-criteria optimization

the designer can expediently select one of these configurations by making use of additional criteria, favoring properties of *Diversity* and *Actionability*.

Note that even if $X$ is a compact space, the problem may not have a solution since, in general, $C(x)$ will not be continuous. Nevertheless, one can always approximate $\inf_{x \in X} d_X(\hat{x}, x)$ s.t. $C(x) = -C(\hat{x})$, which, due to the continuity of $d_X$, would provide points $x$ that may not satisfy $C(x) = -C(\hat{x})$ but are arbitrarily close to points that do satisfy it.

## 4.3 Multi-Criteria Distance Formulations

The dissimilarity function $d_X$ may potentially take into account up to $m$ different indices. Three natural scenarios are elaborated hereafter.

### 4.3.1 Case $m = 1$

For $m = 1$, $d_X$ may be a standard metric on $X$. For instance, if $X \subset \mathbb{R}^n$, a possible definition for $d_X$ could be the usual Euclidean distance:

$$d_X(\hat{x}, x) = d_2(\hat{x}, x) = \sqrt{\sum_{i=1}^{n} d^2_{X_i}(\hat{x}_i, x_i)}, \tag{3}$$

where $d_{X_i}(\cdot, \cdot),\ i = 1, \ldots, n$, represents a one-dimensional distance in the $i$-th component of $X$. A natural choice for such distance could be, for instance, the Euclidean one-dimensional distance $d_{X_i}(\hat{x}_i, x_i) = |\hat{x}_i - x_i|,\ i = 1, \ldots, n$.

Alternatively, the distance $d_X$ may depend on the relative relevance of the different vector components. If such relative importance of the components can be quantified a priori (e.g., via some weights $w_i > 0,\ i = 1, \ldots, n$), a

weighted distance $d_W(\cdot,\cdot)$ can be employed:

$$d_X(\hat{x},x) = d_W(\hat{x},x) = \sum_{i=1}^{n} w_i d_{X_i}(\hat{x}_i, x_i). \tag{4}$$

Note that many alternative weighted combinations of different distances can obviously be defined.

When a standard metric in employed in $X$, one standard model-agnostic scheme for finding CEs is the *Growing spheres* algorithm [34]. This algorithm generates uniformly random points within a sphere defined upon a given distance on the whole space of input features, so that an iterative construction of these spheres with increasing radius leads to the location of approximate solutions of (2).

#### 4.3.2 Case $m = n > 1$

This approach is appropriate when the relative importance among all the factors (input variables) is not restricted a priori. For instance, if $X \subset \mathbb{R}^n$, a possible *Multicriteria (M)* definition for $d_X$ (with $m = n$) could be:

$$d_X(\hat{x},x) = d_M(\hat{x},x) = \big(d_{X_1}(\hat{x}_1,x_1),\ldots,d_{X_n}(\hat{x}_n,x_n)\big), \tag{5}$$

where $(d_X)_i = d_{X_i}$ satisfy the conditions stated above (i.e., pseudometrics and together distinguishable).

As explained above, the solution of Problem (2) with (5) is a Pareto set $P \subset X$ of non-dominated points together with the corresponding Pareto front $F = d_M(\hat{x}, P)$. Such Pareto set $P$ contains, for instance, the solution of (2) with (3) as well as the solution of (2) with (4) for any specific quantification of the relative relevance among the factors in (4). Furthermore, problem (2) with (5) is more general than those problems that use scalar distances which can be derived from (5) via a scalarization scheme $S\colon \mathbb{R}^n \to \mathbb{R}$ so that

$$d_X(\hat{x},x) = d_S(\hat{x},x) = S(d_M(\hat{x},x)) \tag{6}$$

is a distance in $X$.

Usually, $d_S$ will be designed to gather the relative importance among the factors (both $d_W$ and $d_2$ are specific cases of $d_S$). Provided that $d_S$ satisfies some properties [35], the (maybe unique) solution of (2) with (6) should be contained in the Pareto set $P$ of (2) with (5).

#### 4.3.3 Case $m > n$

The multi-objective approach allows incorporating additional measures into our distance function if needed, beyond just the distances in the input coordinates. For instance, when available, the distance between $\hat{x}$ and a

sample $X_{\text{obs}} \subset X$ that represents the distribution associated with $X$ (*Data Manifold Closeness*) could be added as an objective to minimize.

Redefining the minimization problem for this case:

$$\tilde{x} = \arg\min_{x \in X} \left[ d_X(\hat{x}, x), d(x, X_{\text{obs}}) \right], \quad \text{subject to} \quad C(x) = -C(\hat{x}) \tag{7}$$

This formulation allows for a more comprehensive approach to CEs by ensuring that the proposed modifications are not only minimal but also likely and realistic within the context of the observed data distribution.

Another option is, once the Pareto Set is obtained, to select the counterfactual solutions that are closest to $X_{\text{obs}}$ as the final set.

As an example of calculating the distance between $\hat{x}$ and $X_{\text{obs}}$, we can use the weigthed Gower distance. First, we define the Gower distance $d_G$ between two points $\hat{x}$ and $x^*$:

$$d_G(\hat{x}, x^*) = \frac{1}{p} \sum_{j=1}^{p} d(\hat{x}_j, x_j^*)$$

Now, to define the distance between $\hat{x}$ and $X_{\text{obs}}$, we use the weighted average Gower distance between $\hat{x}$ and the $k$ nearest observed data points $x_{[1]}, \ldots, x_{[k]} \in X_{\text{obs}}$ as an empirical approximation of how likely $\hat{x}$ originates from the distribution associated with $X$ [17]:

$$d(\hat{x}, X_{\text{obs}}) = \sum_{i=1}^{k} w_{[i]} d_G(\hat{x}, x_{[i]}) \tag{8}$$

where $\sum_{i=1}^{k} w_{[i]} = 1$.

It is important to standardize or scale the distances between features $d(\hat{x}_j, x_j^*)$ to ensure that each feature contributes equally to the Gower distance metric.

This additional measure has not been implemented in the applications discussed due to simplicity and computational constraints, but the framework and algorithm certainly allow for its inclusion.

## 4.4 Selection of actionable input variables

In general, there may be several reasons for restricting the search in problem (2) to only some of the features of the input domain $X$. To begin with, some of these features may not be modifiable in practice; for instance, if $\hat{x}$ corresponds to an individual who wants to modify such input values to change his/her classification output, an existing variable describing the age could not be modified. Additionally, there can be features that are not included in the model, meaning that their variation will not affect the change of decision or classification output.

Furthermore, explainability is claimed to be better when only few modifications are proposed to individuals [9]. Finally, it may not be easy to provide meaningful metrics for some non-continuous (e.g., qualitative) variables.

If we decompose $X = X^f \times X^a$ into a *fixed* and an *actionable* (or adjustable) part, then we can write any $x \in X$ as $x = (x^f, x^a)$ with $x^f \in X^f$, $x^a \in X^a$; in particular, $\hat{x} = (\hat{x}^f, \hat{x}^a)$, where $\hat{x}^a$ represents the actionable part of the single individual input we may want to characterize. Hence, (2) can be particularized to

$$\begin{aligned}
\tilde{x}^a &= \arg\min_{x^a \in X^a} d_X(\hat{x}, (\hat{x}^f, x^a)) \\
&= \arg\min_{x^a \in X^a} (d_X^f(\hat{x}^f, \hat{x}^f), d_X^a(\hat{x}^a, x^a)) \\
&= \arg\min_{x^a \in X^a} (\underbrace{0, \ldots, 0}_{n_f \text{ elements}}, d_X^a(\hat{x}^a, x^a)) \\
&= \arg\min_{x^a \in X^a} d_X^a(\hat{x}^a, x^a), \\
&\text{subject to } C((\hat{x}^f, x^a)) = -C(\hat{x}),
\end{aligned} \tag{9}$$

where $d_X^a$ gathers only the $n_a$ components of $d_X$ than correspond to $X^a$ (the remaining $n_f$ components of $d_X^f$ corresponding to $X^f$ are always zero and do not affect the search for the minima).

The solution of (2) is given by those points of the form $\tilde{x} = (\hat{x}^f, \tilde{x}^a)$, and such solution can also be written as the Cartesian product $P_C(\hat{x}) = \hat{x}^f \times P_C(\hat{x}^a) = \{x \in X \mid x = (\hat{x}^f, x^a), \text{ with } x^a \in P_C(\hat{x}^a)\}$.

From a multicriteria optimization perspective, the selection of $X^a$ (and the corresponding set of distances) determines the existence of solutions and the size of the Pareto sets. A priori, it is known that, for a search in a fixed domain, the strict Pareto set associated with $n$ features contains the Pareto set associated with $n_a < n$ features (see Lemma 2.5 in [36]).

The following result shows that, even if the domain search is reduced from (2) to (9), due to the specific distance-based cost involved in both problems, the result remains true:

**Proposition 1** (From [7]) *Let $SP_C(\hat{x}) \subset X$ be the Strict Pareto set associated with* (2) *and let $SP_C(\hat{x}^a) \subset X^a$ be the Strict Pareto set associated with* (9)*. Then*

$$\hat{x}^f \times SP_C(\hat{x}^a) \subseteq SP_C(\hat{x}). \tag{10}$$

Proof: See Appendix B.1.

Applying the same reasoning recursively, given any two subsets of actionable variables that satisfy $a' \subset a$, the Pareto set corresponding to $X^{a'}$ is included in the Pareto set corresponding to $X^a$.

This approach enhances the simplicity of CEs, as including all potentially actionable variables ensures that the Pareto Set will determine the

minimum number of necessary variable modifications. The Pareto Set will inherently include solutions where some variables remain static and do not require changes, thus providing all possible solutions, including those with fewer modifications.

**Remark 2:** *Depending on the size of $X^a$ (number and range of its variables) and the restrictive nature of the classifier $C(x)$, some solutions to* (9) *may not exist or may not be easy to find.*

It is also important to note that if the main goal is to explain the model's decision without providing actionable solutions, it may not be necessary to restrict the search. For instance, in the case of a bank loan application, if we inform an applicant that she/he would have been approved if she/he was younger, we do not provide a means to obtain the loan but rather we would just explain the main factors influencing the decision, offering very valuable insights into the decision-making process.

## 4.5 Classifiers based on a threshold

The results discussed in this section are *Intrinsic* and therefore *Model specific*, since they depend on the internal workings and characteristics of the specific classifiers being analyzed.

Many binary classifiers designed according to machine learning paradigms can be formalized as the following composition of functions:

$$Y = C(x) = T(F(x)), \qquad T(z) = \begin{cases} 1, & \text{if } z \geq th, \\ -1, & \text{if } z < th, \end{cases} \tag{11}$$

where $th \in \mathbb{R}$ stands for a threshold value, and $F\colon X \to \mathbb{R}$ represents the transformation associated with all but the last classifier step (or layer), so that the value of $F(x) = z \in \mathbb{R}$ is also provided by the classifier (i.e., available to the user) for any given $x \in X$.

Assuming that $F$ is continuous, we can address (2) by solving the alternative problem:

$$\tilde{x} = \arg\min_{x \in X} d_X(\hat{x}, x), \text{ subject to } F(x) = th, \tag{12}$$

which would provide either a solution to (2) or (see reasoning above) points $x$ that may not satisfy $C(x) = -C(\hat{x})$ but are arbitrarily close to points that do satisfy it.

Three typical examples of machines that fit into this framework are Logistic Regressors (LR), Support Vector Classifiers (SVC) and Neural Networks (NN).

**Remark 3:** *Problem* (12) *can be (approximately) reformulated if we substitute the restriction in the search space by a penalty term $\lambda(F(x)-th)^2$, with $\lambda \gg 1$, to be added to the cost function. In the multi-objective formulation proposed in [17], this term was included as an additional index.*

### 4.5.1 Case of affine $F(x)$

For the case in which all variables are numeric and $X$ is a convex subset of $\mathbb{R}^n$, if we consider a LR and SVC with linear kernel we have that the condition $F(x) = th$ defines a hyperplane in the space of inputs which allows for a straightforward characterization of the Pareto set associated with a given input $\hat{x}$. For instance, in the case of LR we have that

$$F(x) = g(x;\theta) = \frac{1}{1 + e^{\theta'_0 + \theta_1 x_1 + \cdots + \theta_n x_n}}, \tag{13}$$

and the condition $F(x) = th$ can be written as $\theta'_0 + \theta_1 x_1 + \cdots + \theta_n x_n = \ln\left(\frac{1-th}{th}\right)$ which, denoting $\theta_0 = \theta'_0 - \ln\left(\frac{1-th}{th}\right)$, becomes $\theta_0 + \theta_1 x_1 + \cdots + \theta_n x_n = \theta_0 + \boldsymbol{\theta} \cdot x = 0$ (where $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_n)$).

Similarly, for SVC, the restriction $F(x) = th$ defines also a hyperplane $(H)$ so that for both LR and SVC (12) can be formulated as

$$\tilde{x} = \arg\min_{x \in X} d_M(\hat{x}, x), \text{ subject to } \theta_0 + \boldsymbol{\theta} \cdot x = 0. \tag{14}$$

Given $\hat{x}$ and assuming that $\theta_i \neq 0$, we can define:

$$\bar{x}_i = \frac{\theta_i \hat{x}_i - \boldsymbol{\theta} \cdot \hat{x} - \theta_0}{\theta_i}. \tag{15}$$

Note that if $\hat{x} \notin H$ then $|\bar{x}_i - \hat{x}_i| > 0$ and the point

$$\check{x}_i = (\hat{x}_1, \ldots, \bar{x}_i, \ldots, \hat{x}_n) \tag{16}$$

is in the hyperplane $H$, satisfying that it is the closest to $\hat{x}$ along the $i$-th axis. As a matter of fact, all points of $H$ can be written as

$$H = \left\{ x \in \mathbb{R}^n \middle| x = \lambda_1 \check{x}_1 + \cdots + \lambda_n \check{x}_n, \ \sum_{i=1}^{n} \lambda_i = 1 \right\}. \tag{17}$$

If we work with the distances $d_{X_i}(\hat{x}_i, x_i) = |\hat{x}_i - x_i|, \ i = 1, \ldots, n$, it can be shown that for any $x \in H$ the vector of distances (5) takes the form

$$d_M(\hat{x}, x) = \big(|\lambda_1| \cdot |\bar{x}_1 - \hat{x}_1|, \ldots, |\lambda_n| \cdot |\bar{x}_n - \hat{x}_n|\big). \tag{18}$$

We can then state the following result:

**Proposition 2** (From [7]) *The set of Pareto Efficient points of* (14)-(18) *is given by*

$$P = \left\{ x \in \mathbb{R}^n \middle| x = \lambda_1 \check{x}_1 + \cdots + \lambda_n \check{x}_n, \lambda_i \geq 0, \ \sum_{i=1}^{n} \lambda_i = 1 \right\}, \tag{19}$$

*meaning that such Pareto set is the Convex Hull (CH) of the set of points* $\check{x}_1, \ldots, \check{x}_n$ *defined in* (15)-(16).

Proof: See Appendix B.2.

## 4.6 Extension of the framework to Regression Problems

The above proposed framework for classifiers can also be extended to regression problems. In this case, the objective is to find the closest input value $\tilde{x} \in X$ to $\hat{x}$ such that the predicted response variable $Y = F(x)$ meets a desired condition specified by the user. This condition can be a specific target value $y^*$ or an interval of acceptable values $[y_{\min}, y_{\max}]$. For example, given a reference input value $\hat{x}$, we can formalize this as:

$$\tilde{x} = \arg\min_{x \in X} d_X(\hat{x}, x), \quad \text{subject to} \quad F(x) = y^*, \tag{20}$$

or, in the case of an interval:

$$\tilde{x} = \arg\min_{x \in X} d_X(\hat{x}, x), \quad \text{subject to} \quad y_{\min} \leq F(x) \leq y_{\max}. \tag{21}$$

Therefore, by expanding the possible ways to specify target values or ranges in continuous response settings, the CE approach can be adapted to a wider range of applications.

In the following section, we address the problem when the classifier is not accessible but only a set of classification examples is available.

# 5 CE when only samples are available

In the previous sections, it was assumed that a well-tested classifier, capable of making accurate predictions, was provided. However, a different but also relevant scenario for CE shows up when the classifier is not accessible a priori, but only a set of samples $D = (x_i, C(x_i)),\ i = 1, \dots, N_D$, is available. This framework is very natural in the context of supervised machine learning.

In this case, two objectives can be pursued:

1. To provide, given $\hat{x} \in D$, a CE for $C(\hat{x})$.
2. To provide, given $\hat{x} \notin D$, a prediction of $C(\hat{x})$ and a CE for such prediction.

Both problems require constructing some classifier that approximates $C$. However, since different machine learning paradigms can be employed, several approximation models may be available and it is often not straightforward to choose which classifier best approximates $C$, as these models might yield similar results in the measured testing metrics (e.g., Accuracy, F1-score, Recall). Additionally, sometimes it is unclear which metric is most important for evaluating our model.

This scenario can benefit from multi-objective optimization techniques, as discussed in the work of Zufiria [12]. Although we will not delve into multi-objective machine training here, it is important to acknowledge its relevance and potential applications which also benefit from the multi-objective framework.

For ease of exposure, we will first address the computation of the Pareto set for any $\hat{x} \in X$; this computation will only make sense when all approximating classifiers provide the same output for $\hat{x}$. These results will directly apply for the CE associated with any $\hat{x} \notin D$. Afterwards, in Section 5.3 we will address the CE associated with any $\hat{x} \in D$ by considering the implications of knowing the value $C(\hat{x})$.

Let us denote by $\mathcal{C} = \{C_1, \dots, C_K\}$ the set of employed classifiers. Note that for each point $\hat{x}$ and classifier $C_i \in \mathcal{C}$, the whole procedure shown in the previous Section 4.1 can be applied for obtaining the corresponding Pareto set $P_{C_i}(\hat{x})$, an approximation of $P_C(\hat{x})$. More generally, for any subset of machines $\mathcal{C}' \subseteq \mathcal{C}$ whose elements satisfy that $C_i(\hat{x}) = C_j(\hat{x}),\ \forall C_i, C_j \in \mathcal{C}'$, we can also obtain $P_{\mathcal{C}'}(\hat{x})$, the solution of

$$\begin{aligned} \tilde{x} &= \arg\min_{x \in X} d_X(\hat{x}, x), \\ &\text{subject to } C_i(x) = -C_i(\hat{x}),\ \forall C_i \in \mathcal{C}'. \end{aligned} \tag{22}$$

The construction of different sets $P_{\mathcal{C}'}(\hat{x})$ may serve two different purposes:

1. To obtain different (hopefully improved) approximations of $P_C(\hat{x})$ by appropriately merging the results provided by different subsets $\mathcal{C}'$.

2. To comparatively analyze the performance of the classifiers. The comparative quality of the sets $P_{C_i}(\hat{x})$ may shed light on the performance of the $C_i$ classifiers.

**Remark 4:** *Even though, a priori, the comparison among classifiers may not be the main objective, it is a (previous) byproduct to be collected in order to improve the quality of the obtained approximations of the CE Pareto set* $P_C(\hat{x})$.

For ease of exposure, let us consider first the comparison among the classifiers.

## 5.1 Comparative performance analysis

The different $P_{C_i}(\hat{x})$ sets for $C_i \in \mathcal{C}$ can be processed as follows. One can define $[P_{C_i}(\hat{x})]_{C_j} = \{x \in P_{C_i}(\hat{x}) \mid C_j(x) = C_i(x)\} \subseteq P_{C_i}(\hat{x})$, for each $C_j \in \mathcal{C}$, so that we *sieve* $P_{C_i}(\hat{x})$ by selecting from it those points similarly classified by classifier $C_j$, provided $C_i(\hat{x}) = C_j(\hat{x})$. (Note that, obviously, $[P_{C_i}(\hat{x})]_{C_i} = P_{C_i}(\hat{x})$.)

A cross analysis of the relative cardinality of sets $P_{C_i}(\hat{x})$ versus $[P_{C_i}(\hat{x})]_{C_j}$ with $C_i, C_j \in \mathcal{C}$ can provide significant information about the reliability of each classifier $C_i$

## 5.2 Different approximations of the Pareto set $P_C(\hat{x})$

In order to partially relate the different $P_{\mathcal{C}'}(\hat{x})$ sets (depending on $\mathcal{C}'$) that can be constructed to approximate $P_C(\hat{x})$, we present the following result:

**Proposition 3** (From [7]) *Let* $C_i, C_j \in \mathcal{C}$, *then*

$$[P_{C_i}(\hat{x})]_{C_j} \subset P_{\{C_i,C_j\}}(\hat{x}). \tag{23}$$

Proof: See Appendix B.3.

Although there are many ways to merge the resulting $P_{\mathcal{C}'}$, in connection with the previous Section 5.1, we will only consider the cases in which $\mathcal{C}' = \{C_i\}$ for $i = 1, \ldots, K$. Unless otherwise stated, we shall consider that all classifiers $C_i,\ i = 1, \ldots, K$ whose associated $P_{C_i}(\hat{x})$ are to be jointly processed, satisfy the condition $C_i(\hat{x}) = C_j(\hat{x}),\ i, j = 1, \ldots, K$. Hence, let us consider that after computing each $P_{C_i}(\hat{x})$ we construct a fully sieved version of $P_{C_i}(\hat{x})$ (through the rest of the classifiers):

$$P^S_{C_i}(\hat{x}) = [P_{C_i}(\hat{x})]_{C_1,\ldots,C_K} = \{x \in P_{C_i}(\hat{x}) \mid C_1(x) = \cdots = C_K(x)\} \tag{24}$$

Figure 5 illustrates the computation of $P^S_{C_1}(\hat{x})$ for $K = 4$:

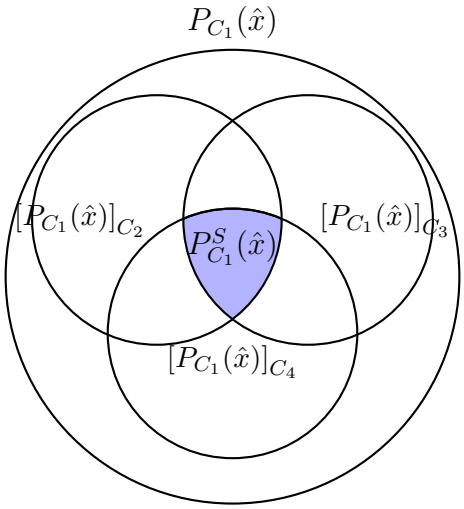


Figure 5: Selection of $P^S_{C_1}(\hat{x})$, fully sieved $P_{C_1}(\hat{x})$.

Hereafter, we can consider the whole set of points that belong to any of the fully sieved Pareto sets, and select the Pareto efficient elements among them:

$$P^S_{\mathcal{C}}(\hat{x}) = \bigcup_{C_i \in \mathcal{C}} P^S_{C_i}(\hat{x}). \tag{25}$$

Alternatively, one can also compute $P_{\mathcal{C}}(\hat{x})$ by considering all the classifier restrictions at once. When carrying out this computation, the comparative performance of the different $C_i$ is not explicitly disclosed; in return, besides obtaining all points of $P^S_{\mathcal{C}}(\hat{x})$ we might get some additional PE point since $P^S_{\mathcal{C}}(\hat{x}) \subset P_{\mathcal{C}}(\hat{x})$, which can be proved via a recursive application of Proposition 3.

We come back now to considering the implications of having available the value $C(\hat{x})$.

## 5.3 CE for $\hat{x} \in D$

When $C(\hat{x})$ is available, we can check if each $C_i \in \mathcal{C}$ satisfies $C_i(\hat{x}) = C(\hat{x})$, (i.e., if $\hat{x}$ is correctly classified by such approximating classifier).

In case that $C_i(\hat{x}) = C(\hat{x})$ the whole procedure shown in the previous Section 4.1 can be applied to $C_i$ for obtaining the corresponding Pareto set $P_{C_i}(\hat{x})$, an approximation of $P_C(\hat{x})$.

In case that $C_i(\hat{x}) = -C(\hat{x})$ it may be natural to discard such classifier in the Pareto set construction procedure. Obviously, this result provides information about the quality of $C_i$; furthermore, we can compute the Pareto

set associated with

$$\tilde{x} = \arg\min_{x \in X} d_X(\hat{x}, x), \text{ subject to } C_i(x) = -C_i(\hat{x}), \tag{26}$$

which gives a CE of the misclassification carried out by $C_i$.

**Remark 5:** *In order to simplify the notation we have considered the standard search in $X$. Nevertheless, depending on the characteristics or requirements of $\hat{x}$, one must determine the desired decomposition $\hat{x} = (\hat{x}^f, \hat{x}^a)$, where the variables included in $X^a$ may condition the existence of solutions (see Proposition 1 and Remark 2). The notation of the corresponding counterfactual Pareto sets can be defined accordingly so that $P_{\mathcal{C}'}(\hat{x}) = \{\hat{x}^f\} \times P_{\mathcal{C}'}(\hat{x}^a)$.*

# 6 Proposed Algorithm

We consider the case in which only the dataset $D = (x_i, C(x_i)),\ i = 1, \dots N_D$, is available. The set $D$ is partitioned into training and test sets, so that the classifiers $C_i$ are constructed using the training set. Although the CE could be addressed for any $\hat{x} \in X$, in this work we are going to pay more attention to those values $\hat{x} \in D$ in the test set that satisfy $C_i(\hat{x}) = C(\hat{x})$ for each $C_i \in \mathcal{C}$ (i.e., which are correctly classified by all the classifiers).

For each $C_i \in \mathcal{C}$, we compute $\tilde{P}_{C_i}(\hat{x})$, an approximation of $P_{C_i}(\hat{x})$, the set of solutions of (9), using the Genetic Algorithm NSGA-III, which is specially suited for multi-objective optimization [37]. The implementation uses the *pymoo* library, available at `https://pymoo.org/algorithms/moo/nsga3.html`.

**Remark 6:** *In practice only $\tilde{P}_{C_i}(\hat{x})$, an approximation of $P_{C_i}(\hat{x})$, can be obtained. Note that if $C_i$ is a threshold-based classifier with affine $F(x)$, all the points of the convex hull $CH(\tilde{P}_{C_i}(\hat{x}))$ will satisfy the condition $C_i(CH(\tilde{P}_{C_i}(\hat{x}))) = -C_i(\hat{x})$ but not all of them need to be Pareto efficient.*

The corresponding approximations of both $P^S_{\mathcal{C}}(\hat{x})$ (see (25)) and $P_{\mathcal{C}}(\hat{x})$ are also computed using NSGA-III.

## 6.1 Distance Metrics

In our multi-objective optimization problem, different distance metrics are employed for continuous and categorical variables to minimize the distances between the original and counterfactual instances.

For continuous variables, we use the absolute difference:

$$d_{X_i}(\hat{x}_i, x_i) = |\hat{x}_i - x_i|.$$

For categorical variables, the distance metric depends on the type of categorical data:

1. **Nominal Variables:** For nominal variables, we use the Hamming distance:

$$d_{X_i}(\hat{x}_i, x_i) = \begin{cases} 0 & \text{if } \hat{x}_i = x_i, \\ 1 & \text{if } \hat{x}_i \neq x_i. \end{cases}$$

2. **Ordinal Variables:** For ordinal variables, we use a distance metric that takes into account the order of the categories. One suitable metric is the normalized absolute difference:

$$d_{X_i}(\hat{x}_i, x_i) = |\text{rank}(\hat{x}_i) - \text{rank}(x_i)|.$$

3. **User-defined Distances:** For some variables, it may be necessary to use a customized distance metric provided by the user. This is represented as a symmetric distance matrix $D$, where $D_{ij}$ is the distance between categories $i$ and $j$. For a given categorical variable $X_i$, the distance is:

$$d_{X_i}(\hat{x}_i, x_i) = D_{\hat{x}_i, x_i}$$

This allows for more flexibility in defining the dissimilarity between different categories based on domain-specific knowledge.

## 6.2 Genetic Algorithm NSGA-III

Genetic algorithms are optimization techniques inspired by natural selection. They operate on a population of potential solutions, applying operators like selection, crossover, and mutation to evolve solutions towards better performance. Key concepts include the fitness function, which evaluates how well a solution meets the desired objective, and the use of genetic operators to explore the search space and introduce variability, ultimately converging towards optimal or near-optimal solutions.

The Genetic Algorithm NSGA-III is employed for our optimization task due to its numerous advantages. It builds upon NSGA-II by efficiently managing a large number of objectives using reference points, ensuring a diverse set of solutions. Its genetic nature allows it to explore a wide search space efficiently, providing robust solutions in complex spaces. Additionally, NSGA-III handles both categorical and continuous variables, making it versatile for various datasets.

NSGA-III can incorporate diverse constraints to maintain realistic causal relationships between features. For instance, when modifying educational attainment, it is essential to account for the corresponding increase in age. The flexibility of NSGA-III allows us to encode such relationships into the optimization problem, ensuring more realistic and *Causality*-preserving counterfactuals. Moreover, NSGA-III enables the determination of the range of movement for variables, facilitating actionable counterfactuals.

Genetic algorithms perform well when the prediction functions of the models are quick to evaluate. They can also be parallelized to accelerate the search for solutions, which aligns with the property of *Amortized Inference* defined in Section 3.2. However, if the prediction functions are computationally expensive, other optimization algorithms such as surrogate-based optimization or Bayesian optimization[38, 12] might be more appropriate.

A typical multi-objective optimization problem that NSGA-III can solve

is formulated as follows:

$$\begin{aligned} \text{Minimize} \quad & (f_1(x), f_2(x), \ldots, f_m(x)) \\ \text{subject to} \quad & g_j(x) \geq 0, \quad j = 1, 2, \ldots, J, \\ & h_k(x) = 0, \quad k = 1, 2, \ldots, K, \\ & x_i^{(L)} \leq x_i \leq x_i^{(U)}, \quad i = 1, 2, \ldots, n. \end{aligned} \tag{27}$$

In scenarios where the Pareto set has an infinite cardinality (note that the Convex Hull of a set containing at least two different points has an infinite cardinality), such as in the examples discussed in 4.5.1, caution is necessary when analyzing the results of the NSGA-III algorithm, which approximates the Pareto set with a finite number of points. Comparisons of the cardinality of these approximations should consider the resolution of the approximation.

For instance, the NSGA-III may produce more points for four variables than for three, which makes sense if the resolutions are similar. The number of points will depend on the population size and the number of iterations in the approximation. It is important to point out that the shape of the Pareto set has been usually outlined without needing to use very large population sets.

# 7 Examples of Application

In this section, we illustrate the practical application of our methodology through two examples. These examples showcase the process of model training, evaluation, and the computation of Pareto sets for binary classification problems. The first example involves predicting heart failure events, while the second example pertains to bank loan approvals.

## 7.1 Example 1: Heart Failure

We have considered a binary classification problem based on the dataset *Heart Failure Prediction*, which provides 11 clinical features for 918 patients aimed at predicting death events (see `https://www.kaggle.com/code/azizozmen/heart-failure-predict-8-classification-techniques/input`).

It has been indicated by several users of the dataset that some of the records present a cholesterol value of 0, which clearly does not correspond to a true value. Hence, those items were removed from the dataset so that finally 746 records (with valid cholesterol values) were employed for our analysis.

We have considered the following four types of binary classifiers: a Logistic Regressor (LR), a Support Vector Classifier (SVC) with a linear kernel, a Random Forest (RF), and a Neural Network (NN). Hence, we denote $\mathcal{C} = C_1$=LR, $C_2$=SVC, $C_3$=RF, $C_4$=NN.

### 7.1.1 Model Training and Evaluation

The dataset was split in a stratified manner into 85% for training (634 indivisuals) and 15% for testing (112 individuals). For the RF and SVC linear models, we performed 5-Fold cross-validation on the training set to select the best hyperparameters that achieved the highest accuracy. In Logistic Regression, L2 regularization was used to prevent overfitting, utilizing the lbfgs solver. These models were trained using scikit-learn library.

For the NN, we used 2 intermediate layers, one with 128 neurons and another with 64, employing the ReLU activation function, Adam optimizer, and binary cross-entropy loss function. We reserved a validation set to determine the number of training epochs, also based on accuracy. This model was trained using tensorFlow library.

Categorical variables were one-hot encoded, and numerical variables were scaled in our models. This preprocessing step does not affect our framework since it is considered within the evaluation function of the model (as part of a pipeline).

In all the classifiers, the resulting values for Accuracy when classifying such data varied between 0.84 and 0.87 (the NN presenting a mildly better performance than the others). Below are the confusion matrices (Figure 6) and metrics (Table 1) obtained for each of the four models in the test set.

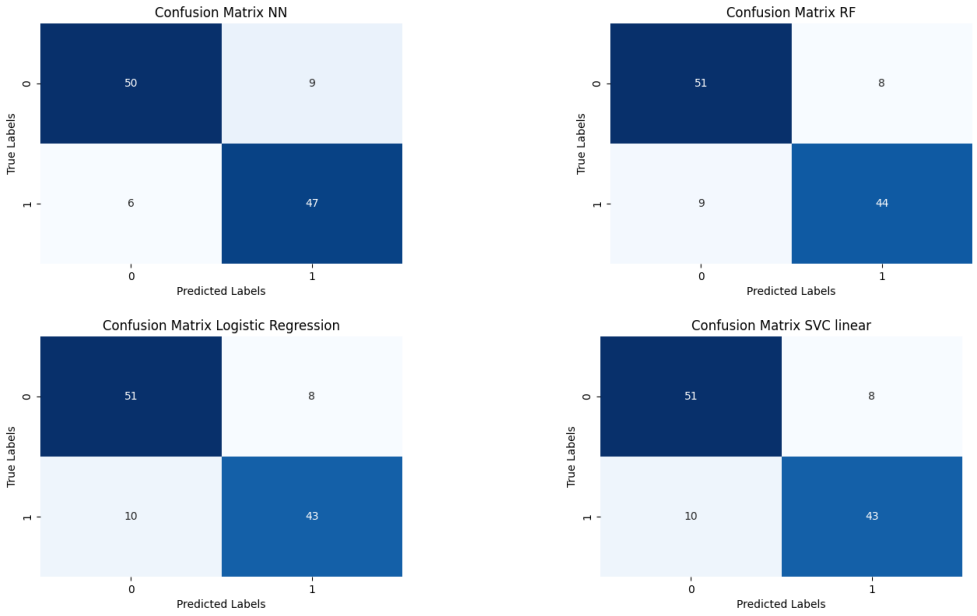


Figure 6: Confusion Matrices for the four models

| Model | Accuracy | Precision | Recall |
|---|---|---|---|
| Logistic Regression | 0.84 | 0.84 | 0.81 |
| SVC | 0.84 | 0.84 | 0.81 |
| Random Forest | 0.85 | 0.85 | 0.83 |
| Neural Network | 0.87 | 0.89 | 0.89 |

Table 1: Summary of Metrics for Each Model

Additionally, the Hamming distances between the predictions of the four models show that they have similar behavior, as represented in the Figure 7.

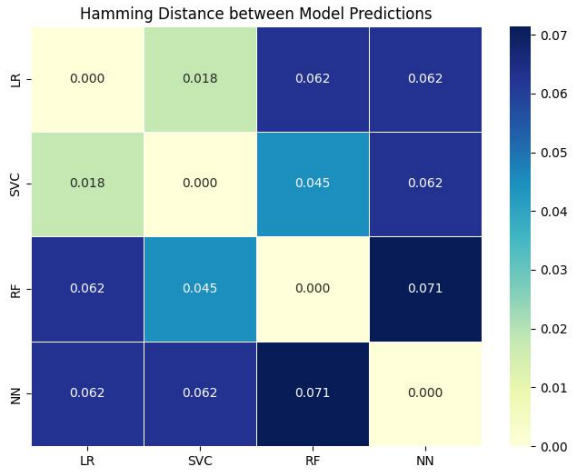


Figure 7: Hamming Distance between Model Predictions

**Remark 8:** *In this context, the Hamming distance between the predictions of the classifiers refers to the number of times their classifications do not match among the total number of individuals. Formally, for two classifiers $C_i$ and $C_j$, the Hamming distance $d_H$ between their predictions on a set of individuals $\{x_k\}_{k=1}^n$ is defined as:*

$$d_H(C_i, C_j) = \frac{1}{n} \sum_{k=1}^{n} \mathbb{I}\{C_i(x_k) \neq C_j(x_k)\},$$

*where $\mathbb{I}\{\cdot\}$ is the logical indicator function.*

Among the 112 individuals of the testing set, 42 of them provide a correct output 1 (heart disease) in the four classifiers (i.e., they satisfy $C_i(\hat{x}) = C(\hat{x}) = 1,\ \forall i = 1, 2, 3, 4$). Additionally, there are 4 individuals that have been misclassified as positive by all classifiers (i.e., $C_i(\hat{x}) = 1 \neq C(\hat{x}) = -1,\ \forall i = 1, 2, 3, 4$). Furthermore, there are 3 individuals for whom at least 2 classifiers satisfy $C_i(\hat{x}) = C(\hat{x}) = 1$. The label numbers associated with those input values are provided in the first column of Table 2, 3 and 4 respectively.

### 7.1.2 Computation of Pareto Sets

In this first application, only continuous variables were modified to ensure a better representation of the solutions. Pareto solutions from different machines were then compared and combined as explained in Section 5.

In the following, we illustrate, for some of these points, the computation of the corresponding $\tilde{PC}_i(\hat{x}^a)$ (approximations of the respective $PC_i(\hat{x}^a)$) for the cases when $X^a$ has 3 and 4 actionable variables. The search for solutions has been restricted to the range of the variables in the dataset, with a small margin, ensuring that the values are achievable by an individual.

For the genetic algorithm, the Das-Dennis method with 12 partitions was used (see `https://pymoo.org/misc/reference_directions.html`). For 3 actionable variables, a population of 150 with 1500 iterations was used, and for 4 actionable variables, a population of 450 with 2000 iterations was employed.

**$\tilde{PC}_i(\hat{x}^a)$ for $X^a$ with 3 actionable variables:**

First, we consider modifying the following set of 3 actionable variables:

1. Resting blood pressure (BP).
2. Cholesterol (C).
3. Maximum heart rate achieved (MaxHR).

These variables are considered modifiable in humans because they can be influenced through lifestyle changes, medication, and other medical interventions. For instance, resting blood pressure can be managed with antihypertensive drugs, diet, and exercise. Cholesterol levels can be altered through dietary changes. Maximum heart rate achieved can be improved with regular physical activity.

By searching in that space, we have only found two individuals $\hat{x}_2$ and $\hat{x}_{68}$ for which the corresponding $\tilde{PC}_i(\hat{x}^a)$ was found for all $C_i$, $i = 1, 2, 3, 4$, as illustrated in Table 2. Such individuals are located close to the boundary of the decision region. Note that the NSGA-III algorithm is able to find many more Pareto sets for the RF classifier, which is the only one that is not based in a threshold as described in (11).

| Label for $\hat{x}$ | LR | SVC | RF | NN |
|---|---|---|---|---|
| 2 | ✓ | ✓ | ✓ | ✓ |
| 7 | ✗ | ✗ | ✓ | ✗ |
| 10 | ✗ | ✗ | ✓ | ✗ |
| 11 | ✗ | ✗ | ✗ | ✗ |
| 15 | ✗ | ✗ | ✗ | ✗ |
| 16 | ✗ | ✗ | ✗ | ✗ |
| 17 | ✗ | ✗ | ✗ | ✗ |
| 19 | ✗ | ✗ | ✓ | ✗ |
| 22 | ✗ | ✗ | ✓ | ✗ |
| 24 | ✗ | ✗ | ✗ | ✗ |
| 25 | ✗ | ✗ | ✗ | ✗ |
| 27 | ✗ | ✗ | ✗ | ✗ |
| 32 | ✗ | ✗ | ✗ | ✗ |
| 33 | ✗ | ✗ | ✗ | ✗ |
| 45 | ✗ | ✗ | ✓ | ✗ |
| 46 | ✗ | ✗ | ✗ | ✗ |
| 51 | ✗ | ✗ | ✗ | ✗ |
| 53 | ✗ | ✗ | ✗ | ✗ |
| 58 | ✗ | ✗ | ✗ | ✗ |
| 60 | ✗ | ✗ | ✓ | ✗ |
| 62 | ✗ | ✗ | ✗ | ✗ |
| 67 | ✗ | ✗ | ✗ | ✗ |
| 68 | ✓ | ✓ | ✓ | ✓ |
| 74 | ✗ | ✗ | ✓ | ✗ |
| 75 | ✗ | ✗ | ✗ | ✗ |
| 76 | ✗ | ✗ | ✓ | ✗ |
| 77 | ✗ | ✗ | ✓ | ✗ |
| 80 | ✗ | ✗ | ✗ | ✗ |
| 83 | ✗ | ✗ | ✗ | ✗ |
| 86 | ✗ | ✗ | ✓ | ✗ |
| 89 | ✗ | ✗ | ✓ | ✗ |
| 91 | ✓ | ✓ | ✗ | ✗ |
| 97 | ✗ | ✗ | ✓ | ✗ |
| 99 | ✗ | ✗ | ✗ | ✗ |
| 100 | ✗ | ✗ | ✗ | ✗ |
| 101 | ✗ | ✗ | ✗ | ✗ |
| 102 | ✗ | ✗ | ✓ | ✗ |
| 103 | ✗ | ✗ | ✓ | ✗ |
| 104 | ✗ | ✗ | ✓ | ✗ |
| 108 | ✗ | ✗ | ✗ | ✗ |
| 109 | ✗ | ✗ | ✗ | ✗ |
| 111 | ✓ | ✗ | ✓ | ✗ |

Table 2: Results of the search of Pareto sets for the 42 correctly classified input vectors $\hat{x}$, in the four types of classifiers (LR, SVC, RF, NN), by modifying the three actionable variables BP, C and maxHR. Numbers in the first column represent the labels for the $\hat{x}$ among the total of 112. Green: the Pareto set was located for point (number) in classifier (column); red: the Pareto set was not located for point (number) in classifier (column).

Table 3 displays those input values that were equally classified by the 4 classifiers with 1 (heart disease), but for which the classification is wrong (opposed to the labeled output). Note that only RF was able to locate a Pareto set for them.

| Label for $\hat{x}$ | LR | SVC | RF | NN |
|---|---|---|---|---|
| 12 | ✗ | ✗ | ✓ | ✗ |
| 54 | ✗ | ✗ | ✓ | ✗ |
| 71 | ✗ | ✗ | ✓ | ✗ |
| 110 | ✗ | ✗ | ✓ | ✗ |

Table 3: Results of the search of Pareto sets for the 4 input vectors $\hat{x}$ equally classified by the four classifiers, but for which the classification is wrong. See caption of Table 2 for explanation of rows, columns and colors.

Table 4 shows the search for Pareto sets for the 3 individuals who were correctly classified as positive by at least 2 classifiers. It is observed that solutions are often found for these individuals as they are near the decision boundary.

| Label for $\hat{x}$ | LR | SVC | RF | NN |
|---|---|---|---|---|
| 55 | ✓ | NP | NP | ✗ |
| 78 | NP | ✓ | ✓ | ✓ |
| 79 | NP | NP | ✓ | ✓ |

Table 4: Search for Pareto sets for the 3 correctly classified individuals by at least 2 classifiers (LR, SVC, RF, NN). NP indicates Negative Prediction.

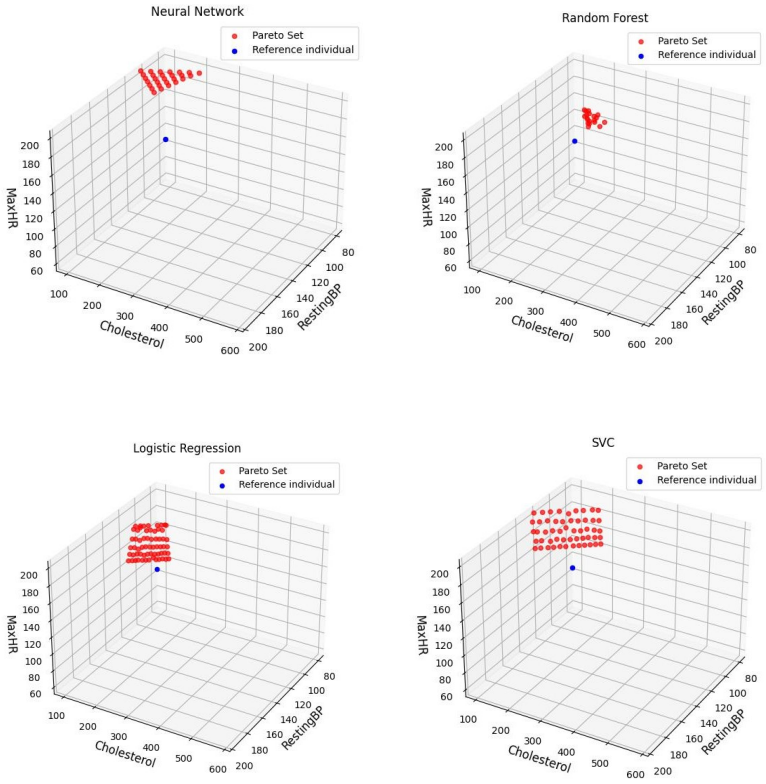


Figure 8: For point 2, the reference input value $\hat{x}_2^a$ (blue), and the associated approximated CE Pareto sets $\tilde{P}_{C_i}(\hat{x}_2^a)$ (red) for the four classifiers.

Figure 8 and 9 displays, for the points labeled 2 and 68, the value of $\hat{x}^a$ and the corresponding $\tilde{P}_{C_i}(\hat{x}^a)$ for LR, SVC, RF and NN. Even though the sets cover similar regions, some significant differences show up. Classifiers LR, SVC and NN, which follow 11, provide regularly structured approximations of the Pareto sets; in particular, LR and SVC, being of the form analyzed in Section 4.5.1, provide very similar Pareto sets whose structure corresponds to the result provided in Proposition 2. The approximation for RF shows a more irregular structure. Notably, the NN model appears to be the most restrictive, as the counterfactual solutions in the Pareto sets are the furthest from the reference individual.

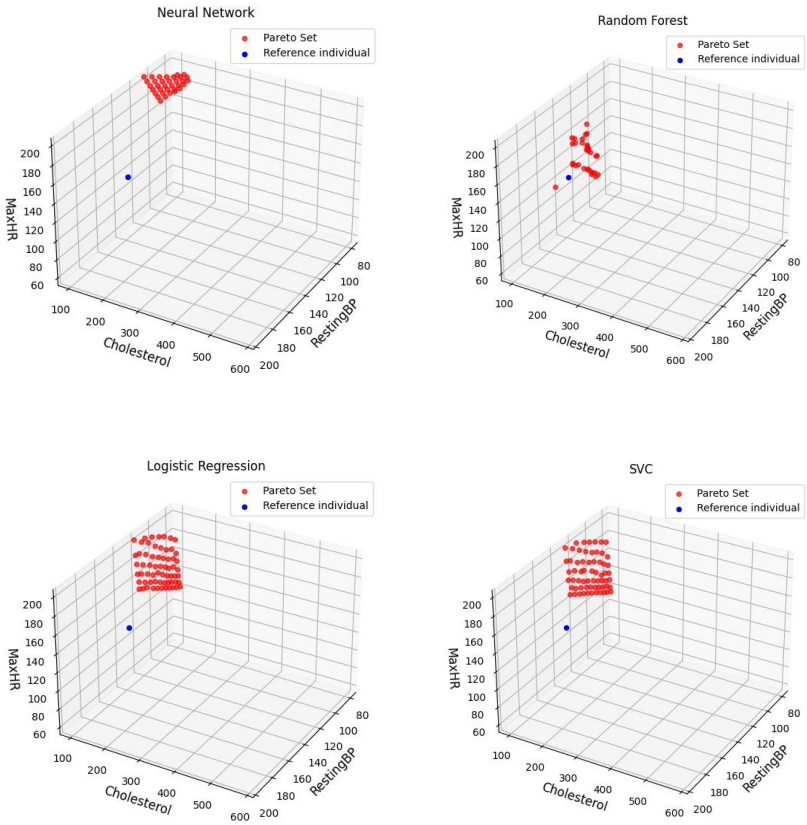


Figure 9: For point 68, the reference input value $\hat{x}^a_{68}$ (blue), and the associated approximated CE Pareto sets $\tilde{P}_{C_i}(\hat{x}^a_{68})$ (red).

Table 5 shows $|[\tilde{P}_{C_i}(\hat{x}^a)]_{C_j}|$, the size of the corresponding sieved approximated Pareto sets for points 2 and 68, where $C_i$ is given by the row classifier and $C_j$ by the column classifier. Note that for both points $\tilde{P}_{\text{NN}}(\hat{x}^a)$ preserves most of its elements when sieved by the other classifiers.

| $C_i$ \ $C_j$ | LR | SVC | RF | NN |
|---|---|---|---|---|
| LR | 62 | 0 | 0 | 0 |
| SVC | 53 | 53 | 11 | 0 |
| RF | 9 | 0 | 19 | 0 |
| NN | 28 | 28 | 25 | 28 |

| $C_i$ \ $C_j$ | LR | SVC | RF | NN |
|---|---|---|---|---|
| LR | 63 | 63 | 46 | 0 |
| SVC | 0 | 63 | 40 | 0 |
| RF | 0 | 0 | 32 | 0 |
| NN | 34 | 34 | 34 | 34 |

Table 5: Case of 3 actionable variables. Sizes of the corresponding sieved Pareto sets $[\tilde{P}_{C_i}(\hat{x})]_{C_j}$ for individuals 2 (up) and 68 (down).

**$\tilde{P}_{C_i}(\hat{x}^a)$ for $X^a$ with 4 actionable variables:**

In this section we address the incorporation of a fourth actionable variable:

4. Old Peak depression in mV (OP).

Old Peak depression, which measures the change in the ST segment of an ECG under stress, can be modified through medical interventions such as medication or lifestyle changes.

The approximated Pareto set has been found for four individuals: 2, 68, 91 and 111. The increase in the number of points for which a solution was found is in line with Proposition 1. The results can be seen in Table 6, where it is noteworthy that the Random Forest finds solutions for almost all individuals. The analogous results for Tables 3 and 4 corresponding to the 4 actionable variables have been omitted due to no significant changes.

Table 7 shows the located $[\tilde{P}_{C_i}(\hat{x})]_{C_j}$ sets for the same points (2 and 68) for $X^a$ having now 4 variables. From the perspective of the sieving results, the LR and the SVC classifiers seem to be the less restrictive ones for point 2 and 68 respectively; on the other hand, the NN classifier seems to be the more restrictive for both points: all points of the sets located by the other architectures are not correctly classified by the NN, whereas almost all the points located by the NN are correctly classified by the other classifiers.

| Label for $\hat{x}$ | LR | SVC | RF | NN |
|---|---|---|---|---|
| 2 | ✓ | ✓ | ✓ | ✓ |
| 7 | ✗ | ✗ | ✓ | ✗ |
| 10 | ✗ | ✗ | ✓ | ✗ |
| 11 | ✓ | ✓ | ✓ | ✗ |
| 15 | ✗ | ✗ | ✓ | ✗ |
| 16 | ✗ | ✗ | ✓ | ✗ |
| 17 | ✗ | ✗ | ✓ | ✗ |
| 19 | ✗ | ✗ | ✓ | ✗ |
| 22 | ✗ | ✗ | ✓ | ✗ |
| 24 | ✗ | ✗ | ✗ | ✗ |
| 25 | ✗ | ✗ | ✓ | ✗ |
| 27 | ✗ | ✗ | ✓ | ✗ |
| 32 | ✗ | ✗ | ✓ | ✗ |
| 33 | ✗ | ✗ | ✓ | ✗ |
| 45 | ✗ | ✗ | ✓ | ✗ |
| 46 | ✗ | ✗ | ✓ | ✗ |
| 51 | ✗ | ✗ | ✓ | ✗ |
| 53 | ✗ | ✗ | ✓ | ✗ |
| 58 | ✗ | ✗ | ✗ | ✗ |
| 60 | ✗ | ✗ | ✓ | ✗ |
| 62 | ✗ | ✗ | ✓ | ✗ |
| 67 | ✗ | ✗ | ✓ | ✗ |
| 68 | ✓ | ✓ | ✓ | ✓ |
| 74 | ✗ | ✗ | ✓ | ✗ |
| 75 | ✗ | ✗ | ✓ | ✗ |
| 76 | ✗ | ✗ | ✓ | ✗ |
| 77 | ✗ | ✗ | ✓ | ✗ |
| 80 | ✗ | ✗ | ✓ | ✗ |
| 83 | ✗ | ✗ | ✓ | ✗ |
| 86 | ✗ | ✗ | ✓ | ✗ |
| 89 | ✗ | ✗ | ✓ | ✗ |
| 91 | ✓ | ✓ | ✓ | ✓ |
| 97 | ✗ | ✗ | ✓ | ✗ |
| 99 | ✗ | ✗ | ✓ | ✗ |
| 100 | ✗ | ✗ | ✓ | ✗ |
| 101 | ✗ | ✗ | ✓ | ✗ |
| 102 | ✗ | ✗ | ✓ | ✗ |
| 103 | ✗ | ✗ | ✓ | ✗ |
| 104 | ✗ | ✗ | ✓ | ✗ |
| 108 | ✗ | ✗ | ✗ | ✗ |
| 109 | ✗ | ✗ | ✗ | ✗ |
| 111 | ✓ | ✓ | ✓ | ✓ |

Table 6: Results of the search of Pareto sets for the 42 correctly classified input vectors $\hat{x}$, in the four types of classifiers (LR, SVC, RF, NN), by modifying the four actionable variables BP, C, maxHR and OP.

| $C_i$ \ $C_j$ | LR | SVC | RF | NN |
|---|---|---|---|---|
| LR | 105 | 0 | 1 | 0 |
| SVC | 71 | 71 | 14 | 0 |
| RF | 30 | 0 | 169 | 0 |
| NN | 30 | 30 | 27 | 30 |

| $C_i$ \ $C_j$ | LR | SVC | RF | NN |
|---|---|---|---|---|
| LR | 222 | 222 | 92 | 0 |
| SVC | 1 | 298 | 96 | 0 |
| RF | 55 | 70 | 155 | 0 |
| NN | 169 | 169 | 145 | 169 |

Table 7: Case of 4 actionable variables. Sizes of the corresponding sieved Pareto sets $[\tilde{P}_{C_i}(\hat{x})]_{C_j}$ for individuals 2 (up) and 68 (down).

Table 8 shows the located $[\tilde{P}_{C_i}(\hat{x})]_{C_j}$ sets for the two new points (91 and 111) for $X^a$ having 4 variables. For both points, the NN classifier seems to be again the more restrictive from the perspective of sieving results: all the elements of the sets located by the other architectures are not correctly classified by the NN, whereas many of the points located by the NN are correctly classified by the other classifiers.

| $C_i$ \ $C_j$ | LR | SVC | RF | NN |
|---|---|---|---|---|
| LR | 392 | 0 | 43 | 0 |
| SVC | 341 | 341 | 80 | 0 |
| RF | 14 | 13 | 94 | 0 |
| NN | 59 | 59 | 56 | 59 |

| $C_i$ \ $C_j$ | LR | SVC | RF | NN |
|---|---|---|---|---|
| LR | 217 | 0 | 58 | 0 |
| SVC | 177 | 177 | 111 | 44 |
| RF | 8 | 0 | 149 | 0 |
| NN | 100 | 56 | 76 | 100 |

Table 8: Sizes of the corresponding sieved Pareto sets $[\tilde{P}_{C_i}(\hat{x})]_{C_j}$ for individuals 91 (up) and 111 (down).

In order to illustrate the influence of increasing the number of actionable variables, Figure 10 displays the 3D projections of the 4-dimensional Pareto set, $\tilde{P}_{\text{LR}}(\hat{x}^a_{68})$, obtained by searching with 4 actionable variables. The right bottom figure, corresponding to the projection of the new approximated CE Pareto set on the previous 3 actionable variables, can be compared to image corresponding to LR in the left high corner of Figure 9. Note that, although similar, the 3D projection seems to gather more points than the 3-variable solution, again in accordance to Proposition 1.

Figure 11 displays the 3D projections of the 4-dimensional Pareto set, $P_{\mathcal{C}}(\hat{x}^a_{111})$, taking into account all classifiers at the same time, obtained by searching in the 4 actionable variables.

On the other hand, Figure 12 shows for the same point 111 the totally filtered Pareto set. Note that we can disclose the contribution provided by each machine, so that the sieved $\tilde{P}_{\text{SVC}}(\hat{x}^a_{111})$ and $\tilde{P}_{\text{NN}}(\hat{x}^a_{111})$ are distinguished by color. Comparing Figures 11 and 12 one can conclude that the results are related in accordance to Proposition 3. Whereas $\tilde{P}_{\mathcal{C}}$ is a mildly large set, the computation of $\tilde{P}^S$ provides information about the comparative performance of the classifiers.

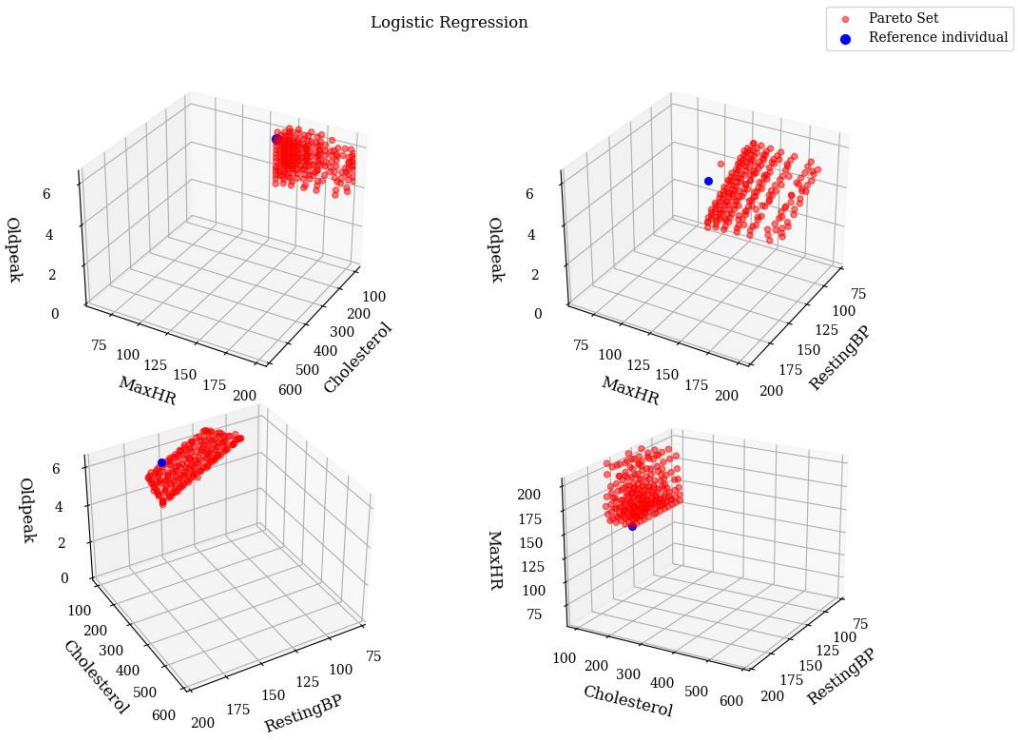


Figure 10: 3D projections of the approximated CE Pareto set, $\tilde{P}_{\text{LR}}(\hat{x}^a_{68})$, with 4 actionable variables.

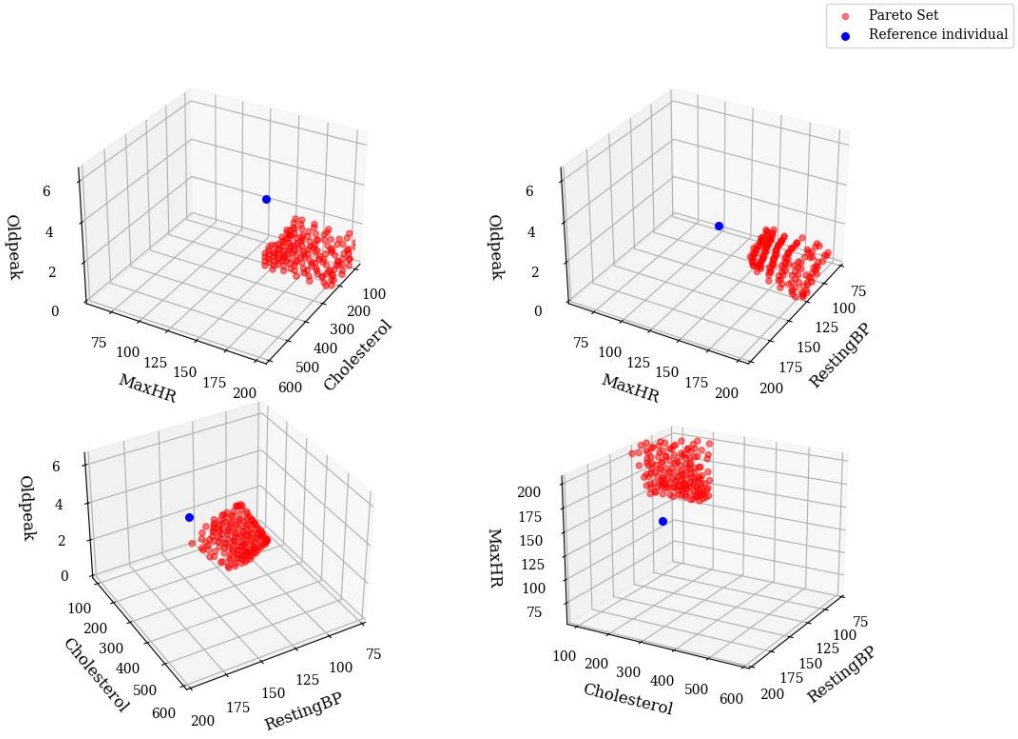


Figure 11: 3D projections of the approximated CE Pareto set, $\tilde{P}_{\mathcal{C}}(\hat{x}^a_{111})$, with 4 actionable variables.

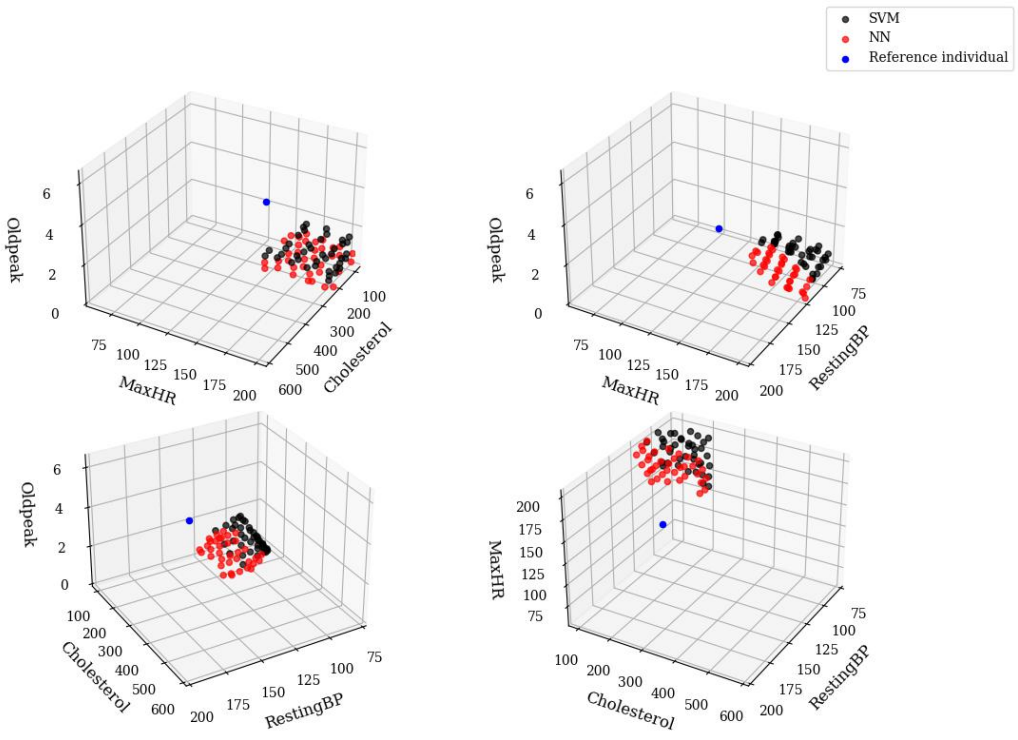


Figure 12: 3D projections of final totally filtered approximated CE Pareto set, $\tilde{P}^S(\hat{x}^a_{111})$, for example 111. The sieved contributions of the SVC and NN are distinguished by color.

## 7.2 Example 2: Bank Loan

In this case, we consider a classification problem using the German Credit dataset, which is available at `https://online.stat.psu.edu/stat508/book/export/html/796`. This dataset contains information about individuals applying for a bank loan, with various attributes that can be used to predict the likelihood of loan approval. The dataset has undergone minor transformations to improve clarity and readability.

The dataset consists of 1000 individuals with 20 features describing each individual's financial status and credit history. For this example, we focus on a logistic regression model to simulate a real-world scenario where the classifier is pre-defined, and the goal is to provide actionable counterfactual solutions to customers who are denied a loan by the model.

### 7.2.1 Model Training and Evaluation

We split the dataset into 70% for training the model and 30% for testing. The testing set allows us to evaluate the model's quality as a loan approval predictor for new customers; in addition one can also use the inputs corre-

sponding to such testing set as examples of customers for assessing our counterfactual explainability framework. The logistic regression model ($C_{LR}$) was trained using a *stepwise* method, ultimately including 12 variables. Nominal categorical variables were one-hot encoded, and ordinal categorical variables were label-encoded according to their order.

The performance metrics obtained are summarized in Table 9. While exploring other models might yield better results, it is not the primary focus of this example.

| **Metric** | **Training Set** | **Test Set** |
|---|---|---|
| Accuracy | 0.76 | 0.75 |
| Precision | 0.79 | 0.78 |
| Recall | 0.89 | 0.88 |

Table 9: Summary of Metrics for Training and Test Sets.

Out of the 300 individuals in the testing dataset, there are 63 instances where the model predicts a negative outcome (i.e., $C_{LR}(\hat{x}) = -1$, meaning the loan is not granted). These are the customers for whom we will calculate their approximating counterfactual solutions.

### 7.2.2 Computation of Pareto Sets

At the beginning of the computation process, 7 of the 12 explanatory variables considered in the model were identified as actionable: 3 categorical, 2 ordinal, and 2 continuous. This example incorporates variables of different types, making use of the distances exposed in Section 6.1.

In the following, we provide a brief description of the actionable variables:

- **Housing (Categorical):** The applicant's housing status: Free, Rented, and Own.
- **Account Balance (Categorical):** The applicant's account balance: No account, None, and Some balance.
- **Instalment (Ordinal):** Percentage of disposable income allocated to installment payments: More than 35%, (25%, 35%], [20%, 25%], and Less than 20%.
- **Savings (Ordinal):** The applicant's savings: None, Less than 100 DM, [100, 1000] DM, and More than 1000 DM.
- **Duration (Continuous):** The duration of the loan in months.
- **Amount (Continuous):** The amount of credit requested.

- **Concurrent Credits (Binary):** The applicant's concurrent credit status in other financial institutions: Other Banks or Stores and None.

Since we do not know the conditions/preferences of the clients, all potentially actionable variables were included. This ensures that the Pareto sets of counterfactual solutions include the solutions with few actionable variables also, allowing for the most suitable solutions to be selected post hoc, in line with Proposition 1.

The employed configuration parameters for the genetic algorithm were: a population size of 150 chromosomes, and 450 iterations. The algorithm efficiently found solutions for all 63 individuals where the model's prediction was negative. The cardinality of the solution sets $\tilde{P}_{C_{LR}}(\hat{x}^a)$ varied significantly among individuals (ranging from 6 to 150).

The relationship between the decision value and the number of approximating solution points found by the genetic algorithm, as illustrated in Figure 13, reveals an interesting trend: individuals whose decision values are closer to the decision boundary tend to have fewer approximating solution points in the approximated Pareto set $\tilde{P}_{C_{LR}}(\hat{x}^a)$, while those further from the boundary have a higher number of such points.

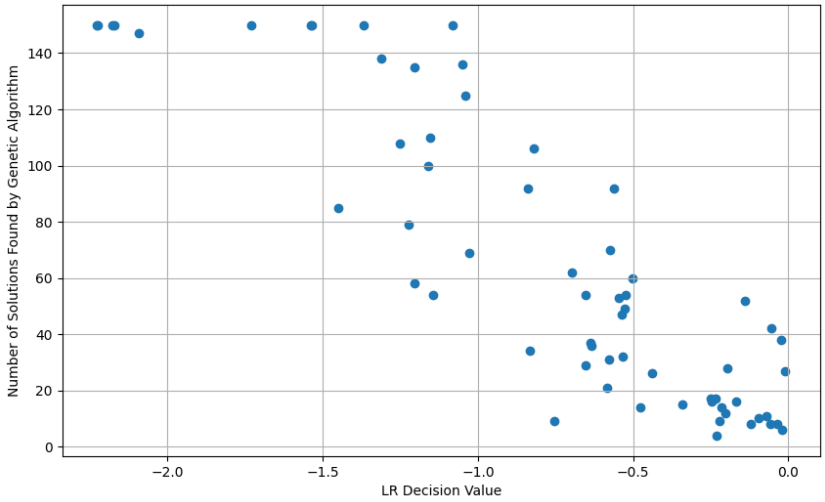


Figure 13: Comparison between LR Decision Value and Number of Solutions Found

This behavior can be explained by considering the relationship between the individual's proximity to the decision boundary and the geometric properties of the Pareto set. For individuals close to the decision boundary, small changes in input features can easily flip the classification outcome, resulting in a fairly limited region of the feature space where feasible counterfactual solutions exist. Consequently, the Pareto set for these individuals tends to be smaller, so that fewer approximating solution points are provided by the genetic algorithm.

In contrast, individuals further from the boundary have a more robust classification outcome, meaning that changes in the features still result in the same prediction outcome. This robustness creates a necessarily larger feasible region in the feature space, yielding a larger Pareto set with more approximating solution points. However, it's crucial to understand that the number of approximating solution points reflects the extent of the feasible region rather than the quality or the ease of finding these points.

Some of the approximations to the Pareto sets obtained, particularly those with low cardinality (for easy of exposure), are presented below:

$\tilde{P}_{C_{LR}}(\hat{x}^a_{63})$ Set:

6 counterfactual solutions were obtained for individual 63 (numbered according to the test order). It is important to note that the values for *Housing* and *Savings* have been suppressed in the solutions table as they did not change.

| **Feature** | **Original Value** |
|---|---|
| Account Balance | No account |
| Previous Credit | Paid Up |
| Purpose | Other |
| Sex Marital | Male Divorced/Single |
| Housing | Rented |
| Concurrent Credits | Other Banks or Dept Stores |
| Valuable Asset | Life Insurance |
| Age | 29 |
| Duration | 24 |
| Amount | 915 |
| Savings | $>$ 1000 DM |
| Instalment | $<$ 20% |

Table 10: Original Features for Individual 63

| **Account** | **Conc_Credits** | **Instalment** | **Duration** | **Amount** |
|---|---|---|---|---|
| No account | Other Banks or Dept Stores | $<$ 20% | 23 | 896.74 |
| Some Balance | Other Banks or Dept Stores | $<$ 20% | 24 | 915.00 |
| No account | None | $<$ 20% | 24 | 915.00 |
| No account | Other Banks or Dept Stores | [20%,25%] | 24 | 915.00 |
| No account | Other Banks or Dept Stores | $<$ 20% | 22 | 915.00 |
| No account | Other Banks or Dept Stores | $<$ 20% | 24 | 784.32 |

Table 11: Counterfactual Solutions for Individual 63

The counterfactual solutions in Table 11 suggest various ways individual 63 could alter his/her features of Table 10 to change the loan approval decision:

- Adjusting the account balance to have some balance or reducing the number of concurrent credits.
- Modifying the duration slightly to 22 months or reducing the amount requested by 131 DM.
- Decreasing the duration to 23 months and reducing the amount by only 18 DM (combined or mixed solution).
- Increasing the instalment percentage to [20%,25%).

$\tilde{P}_{C_{LR}}(\hat{x}^a_{262})$ Set:

9 counterfactual solutions were obtained for individual 262. As with the previous individual, the values for *Housing* and *Concurrent Credits* have been suppressed in the solutions table because they did not change.

| **Feature** | **Original Value** |
|---|---|
| Account Balance | None |
| Previous Credit | Paid Up |
| Purpose | Home Related |
| Sex Marital | Female |
| Housing | Rented |
| Concurrent Credits | None |
| Valuable Asset | Life Insurance |
| Age | 24 |
| Duration | 72 |
| Amount | 5595 |
| Savings | $<$ 100 DM |
| Instalment | (25%,35%] |

Table 12: Original Features for Individual 262

| **Account** | **Savings** | **Instalment** | **Duration** | **Amount** |
|---|---|---|---|---|
| None | $<$ 100 DM | (25%,35%] | 60 | 5489.18 |
| None | $<$ 100 DM | $>$ 35% | 61 | 5444.64 |
| None | $<$ 100 DM | (25%,35%] | 61 | 5398.53 |
| None | $<$ 100 DM | $>$ 35% | 59 | 5595.00 |
| Some Balance | $<$ 100 DM | (25%,35%] | 61 | 5595.00 |
| None | [100,1000] DM | (25%,35%] | 61 | 5595.00 |
| None | $<$ 100 DM | (25%,35%] | 59 | 5595.00 |
| None | $<$ 100 DM | $>$ 35% | 52 | 5595.00 |
| None | $>$ 1000 DM | (25%,35%] | 60 | 5595.00 |

Table 13: Counterfactual Solutions for Individual 262.

The counterfactual solutions in Table 13 suggest various ways individual 262 could alter her/his features (in Table 12) to change the loan approval decision:

- Reducing the loan duration between 52 and 61 months is mandatory to achieve a positive loan approval.
- Increasing the savings to more or equal than 100 DM, or increasing the instalment to $> 35\%$ could determine obtaining the credit depending on the loan duration.
- Changing the account balance to Some Balance, accompanied by a reduction of the loan duration from 72 to 61 months can also lead to credit approval.
- Reducing the amount of credit requested by 100-200 DM, along with a corresponding reduction in duration or increase in instalment, could also secure the loan approval.

These examples demonstrate the practical application of our framework, providing actionable insights to clients who have been denied a loan, and showing the flexibility and efficiency of our approach in handling different types of variables.

# 8 Conclusions

This thesis has presented a novel framework for Counterfactual Explainability in machine learning, focusing on a multi-criteria approach to handle the varying importance of input variables. The proposed methodology offers a flexible and robust solution to the challenges associated with understanding and interpreting the decisions made by machines. By introducing a multi-criteria distance metric, we have provided a more refined way to measure the feasibility and actionability of counterfactual explanations, accommodating both continuous and categorical variables.

Our experiments with binary classification models have demonstrated the effectiveness of our approach in generating realistic CEs. The results have highlighted the advantages of using multi-objective optimization techniques, such as the NSGA-III algorithm, to take into account multiple criteria and ensure diverse, actionable, and minimal modifications to the input instances.

The comparative analysis of different classifiers also emphasises the potential of our framework to improve the interpretability and consistency of machine learning models. By integrating insights from multiple classifiers, we have improved the reliability of the generated counterfactuals, providing a more comprehensive understanding of the model decision-making process.

As AI continues to be integrated into critical areas such as healthcare, finance, and criminal justice, the need for transparent and interpretable models will become increasingly capital. The ability to generate clear CEs will play a crucial role in our society for building trust and ensuring fairness. Our framework not only addresses current challenges but also sets the stage for future advancements in AI interpretability, supporting the development of ethical and responsible AI systems. This work lays a solid base for future research in explainability, offering practical tools and theoretical insights to advance in the field.

# 9 Future Work

Future research in CE can explore several promising directions.

First, Proposition 2, whose applicability has been defined for numerical variables, might be extended to the cases in which some ordinal variables are also involved.

Another area of research can address the use of Pareto Sets of counterfactuals as a means to compare the consistency of different models through relevant metrics. Additionally, comparing models trained with various loss functions could provide insights into their robustness and stability.

Investigating the sensitivity of models to small input feature changes may reveal those with higher variance, guiding the development of more stable models. Applying CEs to individuals with incomplete data could improve classification accuracy by inferring the most probable output through varying the missing features.

Finally, enhancing the sequential aspect of CEs by providing a series of actionable steps to achieve counterfactual solutions, as introduced by Naumann and Ntoutsi in their work [24], would make these explanations more practical and useful in real-world applications. These directions will advance the field of CE, promoting the development of more interpretable, reliable, and actionable machine learning models.

# A Implications for Sustainable Development Goals (SDGs)

The framework developed in this thesis, which focuses on explainability in machine learning, aligns closely with several of the United Nations' Sustainable Development Goals (SDGs), particularly those aimed at reducing inequalities (SDG 10), promoting peace, justice, and strong institutions (SDG 16), and ensuring quality education (SDG 4). Explainable machine learning models are crucial for identifying and mitigating biases that can lead to unfair or discriminatory outcomes. For example, if our framework detects that changing certain attributes, such as race or gender, alters the prediction of a model, it highlights potential biases. Addressing these biases ensures that machine learning systems contribute to equity and justice, rather than perpetuating discrimination.

Moreover, by making machine learning models more transparent and interpretable, our framework supports informed decision-making in various sectors, thereby contributing to sustainable economic growth (SDG 8) and fostering innovation in industry and infrastructure (SDG 9). Transparent models enable stakeholders to understand the rationale behind predictions, facilitating decisions that align with sustainability and ethical standards. In this way, the application of our framework not only improves the fairness and reliability of machine learning models but also supports broader goals of sustainability and social justice.

# B Proof of Propositions

## B.1 Proof of Proposition 1

Consider $SP_C(\hat{x}^a)$, the (Strict) Pareto set, solution of (9), where the search has been performed within space $X^a$. The corresponding (Strict) Pareto Front is $SF_C(SP_C(\hat{x}^a)) = d_X^a(\hat{x}^a, SP_C(\hat{x}^a))$.

Consider now the search in the whole space $X$ defined in (2). In that scenario, the points of the form $\hat{x}^f \times SP_C(\hat{x}^a) \subset X$ have an associated cost (see (9)):

$$d_X(\hat{x}, (\hat{x}^f, SP_C(\hat{x}^a)) = (\underbrace{0, \ldots, 0}_{n_f \text{ elements}}, d_X^a(\hat{x}^a, SP_C(\hat{x}^a))). \tag{28}$$

Note that if the search were restricted to $\{x \in \hat{x}^f \times X^a \mid C(x) = -C(x)\}$, the first $n_f$ components of $d_X(\hat{x}, x)$ would remain being zero; furthermore, by construction, the elements $x \in \hat{x}^f \times SP_C(\hat{x}^a)$ would obviously satisfy the condition $C(x) = -C(\hat{x})$ and they would also be SPE within the restricted search set.

Consider now any $x \in X = X^f \times X^a$ such that $x \notin \hat{x}^f \times X^a$; it implies that $x^f \neq \hat{x}^f$ so that $d_X^f(\hat{x}^f, x^f) \neq (0, \ldots, 0)$. Therefore $x$ cannot weakly dominate any element of $\hat{x}^f \times SP_C(\hat{x}^a)$, and this implies that all elements of $\hat{x}^f \times SP_C(\hat{x}^a)$ belong to $SP_C(\hat{x})$. □

## B.2 Proof of Proposition 2

We start by proving that every point of $H \cap \overline{CH}$ is Pareto dominated. Let us consider $x \in H \cap \overline{CH}$, so that it is characterized via (17) with some $\lambda_j < 0$. Note that from $\sum_{i=1}^n \lambda_i = 1$ there must always exist some $\lambda_k > 0$; selecting $\delta = \min\{|\lambda_j|, \lambda_k\}$ we can define $\lambda'_j = \lambda_j + \delta$ and $\lambda'_k = \lambda_k - \delta$.

Consider now the point $x'$ defined by preserving the rest of $\lambda_i$ values of $x$ and with the new $\lambda'_j$ and $\lambda'_k$ values: we have that $x' \in H$ and it dominates $x$ since $|\lambda'_j| < |\lambda_j|$ and $|\lambda'_k| < |\lambda_k|$. Note that the dominating point $x'$ may or may not be PE, but by this reasoning all points in $H \cap \overline{CH}$ are Pareto dominated.

We prove now that every point of $CH$ is PE. Consider $x \in CH$ so that the corresponding $\lambda_i$ values satisfy $\lambda_i > 0$ and $\sum_{i=1}^n \lambda_i = 1$, and assume there is other point $x'$ in $H$ that dominates $x$, i.e., for some $j$, we have that $|\lambda'_j| < |\lambda_j|$.

Since $\lambda_j > 0$ and $\sum_{i=1}^n \lambda_i = 1$ we have that $\sum_{i \neq j} \lambda_i = 1 - \lambda_j < 1 - \lambda'_j =$

$\sum_{i \neq j} \lambda'_i$ (irrespective of the sign of $\lambda'_j$). Hence, there must exist some $k \neq j$ for which $\lambda_k < \lambda'_k$ and since $\lambda_k > 0$, we have $|\lambda_k| < |\lambda'_k|$ leading to a contradiction. $\square$

## B.3 Proof of Proposition 3

Given $C_i, C_j \in \mathcal{C}$ we have that every point $x \in P_{\{C_i,C_2\}}(\hat{x})$ is PE the search having been restricted to $\{x \in X \mid C_i(x) = C_j(x) = -C_i(\hat{x})\}$. Obviously $x$ may not be PE if (keeping the same cost vector) the search is enlarged to $\{x \in X \mid C_i(x) = -C_i(\hat{x})\}$. Thus, $x$ may not be included in $[P_{C_i}(\hat{x})]_{C_j}$.

On the other hand, every point $x \in [P_{C_i}(\hat{x})]_{C_j}$ is PE the search having been performed in $\{x \in X \mid C_i(x) = -C_i(\hat{x})\}$. If the search is restricted to $\{x \in X \mid C_i(x) = C_j(x) = -C_i(\hat{x})\}$ we have that $x$ still belongs to the search space and it will keep being PE. $\square$

# References


[1] S. Wachter, B. Mittelstadt, and C. Russell. "Counterfactual explanations without opening the black box: Automated decisions and the GDPR". In: *Harv. JL & Tech.* 31.2 (2018), pp. 841–887.

[2] A. Adadi and M. Berrada. "Peeking inside the black-box: a survey on explainable artificial intelligence (XAI)". In: *IEEE Access* 6 (2018), pp. 52138–52160.

[3] D. V. Carvalho, E. M. Pereira, and J. S. Cardoso. "Machine learning interpretability: A survey on methods and metrics". In: *Electronics* 8.8 (2019), p. 832.

[4] C. Zednik. "Solving the black box problem: A normative framework for explainable artificial intelligence". In: *Philosophy & Technology* 34.2 (2021), pp. 265–288.

[5] M. T. Ribeiro, S. Singh, and C. Guestrin. "Model-agnostic interpretability of machine learning". In: *arXiv preprint arXiv:1606.05386* (2016).

[6] R. Mochaourab et al. "Post hoc explainability for time series classification: Toward a signal processing perspective". In: *IEEE Signal Processing Magazine* 39.4 (2022), pp. 119–129.

[7] P. J. Zufiria, I. Fernández Sánchez-Pascuala, and C. R. Rojas. "Multicriteria Model-Agnostic Counterfactual Explainability for Classifiers". In: *2024 International Joint Conference on Neural Networks (IJCNN)*. 2024, pp. 1–8.

[8] C. Molnar. *Interpretable Machine Learning*. 2020.

[9] S. Verma et al. "Counterfactual explanations and algorithmic recourses for machine learning: A review". In: *arXiv preprint arXiv:2010.10596* (2020).

[10] M. Van De Velden et al. "A general framework for implementing distances for categorical variables". In: *arXiv preprint arXiv:2301.02190* (2023).

[11] Y. Jin. *Multi-Objective Machine Learning*. Springer, 2006.

[12] P. J. Zufiria, C. Borrajo, and M. Taibo. "Multi-objective machine training based on Bayesian hyperparameter tuning". In: *2023 International Joint Conference on Neural Networks (IJCNN)*. 2023, pp. 1–7.

[13] S. Gu, R. Cheng, and Y. Jin. "Multi-objective ensemble generation". In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 5.5 (2015), pp. 234–245.

[14] E. R. Fernandes, A. C. de Carvalho, and X. Yao. "Ensemble of classifiers based on multiobjective genetic sampling for imbalanced data". In: *IEEE Trans. Knowl. Data Eng.* 32.6 (2019), pp. 1104–1115.

[15] Z. Wang et al. "Multi-objective feature attribution explanation for explainable machine learning". In: *ACM Transactions on Evolutionary Learning* (2023).

[16] J. C. Gower. “A general coefficient of similarity and some of its properties”. In: *Biometrics* (1971), pp. 857–871.

[17] S. Dandl et al. “Multi-objective counterfactual explanations”. In: *International Conference on Parallel Problem Solving from Nature*. 2020, pp. 448–469.

[18] Marco Tulio Ribeiro, Sameer Singh, and Carlos 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*. 2016, pp. 1135–1144.

[19] Divyat Mahajan, Chenhao Tan, and Amit Sharma. *Preserving Causal Constraints in Counterfactual Explanations for Machine Learning Classifiers*. 2020. arXiv: 1912.03277 [cs.LG].

[20] Sahil Verma, Keegan Hines, and John P. Dickerson. “Amortized Generation of Sequential Counterfactual Explanations for Black-box Models”. In: (2021). arXiv: 2106.03962 [cs.LG].

[21] Ramaravind K. Mothilal, Amit Sharma, and Chenhao Tan. “Explaining machine learning classifiers through diverse counterfactual explanations”. In: *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*. FAT* ’20. Barcelona, Spain: Association for Computing Machinery, 2020, 607â617. ISBN: 9781450369367. DOI: 10.1145/3351095.3372850. URL: https://doi.org/10.1145/3351095.3372850.

[22] Amir-Hossein Karimi et al. *Model-Agnostic Counterfactual Explanations for Consequential Decisions*. 2020. arXiv: 1905.11190 [cs.LG].

[23] Shubham Sharma, Jette Henderson, and Joydeep Ghosh. “CERTIFAI: A Common Framework to Provide Explanations and Analyse the Fairness and Robustness of Black-box Models”. In: *Proceedings of the 2020 AAAI/ACM Conference on AI, Ethics, and Society (AIES ’20)*. New York, NY, USA: Association for Computing Machinery, 2020, pp. 166–172. ISBN: 978-1-4503-7110-0. DOI: 10.1145/3375627.3375812. URL: https://doi.org/10.1145/3375627.3375812.

[24] P. Naumann and E. Ntoutsi. “Causal Disentanglement for Counterfactual Explanations”. In: *arXiv preprint arXiv:2104.05592* (2021).

[25] Maximilian Schleich et al. *GeCo: Quality Counterfactual Explanations in Real Time*. 2021. arXiv: 2101.01292 [cs.LG].

[26] Fabian Hinder and Barbara Hammer. *Counterfactual Explanations of Concept Drift*. 2020. arXiv: 2006.12822 [cs.LG].

[27] Deborah Sulem et al. “Diverse Counterfactual Explanations for Anomaly Detection in Time Series”. In: *arXiv preprint arXiv:2203.11103* (2022). DOI: 10.48550/ARXIV.2203.11103. URL: https://doi.org/10.48550/ARXIV.2203.11103.

[28] Roozbeh Yousefzadeh and Dianne P. O'Leary. "Debugging Trained Machine Learning Models Using Flip Points". In: (2019). Presented at the Debugging Machine Learning Models workshop at ICLR 2019. URL: https://debug-ml-iclr2019.github.io/cameraready/DebugML-19_paper_11.pdf.

[29] Mohammed Temraz and Mark T. Keane. "Solving the Class Imbalance Problem Using a Counterfactual Method for Data Augmentation". In: *arXiv preprint arXiv:2111.03516* (2021). URL: https://doi.org/10.48550/ARXIV.2111.03516.

[30] Tri Minh Nguyen et al. "Counterfactual Explanation with Multi-Agent Reinforcement Learning for Drug Target Prediction". In: *arXiv preprint arXiv:2103.12983* (2021). URL: https://arxiv.org/abs/2103.12983.

[31] Jake Fawkes, Robin Evans, and Dino Sejdinovic. "Selection, Ignorability and Challenges With Causal Fairness". In: *arXiv preprint arXiv:2202.13774* (2022). URL: https://doi.org/10.48550/ARXIV.2202.13774.

[32] Chelsea M. Myers et al. "Revealing Neural Network Bias to Non-Experts Through Interactive Counterfactual Examples". In: *arXiv preprint arXiv:2001.02271* (2020). URL: https://doi.org/10.48550/ARXIV.2001.02271.

[33] Matthias Ehrgott. *Multicriteria Optimization*. 2nd. Springer, 2005.

[34] Thibault Laugel et al. "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*. Springer. 2018, pp. 100–111.

[35] M. A. Braun. "Scalarized preferences in multi-objective optimization". Ph.D. dissertation. Karlsruher Instituts für Technologie (KIT), 2018.

[36] P. J. Zufiria and J. A. Álvarez-Cubero. "Generalized lexicographic multiobjective combinatorial optimization. Application to cryptography". In: *SIAM Journal on Optimization* 27.4 (2017), pp. 2182–2201.

[37] Kalyanmoy Deb and Himanshu Jain. "An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach: Part I - Solving Problems With Box Constraints". In: *IEEE Transactions on Evolutionary Computation* 18.4 (2014).

[38] Bobak Shahriari et al. "Taking the human out of the loop: A review of Bayesian optimization". In: *Proceedings of the IEEE* 104.1 (2016), pp. 148–175.