Universidade Estadual de Campinas
Instituto de Computação

Arthur Hendricks Mendes de Oliveira

MAPOFCEM: Model-Agnostic Pareto-Optimal Feasible Counterfactual Explanations Mining

MAPOFCEM: Mineração de Explicações Contrafactuais Viáveis de Modelo Agnóstico Pareto-ótimo

CAMPINAS
2024

Arthur Hendricks Mendes de Oliveira

MAPOFCEM: Model-Agnostic Pareto-Optimal Feasible Counterfactual Explanations Mining

MAPOFCEM: Mineração de Explicações Contrafactuais Viáveis de Modelo Agnóstico Pareto-ótimo

Dissertação apresentada ao Instituto de Computação da Universidade Estadual de Campinas como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação.

Supervisor/Orientador: Prof. Dr. Marcos Medeiros Raimundo

Este exemplar corresponde à versão final da Dissertação defendida por Arthur Hendricks Mendes de Oliveira e orientada pelo Prof. Dr. Marcos Medeiros Raimundo.

CAMPINAS
2024

Ficha catalográfica
Universidade Estadual de Campinas (UNICAMP)
Biblioteca do Instituto de Matemática, Estatística e Computação Científica
Ana Regina Machado - CRB 8/5467

OL4m

Oliveira, Arthur Hendricks Mendes de, 1998-
MAPOFCEM : Model-Agnostic Pareto-Optimal Feasible Counterfactual Explanations Mining / Arthur Hendricks Mendes de Oliveira. – Campinas, SP : [s.n.], 2024.

Orientador: Marcos Medeiros Raimundo.
Dissertação (mestrado) – Universidade Estadual de Campinas (UNICAMP), Instituto de Computação.

1. Sistemas de credit scoring. 2. Aprendizado de máquina. I. Raimundo, Marcos Medeiros, 1988-. II. Universidade Estadual de Campinas (UNICAMP). Instituto de Computação. III. Título.

Informações Complementares

Título em outro idioma: MAPOFCEM : Mineração de Explicações Contrafactuais Viáveis de Modelo Agnóstico Paretoótimo
Palavras-chave em inglês:
Credit scoring systems
Machine learning
Área de concentração: Ciência da Computação
Titulação: Mestre em Ciência da Computação
Banca examinadora:
Marcos Medeiros Raimundo [Orientador]
Luis Gustavo Nonato
Marcelo da Silva Reis
Data de defesa: 29-08-2024
Programa de Pós-Graduação: Ciência da Computação

Identificação e informações acadêmicas do(a) aluno(a)
- ORCID do autor: https://orcid.org/0009-0001-0275-9251
- Currículo Lattes do autor: http://lattes.cnpq.br/5481026048248453

Universidade Estadual de Campinas
Instituto de Computação

Arthur Hendricks Mendes de Oliveira

MAPOFCEM: Model-Agnostic Pareto-Optimal Feasible Counterfactual Explanations Mining

MAPOFCEM: Mineração de Explicações Contrafactuais Viáveis de Modelo Agnóstico Pareto-ótimo

Banca Examinadora:

- Prof. Dr. Marcos Medeiros Raimundo.
  Universidade Estadual de Campinas
- Prof. Dr. Luis Gustavo Nonato
  Universidade de São Paulo Campus de São Carlos
- Prof. Dr. Marcelo da Silva Reis
  Universidade Estadual de Campinas

A ata da defesa, assinada pelos membros da Comissão Examinadora, consta no SIGA/Sistema de Fluxo de Dissertação/Tese e na Secretaria do Programa da Unidade.

Campinas, 29 de agosto de 2024

Acknowledgments

My journey in Campinas was arduous, yet enriching. Over the years, I have learned many valuable lessons and realized that happiness is not derived from a place, but from the context and the sum of all its factors. São Paulo welcomes you in its own particular way, and you embrace São Paulo. It's not merely a relationship of tolerance, but one of sharing and learning. Therefore, I extend my heartfelt recognition and gratitude to everyone who has contributed, directly or indirectly, to my development thus far. Without your support, this dream would be meaningless.

First and foremost, I would like to thank God for never abandoning me on this journey. It was during the most challenging moments, when I felt weak and hopeless, that He revealed His glory to me, giving me the strength to rise again, more joyful and motivated to face the challenges ahead.

I am deeply grateful to my parents, Pedro and Ana, who believed in me and gave me the freedom to pursue my dreams. I also extend my gratitude to other family members, especially my grandparents Pedro and Avani, Genival and Nina (in memoriam), my aunts Kalina, Chaiane, and Sandra, my brother Augusto, and my cousins Gabrielle, Ariana, Jairo, Mirian and Rafaela, for embodying the true meaning of familial love. I carry a part of each of you wherever I go.

I would like to express special thanks to my fiancée, Samilly, for being my daily source of encouragement in the face of difficulties and for comforting me during moments of weakness, stress, and anxiety. Thank you for believing in every small step I took along this path and, most importantly, for sharing in the great victories throughout this entire journey. You are the embodiment of many of these joyful moments. I am also grateful to your family for their warm welcome and love.

I am grateful to Professor Marcos for his guidance, patience, and humanity throughout our interactions. I also extend my thanks to my colleague Giovani and the rest of my teammates for being pivotal in achieving the results of this work. Additionally, I am thankful to the Recod.ai and H.IAAC laboratories, as well as the Unicamp Institute of Computing, for providing a sense of home during many crucial moments.

To my companions, with whom I had the honor of sharing fantastic moments both inside and outside the university: Athyrson, Luan, Isabela, André, Camila, Jansen, João, Juan, Aline, Beatriz, Daniel, Vladimir, Patrick, Victor, Sadeeq, Soheil and Filipe. Thank you for your company. Without you, I wouldn't have come this far. I carry with me all the cherished memories of the days we spent together.

Take these broken wings and learn to fly
Paul McCartney, **Blackbird**

Resumo

Os modelos de aprendizado de máquina estão assumindo um papel cada vez maior na tomada de decisões de pontuação de crédito devido à sua precisão na previsão do reembolso do empréstimo. Contudo, uma crítica relativa à implementação destes modelos é a dificuldade de explicar a tomada de decisão do algoritmo para indivíduos cujos pedidos de crédito foram rejeitados. Estudos recentes revelam que explicações contrafactuais fornecem aos usuários o feedback da decisão do modelo através de uma lista de mudanças que eles podem fazer em seu perfil para orientar aplicações futuras. Portanto, fornecer explicações contrafactuais viáveis é um factor crucial para garantir que as alterações propostas estejam ao alcance dos utilizadores. Propomos um método chamado Model-Agnostic Pareto-Optimal Feasible Counterfactual Explanations Mining (MAPOFCEM) para fornecer feedback viável e acionável sobre decisões tomadas por um algoritmo de risco de crédito. Este método permite que indivíduos a quem foi negado um empréstimo façam ajustes específicos em seus perfis, aumentando assim suas chances de aprovação de empréstimos no futuro. Nossa abordagem integra um mecanismo de detecção de valores discrepantes no processo de busca contrafactual para gerar explicações contrafactuais viáveis. Os resultados experimentais demonstram que o MAPOFCEM fornece uma estrutura mais viável e robusta em comparação com os modelos de referência de código aberto existentes na literatura, o que aumenta a usabilidade de tais ferramentas para avaliar modelos de risco de crédito em aplicações do mundo real.

Palavras-chave: Pontuação de Crédito; Aprendizado de Máquina; Explicações Contrafactuais.

Abstract

Machine learning models are assuming an ever-expanding role in making credit score decisions due to their accuracy in predicting loan repayment. However, a criticism concerning the implementation of these models is the difficulty of explaining the algorithm's decision-making for individuals whose credit applications have been rejected. Recent studies reveal that counterfactual explanations provide to users the model decision feedback through a list of changes they can make to their profile to guide future applications. Therefore, providing feasible counterfactual explanations is a crucial factor in ensuring that proposed changes are within the reach of users. We proposed a method called Model-Agnostic Pareto-Optimal Feasible Counterfactual Explanations Mining (MAPOFCEM) to provide feasible and actionable feedback regarding decisions made by a credit risk algorithm. This method empowers individuals who have been denied a loan to make specific adjustments to their profiles, thereby increasing their chances of loan approval in the future. Our approach integrates an outlier detection mechanism within the counterfactual search process to generate feasible counterfactual explanations. The experimental results demonstrate that MAPOFCEM provides a more feasible and robust framework compared to existing open-source reference models in the literature, which enhances the usability of such tools for evaluating credit risk models in real-world applications.

Keywords: Credit Scoring; Machine Learning; Counterfactual Explanations.

List of Figures

List of Tables

Contents

Chapter 1

Introduction

Bank lending has become almost exclusively algorithmic in developed countries like the United States [3]. Machine Learning approaches predict an individual's willingness and ability to repay a loan more accurately than older approaches [12]. Such techniques support the classification of risk from patterns understood through data, such as personal information, name restrictions, credit profile, and income of the candidate [15]. However, one of the great points of criticism in applying these models is to explain the algorithm's decision-making process to the individuals who were credit denied. In this context, this Master Dissertation aims to explain the decision made by a Credit Risk algorithm through actionable feedback so the denied individual can make feasible changes to increase their chances of obtaining a loan in the future.

1.1 Motivation and Problem Overview

Enhancing the interpretability of a Machine Learning algorithm boosts user confidence in utilizing this powerful tool [13]. The comprehensive data science cycle illustrated in Figure 1.1 provides a holistic view of the entire process, highlighting key stages where interpretability can be integrated within the model learning stream [23] .

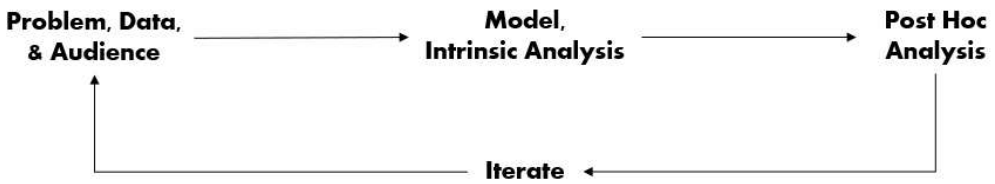

Figure 1.1: The generic cycle of a data science process.

The generic cycle of a data science process outlined by Murdoch et al. (2019) [23] emphasizes the importance of comprehensively understanding the problem, analyzing available data, and identifying the target audience to iteratively model a feasible solution. In the realm of interpreting problem/data, interpretability techniques can be effectively utilized both during the model development phase and in post-hoc analysis [23]. In the

context of Credit Risk Assessment, although the inherent interpretability of model development is valuable, delving into post-hoc analysis becomes essential due to the intricate nature of certain black box algorithms. In this context, Counterfactual Explanations, formalized by Wachter et al. (2017) [39], offer a solution to this challenge. These explanations provide actionable insights by outlining the changes a denied individual should make to their profile to secure credit approval. Several studies have employed counterfactual explanations to enhance understanding and decision-making processes. For instance, Ustun et al. (2019) [34] evaluated a linear classification model, focusing on implementing user profile changes to facilitate credit approval. Additionally, Poyiadzi et al. (2020) [27] introduced an algorithm that identifies feasible counterfactual paths incorporating weighted data density metrics.

Despite popularizing counterfactual explanations usage in recent years, its feasibility is a growing concern, since there is limited research dedicated to assessing the usability of counterfactual explanations by end-users [37]. This underscores the potential for innovation in developing solutions that enhance the user experience. Therefore, this project aims to build a framework that addresses the feasibility concerns of counterfactual explanations for Credit Risk Assessment models.

1.2 Contributions

We proposed a framework called Model-Agnostic Pareto-Optimal Feasible Counterfactual Explanations Mining (MAPOFCEM) to offer feasible and actionable insights concerning decisions made by a credit risk algorithm. This technique enables individuals with a denied loan to make targeted adjustments to their profiles, thereby increasing their chances of loan approval in the future. Our methodology incorporates an outlier detection mechanism within the counterfactual search process to produce feasible counterfactual explanations, thereby enhancing the practical applicability of such tools for assessing credit risk models in real-world scenarios.

1.3 Dissertation Structure

This dissertation is organized as follows:

- **Chapter 2.** It introduces the concepts and literature of Interpretability and Counterfactual explanations and identifies gaps that this research aims to address.
- **Chapter 3.** It covers the key concepts and theories that form the basis of the proposed framework, including outlier detection and counterfactual search processes.
- **Chapter 4.** It explains the innovative contributions of the proposed framework.
- **Chapter 5.** It outlines the research methodology employed in the study.
- **Chapter 6.** It presents the findings from our experiments.
- **Chapter 7.** It summarizes the key findings and limitations of the research.

Chapter 2

Interpretability and Literature Review

Integrating Machine Learning into various sectors of society has sparked numerous debates regarding the reliability of decisions made by artificial intelligence algorithms. Understanding these intricate models poses a significant challenge in facilitating their application across diverse scenarios. In this context, Interpretability and Explainability emerge as areas of study that aims to develop techniques that can reveal insights about the model's predictions [23]. Another perspective introduced by Varshney (2022) [36] suggests that interpretability can also guide developers in collaborating on model decisions to mitigate and manage their impacts.

Definition 1 (Interpretability)**.** *Interpretability is the degree to which a human can understand the cause of a decision or the internal mechanics of a model. An interpretable model allows users to see how inputs are transformed into outputs, making it easier to validate, debug, and trust the model. Interpretability is crucial for ensuring accountability, transparency, and fairness in AI systems.*

Definition 2 (Explainability)**.** *In the context of artificial intelligence and machine learning, explicability involves the ability to provide clear, understandable explanations for the behavior, decisions, or predictions made by a model. Explicability is important for building trust and ensuring that users can comprehend and reason about the outcomes produced by AI systems.*

In Credit Scoring, the necessity for both Interpretability and Explainability in intelligent algorithms becomes crucial, especially when decisions substantially impact users in situations that influence social welfare and require transparency. Various techniques are used to interpret machine learning models, and their relevance may vary according to the target audience and the objectives [36]. Molnar (2022) [21] established a taxonomy to determine the appropriate method to use, considering the following criteria:

- **Intrinsic and Post hoc methods.** In cases where the model exhibits simple structures and allows for a comprehensive view of the learning process, **Intrinsic** methods are suitable. For instance, Decision Trees with limited branches exemplify models where intrinsic methods can be effectively utilized. Conversely, when the model is highly complex, hindering direct comprehension, **Post hoc** methods are

employed post-training. For instance, understanding the learning process of a Gradient Boosting Tree individually can be challenging, requiring the creation of an approximation with multiple trees to grasp the impact of each feature.

- **Model-specific and Model-agnostic methods. Model-specific** methods leverage the internal characteristics of the model to elucidate its decisions. For instance, the gradient in a neural network can be utilized for a single example to identify the most influential feature for that particular prediction. On the other hand, **Model-agnostic** methods derive explanations from the correlation between the input data and output response, devoid of insights into the model's internal workings.
- **Local and Global methods. Global methods** facilitate the comprehension of the entire model and its behavior by utilizing approximations or delving into its internal workings, typically by elucidating the significance of features to the model and establishing the relationship with the data. Conversely, **Local methods** focus on explaining the behavior of a specific sample (individual) or a small subset of samples.

2.1 Global Methods

The techniques presented in this section can enhance comprehension of how features and samples impact algorithm predictions. Each approach employs various explanation tools, including visual and statistical methods.

Partial Dependence Plot (PDP) was introduced by Friedman (2001) [10] and stands as a prominent method for interpretability in research. PDPs analyze the impact of individual features on the model's average prediction through feature-specific plots. By using synthetic data points, PDPs may present misleading trends in certain features due to potential correlations among features.

Accumulated Local Effects (ALE) is a feature effect technique introduced by Apley and Zhu (2020) [1], offering an alternative to the issue of dependent features in Partial Dependence Plots (PDPs). ALE calculates the cumulative local effects on model predictions instead of relying on artificial class variations like the PDP method. While ALE is considered an unbiased PDP approach, its implementation is more intricate.

Feature Interaction is a global method that mathematically assesses the strength of interactions between features. It leverages Friedman's H-statistic [11] to quantify this strength and produces a plot that ranks interactions with other features and the target variable. This approach is valuable for evaluating the relationships and impact of features.

Functional Decomposition is a fundamental global technique applied in regression models. It visually illustrates the impact and interaction of features by breaking down the total of individual contributions per feature. This method visually represents each feature's influence on the model's prediction.

Permutation Feature Importance is a global method introduced by Breiman (2001) [5] and enhanced by Fisher et al. (2019) [8]. This method involves permuting feature values to assess the increase in prediction errors, enabling the visualization of feature influence on algorithm responses.

Global Surrogate is a technique commonly applied in engineering contexts to model real-world phenomena [21]. In Machine Learning, it utilizes an interpretable model, like a linear model or decision tree, to interpret predictions of a complex black box model. By approximating the black box model's predictions and leveraging its internal parameters, Global Surrogate explains the black box model's predictions.

Prototypes and Criticisms follow a distinct approach compared to other methods. It identifies clusters of data with similar behavior as "Prototypes" [29], while data points not well represented are termed "Criticisms". This method helps understand which features characterize Prototype behaviors.

2.2 Local Methods

In contrast to Global methods, Local methods focus on explaining individual predictions made by the model.

Individual Conditional Expectation (ICE) is a local method introduced by Goldstein (2015) [14] that functions similarly to a Global PDP method but for an individual instance. ICE manipulates feature values to observe how the prediction of the instance responds to the changes. Like the PDP method, it uses line plots to demonstrate the variation in behavior.

Local Surrogate (LIME) is another local method developed by Ribeiro (2016) [30]. In contrast to Global Surrogate models, LIME does not train an interpretable model to explain black box predictions. Instead, it generates perturbed samples around individual predictions and trains an interpretable Machine Learning model using these samples to understand the weights assigned by the interpretable model to that specific prediction based on proximity. LIME provides a visual plot to display the strength and predicted orientation of individual data point features.

Scoped Rules (Anchors) was developed by Ribeiro (2018) [31], the same author of Local Surrogate. This technique identifies a decision point that sufficiently anchors the model's prediction explanation. It applies reinforcement learning and perturbed samples, akin to LIME, to discover these anchors.

Shapley Values is a local method introduced by Shapley (1953) [33], utilizing the Global PDP method approximation to calculate differences in class features of individual instances compared to the model's average predictions. Similar to the PDP method, this technique employs plots for explanations.

SHAP (SHapley Additive exPlanations) was developed by Lundberg and Lee (2017) [19], incorporating the Shapley values technique. This method focuses on the absolute differences in Shapley values to elucidate feature importance in individual predictions.

Counterfactual Explanations is a local method introduced by Wachter (2017) [39]. It uses the features of an individual instance as input to explain the model's prediction compared to another instance with different predictions. By comparing feature differences using data points, it identifies adjustments needed in the original instance's features to alter its prediction.

2.3 Counterfactual Explanations

Using data points to explain a Machine Learning model's prediction can be highly beneficial in various scenarios, such as Credit Scoring. Consider a hypothetical individual who approaches a bank for a loan to fund his mother's cancer treatment. If the bank's Credit Risk Assessment Machine Learning algorithm denies the request without providing any feedback, it can result in significant frustration for the customer and strain his relationship with the bank. In such cases, Counterfactual Explanations can be employed to compare the characteristics of an approved customer, thereby offering actionable feedback on what changes the requestor could make to obtain the loan. For instance, suggestions might include increasing the salary to over $1.000 or attaining a higher education level, such as a graduate degree.

Although Counterfactual Explanations have a historical basis in social sciences, Wachter et al. (2017) [39] were the first to formulate it as an optimization problem. Since then, discussions have surfaced regarding the trade-off between enhancing the optimal method to achieve a Pareto solution and delivering a feasible explanation to the customer. Verma et al. (2022) [37] review several properties of Counterfactual Explanations that should be considered to distinguish the proposed methods:

- **Model Access.** This property pertains to the extent of information that the method can retrieve from the trained Machine Learning model. Methods can be categorized into three groups: those that offer **total access** to the model internals, such as linear models and decision trees, allowing a comprehensive understanding of model decisions or tree branches; those that provide **only access to the gradients**, enabling optimization of the counterfactual search by modifying the loss function; and **black box models**, which restrict access to just the input and output data.

- **Counterfactual Attributes.** Counterfactual Explanation methods can be further distinguished based on certain attributes. **Sparsity** involves how the model optimizes feature values during the counterfactual search, e.g., it is not feasible to use decreasing values for the age feature. **Data Manifold** or **Feasibility** solutions aim to find explanations close to the data distribution, enhancing the method's practicality. **Causality** considers the interrelations between features in the counterfactual

search, e.g., changing the educational level implies an increase in the customer's age due to the time required to achieve it.

- **Counterfactual Optimization.** Other Counterfactual Methods propose optimizing different attributes, such as **Fairness** by excluding features related to sex, race, etc. in the counterfactual search. Additionally, some methods handle categorical features by employing various techniques to measure the distance between values and their impact on the counterfactual search.

2.4 Feasibility in Counterfactual Explanations

Counterfactual Explanation methods recently proposed in the literature have combined different types of model access with various manipulation techniques to achieve feasibility. **Complete Access algorithms** have been proposed to Linear Models, such as Artelt et al. (2020) [2] that proposed a feasible solution using Gaussian Mixture to determine data density through Kernel Density Estimation (KDE). Kanamori et al. (2020)[16] introduced a framework employing Outlier Detection (LOF) to verify if the identified counterfactual is an outlier, thereby enhancing feasibility. Parmentier et al. (2021)[25] presented an efficient method using Isolation Forests to locate regions with low outlier scores in Ensemble Trees.

The majority of related works are trying to find optimal solutions using the **Total Model Access of Gradient Models**. For example, Mahajan et al. (2019) [20] employed Variational Autoencoders (VAEs) within a generative model, using loss functions to learn feasibility based on user feedback. Pawelczyk et al. (2020) [26] proposed C-CHVAE, an algorithm that leverages VAEs to measure the distance of an individual to the data distribution. Shao et al. (2022) [32] also used VAEs as density estimators in Sum-product Networks (SPNs), a "generative classifier" designed based on Bayes' rules. Similarly, Xiang et al. (2022) [40] utilized VAEs to measure counterfactual distances to k-nearest neighbors (k-NN).

The current challenge is to develop an efficient and feasible method for **Black Box models**. In this domain, most studies have employed density estimators to identify the data distribution. For instance, Dandl et al. (2020) [7] used a loss function that calculates the distance between the data point and the cluster using k-NN as a density estimator. Poyiadzi et al. (2020) [27] introduced FACE, an efficient algorithm that uses Dijkstra's algorithm to compute the shortest path through the data density, employing k-NN, KDE, and other density estimators. Studies by Nemirovsky et al. (2022) [24] and Van et al. (2021) [35] utilized the global method class prototype to find data distribution clusters, guiding the counterfactual search. Förster et al. (2021) [9] applied KDE with Gaussian Kernel for numerical data and Wang-Ryzin kernel for categorical data. Yang et al. (2021) [41] used Umbrella Sampling to allocate data distribution into "umbrella distributions" to enhance feasibility. Verma (2022) [38] also employed k-NN to ensure that the identified counterfactuals are close to the data distribution.

2.5 Related Works

Regarding methods for achieving the feasibility of counterfactual explanations for black-box models, Table 2.1 summarizes each method used by related works.

Table 2.1: Types of methods used to achieve feasibility in black box models.

Authors	Methods	Method Type
Dandl et al. (2020)	k-NN	Density Estimator
Poyiadzi et al. (2020)	k-NN, KDE	Density Estimator
Nemirovsky et al. (2022)	Class Prototype	Density Estimator
Förster et al. (2021)	KDE with Gaussian Kernel	Density Estimator
Verma (2022)	k-NN	Density Estimator

It is possible to see that the majority focus is on the use of density estimators to identify the distribution of data to obtain feasibility. Although all of which aim to ensure that the solutions they generate are representative of the underlying data distribution, it is important to note that while these methods focus on finding solutions within the kernel of the distribution, they often exclude many potential solutions at the edge of the data distribution, which is closer to the individual. This exclusion can limit the scope of solutions, potentially overlooking feasible explanations near the data point.

In the literature, some techniques applied to Full Access algorithms such as Kanamori et al. (2020)[16] and Parmentier et al. (2021)[25] who employed Outlier Detection to check counterfactual feasibility, explored solutions at the edge of data distribution without limiting user options. It improved the user experience, by identifying solutions closer to the user, that is, reducing the need for extensive modifications to the user profile. This technique was not explored for black-box models, which is the main research topic for this master's work. Table 2.2 summarized the properties of Counterfactual Explanations found in the work of Verma et al. (2022) [37] and shows the contribution of this work to the literature.

Table 2.2: Counterfactual properties considered in the scope of this project.

Model access	Sparsity	Feasibility	Causality	Method Type
Black-Box		X		Outlier Detection

In contrast to the approaches taken by Dandl et al. (2020) [7], Poyiadzi et al. (2020) [27], Nemirovsky et al. (2022) [24], Förster et al. (2021) [9], and Verma (2022) [38], which primarily focus on leveraging data density to provide feasibility in proposed solutions, this research innovates by employing outlier detection techniques to achieve a superior balance between maintaining alignment with the data distribution and ensuring proximity to the target instance, thereby enhancing the effectiveness and precision of the analysis.

Chapter 3

Theoretical Foundations

3.1 Counterfactual Explanations

Molnar (2022) [21] explains that Counterfactual Explanations are an accessible method for illustrating "what-if" scenarios. These explanations work by contrasting the features of an instance with slight modifications, thereby generating hypotheses like "if this aspect were different, then this outcome would occur".

Definition 3 (Counterfactual Explanations)**.** *Starting from a sample predicted as negative (e.g., an individual with denied credit), a counterfactual explanation consists of changes the individual with the denied resource should make to access the resource.*

3.2 Mining Counterfactual Explanations

Each counterfactual offers a unique "story" of how a specific outcome was reached. Therefore, mining counterfactual explanations involves exploring alternative instances within a feature space to understand the factors behind a specific outcome and identify actionable changes that could lead to a different result.

MAPOCAM

Model-agnostic Pareto-Optimal Counterfactual Antecedent Mining (MAPOCAM) is a technique introduced by Raimundo et al. [28] (2022) that searches counterfactuals within a feature space efficiently through a branch-and-bound search strategy.

Considering a binary classification problem, each sample consists of a feature vector $x \in \mathbb{R}^d$, where d comprises a vector of features, and a binary outcome $y \in \{0, 1\}$. The classifier is denoted by a decision function $r(\cdot)$, which outputs 1 if $r(x) \geq \tau$, where τ is a predetermined threshold. For simplicity, we will consider 1 to be the target outcome. An action $a = [a_1, ..., a_d]$ applied to an observation x is defined as a list of changes $a_i \in \mathbb{R}$, $\forall i \in \{1, ..., d\}$ on the feature values, resulting in a new synthetic sample $x + a$. Conterfactual Explanations are actions in which $r(x) \leq \tau$ and $r(x + a) \geq \tau$.

Definition 4 (Action)**.** *An action $a \in \mathbb{R}^d$ is a counterfactual explanation if and only if it achieves the desired outcome $r(x + a) \geq \tau$.*

An action that follows the preferences of the decision maker is obtained from an optimization process of an objective function c(·)∈ ℝ. Thus, actions are sought with the lowest value of the objective function. In this context, MAPOCAM proposes multiple objective functions $c_1(\cdot), \ldots, c_m(\cdot)$ to address different options for users in the feature space, where each objective function $c_i(\cdot)$, $i \in 1, \ldots, m$, represents a specific goal defined by the user. Nonetheless, it is possible to identify a dominant action — an action that performs better across all objectives compared to other actions.

Definition 5 (Dominant Action)**.** *An action $a : r(x + a) \geq \tau$ dominates a if and only if $c(a) \succeq c(a')$.*

The idea of dominant action in the context of multi-objective optimization leads us to the definition of the concept of Pareto-optimal action.

Definition 6 (Pareto-Optimal Action)**.** *Given an objective function vector $c(\cdot) : \mathbb{R}^d \to \mathbb{R}^m$ that we want to minimize and a set of solutions $\mathcal{A}$, an action a' is Pareto-optimal iff there is no action $a \in \mathcal{A}$ that dominates a'.*

MAPOCAM Algorithm

Algorithm 1 presents the pseudocode for MAPOCAM. The algorithm functions as follows: it iterates through the search grid by modifying one feature at a time (line 20). For each potential feature value being changed, it evaluates the impact on secondary features, continuing this process until the maximum number of changes is reached. The vector $\mathcal{A}$ is used to record the solutions found (line 2); any value greater than 0 means a change.

In MAPOCAM's branch-and-bound search strategy, each parameter call corresponds to a decision node (line 1) that can generate additional nodes through a branching procedure or stop the exploration of subsequent nodes through a pruning procedure (line 20). Therefore, these nodes can serve as potential counterfactual explanations or endpoints in the search process.

Definition 7 (Node)**.** *A node consists of an action a: $a_i \geq 0$, $\forall i \in \{1, ..., d\}$ and a set of fixed features D.*

Definition 8 (Branching Procedure)**.** *The branching procedure consists of selecting a non-fixed feature $i \notin \mathcal{D}$, calling the recursive function with different actions' values $a_i \geq 0$ for that feature i that now is fixed $D \equiv D \cup i$ for the subsequent calls. All features with no decision are considered null $a_i = 0$, $\forall i \notin \mathcal{D}$ and might have their value altered in the subsequent branching.*

MAPOCAM establishes that since every recursive call will always increase the magnitude of the action $a_i \geq 0$, the associated cost of creating an action increases. This property is called monotonicity.

Definition 9 (Monotonicity)**.** *Given any two actions a, $a' \in \mathbb{R}^d$ such that $a_i \geq a$, $\forall i \in \{1, ..., d\}$, a function vector $f(\cdot)$: $R^d \to R^m$ is monotone if only if $f_j(a) \geq f_j(a')$, $\forall j \in \{1, ..., m\}$.*

Algorithm 1: MAPOCAM

Input: A sample x, an objective function c, an outlier detection function $\mu(\cdot)$, a decision rule $r(\cdot)$, a threshold τ and a number of allowed changes k.

1 **procedure** ENUMERATE$(a, \mathcal{D}, \mathcal{A})$
2 **if** $|i : a_i \neq 0 \forall i \in \mathcal{D}| > k$ *or* $\exists a' \in \mathcal{A} : c(a) \succeq c(a')$ **then**
3 return;
4 **end**
5 **if** $\overline{r} < \tau$ **then**
6 return;
7 **end**
8 **if** $r(x + a) \geq \tau$ **then**
9 $\mathcal{A} = \mathcal{A} \cup \{a\}$ return;
10 **end**
11 $i =$ SELECT_FEATURE$(\forall i : i \notin \mathcal{D})$;
12 **for** $\forall a' : a'_i \geq a_i$ **do**
13 ENUMERATE$(a', \mathcal{D} \cup \{i\}, \mathcal{A})$
14 **end**
15 **end procedure**
16 **Algorithm** MAPOCAM$(x, c, \mu, r(\cdot), \tau, k)$
17 $\mathcal{D} = \{\}, \mathcal{A} = \{\}$;
18 $a_i = 0 \ \forall a_i \in \{1,...,d\}$;
20 ENUMERATE$(a, \mathcal{D}, \mathcal{A})$;
22 **return** $\mathcal{A}$;

Since any recursive call will always increase the magnitude of the action $a_i \geq 0$, the monotonicity function stops the current call to make other recursive calls.

Definition 10 (Bounding Step)**.** *The bounding step consists of inspecting any branch that will not improve the optimal set of solutions; thus, the node should not go deeper.*

The bounding step can occur under some possible conditions:

- When the node has more than k changes: $|i : a_i \neq 0 \forall i \in \mathcal{D}| > k$. Since any other recursive call will increase the number of changes.
- When any subsequent bounding will not generate a feasible solution. Since any other recursive call will increase the size of the action (line 5).

The second pruning condition employed in the MAPOCAM algorithm involves evaluating the maximum probability that a partial counterfactual can achieve. This technique first calculates the initial probability of the given sample. Then, it checks whether this maximum probability is lower than τ. If this value is lower than the threshold, traversing this branch is unnecessary since counterfactuals will not be found.

Definition 11 (Maximum Probability)**.** *Given a node with an action a and a set of fixed features D. The model's probability consists of the maximal probability* $\max f(a')$ *where a' is an action with the same fixed features as a.*

MAPOCAM presents counterfactual explanations to the user by showcasing actionable insights through alternative instances in the feature space. This multi-objective criteria ability is one notable aspect of this algorithm. It allows the user to determine the optimal trade-off of variable changes.

Despite his promising contributions, MAPOCAM faces limitations. One significant constraint is its tendency to traverse a vast grid of feature values without ensuring they align with the underlying data distribution. This can lead to the generation of counterfactual explanations that, while technically valid, may not be practically feasible for end-users, reducing their utility. Another notorious limitation is its dependence on model monotonicity for computational efficiency. While it is effective for linear models, this assumption may not apply to complex models like neural networks, possibly impacting the algorithm's effectiveness.

3.3 Feasible Counterfactual Explanations

Generating alternative scenarios or instances that are valid but also practical and useful for end users improves the user experience and usability of counterfactual explanations. Verma et al. (2022) [37] shows that the literature defines feasible counterfactual explanations as instances that present data distribution behavior.

Definition 12 (Feasible Counterfactual Explanations)**.** *Feasible Counterfactual Explanations are composed of instances that present the behavior of the data distribution, making your actions more applicable and closer to reality.*

3.3.1 Outlier Detection

An efficient technique used by Kanamori et al. (2020)[16] to verify the feasibility of counterfactual explanations is Outlier Detection. Counterfactual explanations are assessed by incorporating outlier detection techniques to ensure they do not deviate significantly from the normal data distribution, thus enhancing their feasibility and relevance in decision-making processes.

Definition 13 (Outlier Detection)**.** *Outlier Detection or Anomaly Detection, refers to the process of identifying data points that significantly deviate from most of the dataset. These outliers differ markedly from other observations, potentially indicating variability in the data, errors, or novel insights.*

3.3.2 Isolation Forest

Liu et al. (2008) [18] proposed the Isolation Forest algorithm to classify outliers by analyzing the distance between regular and anomaly points. It creates random partitions or splits in the data, which helps to isolate anomalies more quickly than normal data points. The algorithm works by building an ensemble of isolation trees, where each tree is constructed by randomly selecting a feature and a split value to isolate anomalies. Thus, anomalies are identified as data points requiring fewer splits to isolate, indicating that

they differ from most of the data. It offers efficient computational processing compared to density estimation techniques because by replacing clusters of regular points with subsamples, the algorithm quickly identifies anomalies in a lower score sample space. This approach makes Isolation Forests more scalable than traditional methods in anomaly detection literature.

3.4 Model-Agnostic Counterfactual Explanations

Generating Model-Agnostic Counterfactual Explanations from a Machine Learning model is a complex task, as the model operates like a black box - concealing its internal workings. This process facilitates a deeper understanding of the decision-making process in an independent and flexible manner, regardless of the type of Machine Learning algorithm employed. Consequently, this approach broadens the applicability of counterfactual explanations to any type of Machine Learning model, enhancing their utility across diverse applications.

Definition 14 (Model-Agnostic Counterfactual Explanations)**.** *Model-Agnostic Counterfactual Explanations refer to the application of Contrafactual Explanation techniques that are independent of factors internal to the model to access its decision-making.*

SHAP Values

SHAP (SHapley Additive exPlanations) values proposed by Lundberg and Lee (2017) [19] serve as a crucial method in Machine Learning for elucidating the model's output by assigning significance to each feature's contribution towards the final prediction. These values are calculated by evaluating the model predictions with and without the inclusion of a specific feature, to verify the contribution of each feature or sample in the dataset.

By incorporating the intricate relationships and dependencies between features in prediction outcomes, SHAP offers a sophisticated and realistic assessment of feature importance. Unlike conventional feature importance techniques, which often provide an oversimplified view by considering features in isolation, SHAP accounts for the interaction effects and the contextual influence of each feature on the model's predictions. This nuanced approach allows SHAP to distribute the contribution of each feature fairly and consistently, based on cooperative game theory principles. As a result, it provides a more accurate and interpretable measure of feature importance, which is particularly valuable in complex models with interdependent features.

Although SHAP may incur slightly higher computational costs in terms of time, its combination with branch-and-bound counterfactual search techniques can provide significant benefits that justify this investment. The branch-and-bound method systematically explores the solution space, efficiently narrowing down potential counterfactuals by eliminating suboptimal branches. Consequently, the slightly increased computational time is offset by the substantial gains in the precision and reliability of the outcomes, making it a worthwhile trade-off in many applications where understanding and trust in model predictions are paramount.

Chapter 4

Proposed Framework

In this study, we introduce a novel framework that prioritizes the development of feasible solutions and underscores model-agnosticism for black box models. We call this framework Model-Agnostic Pareto-Optimal Feasible Counterfactual Explanations Mining (MAPOFCEM). Its structure is based on the MAPOCAM methodology proposed by Raimundo et al. (2022) [28] but ensures the feasibility of the solutions generated through an approach similar to that of Axel Parmentier and Tribaut Videl (2021) [25], which utilizes the Isolation Forest algorithm to identify outliers during the search for counterfactuals. Additionally, MAPOFCEM improves MAPOCAM by removing the constraint on model monotonicity necessary for an efficient search of counterfatuals. To this end, it promotes the exploration of solutions by leveraging SHAP [19] to assess the impact of each potential solution in the breach-and-bound search.

4.1 Excluding Outliers in the Counterfactual Search

One critical gap that MAPOFCEM aims to adress in MAPOCAM is its inability to provide actionable insights due to detachment from real-world constraints. MAPOCAM frequently explores an extensive range of feature values without ensuring alignment with the underlying data distribution. Consequently, this results in counterfactual explanations that may not be practically feasible for end-users, reducing their utility. To address these shortcomings, MAPOFCEM aims to integrate domain-specific knowledge and data distribution awareness by using Outlier Detection, in order to avoid proposing unfeasible counterfactuals in the proposed solutions.

The formulation of the problem is as follow: with a observation $x \in \mathbb{R}^d$, where d comprises a vector of features, and a binary outcome $y \in \{0, 1\}$ that is the output of a decision function $r : \mathbb{R}^d \to [0, 1]$. The output of r is the probability of the positive outcome, and is discretized with a threshold τ that y is 1 when $r(x) \geq \tau$ and 0 when $r(x) < \tau$. An action $a = [a_1, ..., a_d]$ on an observation x is a list of changes $a_i \in \mathbb{R}$, $\forall i \in \{1, ..., d\}$ on feature values that creates a new synthetic sample $x + a$.

We are interested in actions $x + a$ where $r(x) \leq \tau$ and achieve $r(x + a) \geq \tau$. Thus, let $\mathcal{A}' = \{a \in \mathbb{R} \mid r(x + a) \geq \tau\}$ be the set of solutions that are defined as conterfactual, and $\mu(\cdot)$ an outlier detection function (where $\mu(\cdot) = 0$ defines an inline sample and $\mu(\cdot) = 1$ an

outline sample). If $\mu(x+a) \neq 1$, i.e, $x+a$ is not an outlier, then a is a solution proposed by MAPOFCEM. The algorithm search for solutions in the set $\mathcal{A} = \{a \in \mathcal{A}' \mid \mu(x+a) \neq 1\}$.

We illustrate this scenario in Figure 4.1, where the Isolation Forest algorithm was applied to a set of data that has been classified by a Credit Risk Assessment algorithm as good and bad payers based on the loan amount requested and age. The green items refer to good payers, with approved credit analysis, and the red were credit denied. In this 2D visualization, it is possible to observe that the data density increases near the bottom left corner, that is, the majority of the data presents these characteristics. The points marked with "x", identified as outliers by the Isolation Forest algorithm, display values outside this data density, where feature values are less frequent compared to the majority of the data set, indicating sparser regions within the feature space.

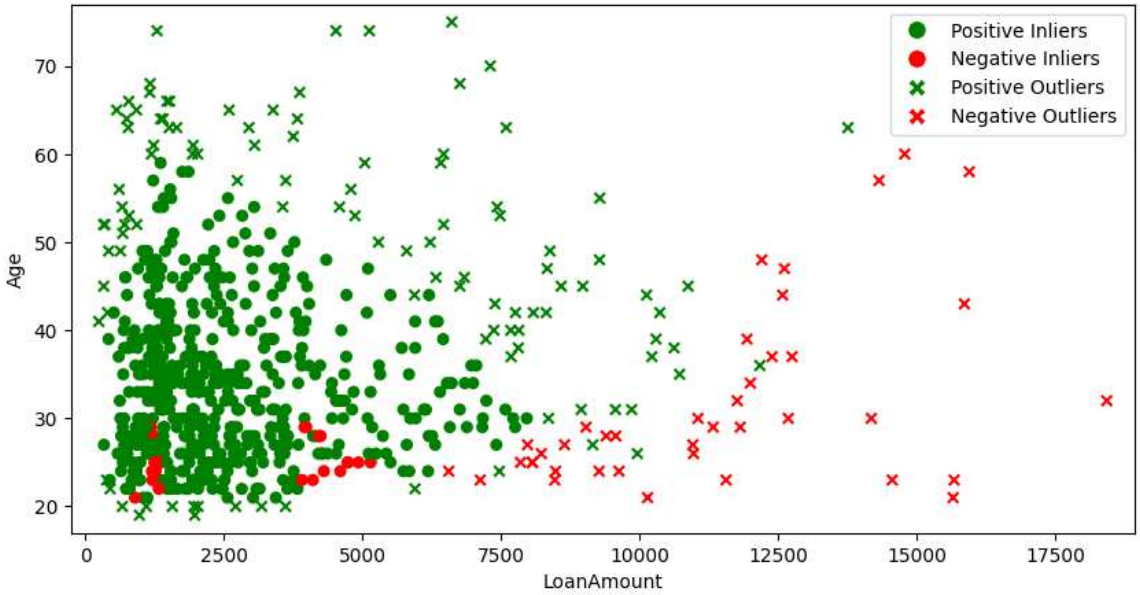

Figure 4.1: Outliet Detection technique.

By incorporating the criteria for choosing solutions based on outlier detection, the MAPOFCEM approach may restrict some of the solutions found by MAPOCAM. For instance, consider that for a given instance, MAPOCAM identified 5 solutions, which 3 of them are discrepant. MAPOFCEM search will consider only the 2 feasible solutions found by MAPOCAM and will continue the search if a number of solutions equal to 2 has not been defined. Therefore, evaluating the impact of Isolation Forest's Contamination hyperparameter is necessary, as it adjusts the model's sensitivity to anomaly detection. A substantial increase in this parameter leads to heightened outlier identification, potentially reducing the number of solutions yielded by MAPOFCEM. Given the necessity for domain-specific insights, this parameter was incorporated as a user-input criterion to enhance adaptability across different datasets, ensuring flexibility in application.

Furthermore, the outlier detection approach increases the efficiency of the counterfactual search, as it works as a pruning mechanism with the MAPOCAM branch-and-bound strategy. For instance, deeming a decision node non-viable can cascade to render subsequent nodes non-viable, streamlining the search process.

4.2 Adapting the Maximum Probability Function

The pruning technique employed in MAPOCAM [28] involves assessing the maximum probability that a partial counterfactual can achieve. This technique calculates the initial probability of the given sample. Then it verifies if the maximum probability is lower than τ. If this value is lower than the threshold, traversing through this branch becomes unnecessary, as counterfactuals will not be found.

MAPOCAM estimates the maximum probability under the assumption of model monotonicity. If a feature value increases and the probability increases, increasing this feature value even more should only increase the probability (or at least keep the same value). It is worth noting that many widely used Machine Learning models, including KNN, Neural Networks, SVM, and others, do not adhere to this assumption.

Considering Feature and Sample Importances

To address this issue, SHAP values [19] were employed to estimate an upper bound on the maximum probability in a node by considering the importance of features. In more detail, we have a partial solution x with a set of features $\mathcal{D}$ that will not have their values altered (a constraint set by the user fixes them or they are fixed because their values have already been altered previously in the search process).

Let $\mathcal{X}_{(x,\mathcal{D})}$ denote the space where all elements have the same values on the fixed features as x, i.e. if $\tilde{x} \in \mathcal{X}_{(x,\mathcal{D})} \implies x_i = \tilde{x}_i, \forall i \in \mathcal{D}$. The goal is to determine $\max_{\tilde{x} \in \mathcal{X}_{(x,\mathcal{D})}} r(\tilde{x})$, which represents the highest probability among observations with identical feature values as x in the features of $\mathcal{D}$. If $\max r(\tilde{x}) < \tau$, than there isn't any valid counterfactual in $\mathcal{X}_{(x,\mathcal{D})}$ and a search is not necessary in this region.

SHAP obtains additive feature attributions, i.e., for each sample x, the feature contribution value to the prediction can be calculated. For instance, $r(x) = \mathbb{E}[r(x)] + \sum_{i=0}^{d} \phi_i(x)$, where $\phi_i(x)$ represents the contribution of the i-th feature. Notice that $\sum_{i \in \mathcal{D}} \phi_i(x)$ reflects the contribution of fixed features, while $\sum_{i \notin \mathcal{D}} \phi_i(x)$ pertains to the open ones. These feature attributions can be used to estimate the maximum effect of an open feature (not in $\mathcal{D}$) have on the final prediction.

With access to a dataset $X \in \mathbb{R}^{n \times d}$, feature attributions can be calculated for all samples, resulting in a dataset of feature attributions $\Phi \in \mathbb{R}^{n \times d}$, where $\phi_{i,p}$ denotes the feature attribution of sample i for attribute p. Subsequently, the maximum contribution of each feature can be determined as $\overline{\phi}_p = \max_i \phi_{i,p}, \forall p \in \{1, \ldots, d\}$. This approach allows establishing the upper limit for the initial problem:

$$
\begin{aligned}
\max_{\tilde{x}\in\mathcal{X}_{(x,\mathcal{D})}} r(\tilde{x}) =& \mathbb{E}[r(x)] + \sum_{i\in\mathcal{D}} \phi_i(x^\star) + \sum_{i\notin\mathcal{D}} \phi_i(x^\star) \\
\leq& \mathbb{E}[r(x)] + \sum_{i\in\mathcal{D}} \phi_i(x^\star) + \sum_{i\notin\mathcal{D}} \overline{\phi}_i \\
=& \mathbb{E}[r(x)] + \sum_{i\in\mathcal{D}} \phi_i(x) + \sum_{i\notin\mathcal{D}} \overline{\phi}_i + \left(\sum_{i\in\mathcal{D}} \phi_i(x^\star) - \sum_{i\in\mathcal{D}} \phi_i(x)\right) \\
=& \mathbb{E}[r(x)] + \sum_{i\in\mathcal{D}} \phi_i(x) + \sum_{i\notin\mathcal{D}} \overline{\phi}_i + R(x,\mathcal{D}) \\
\approx& \mathbb{E}[r(x)] + \sum_{i\in\mathcal{D}} \phi_i(x) + \sum_{i\notin\mathcal{D}} \overline{\phi}_i \\
=& r(x) - \sum_{i\notin D} \phi_i(x) + \sum_{i\notin\mathcal{D}} \overline{\phi}_i
\end{aligned}
\tag{4.1}
$$

where $x^\star$ solves the maximization problem, $\phi_i(x)$ and $\phi_i(x^\star)$ are the SHAP values for the i-th feature for the solutions x and $x^\star$ respectivelly. Notice that this upper bound has a residual term $R(x,\mathcal{D})$ that is the difference between features attributions of the initial observation x and the solution $x^\star$ have on the set of fixed features $\mathcal{D}$. We can show that $R(x,\mathcal{D})$ is bounded by the size $\mathcal{X}_{(x,\mathcal{D})}$ when r has certain characteristics.

By leveraging an upper bound, the maximum probability of branches is overstated, which could increase computational time. Nevertheless, this overestimation does not compromise the quality of the solutions obtained, ensuring that unnecessary branches are pruned effectively. Since determining the maximum ($\overline{\phi}$) SHAP values for samples in $\mathcal{X}$ is not possible, we approximate these maximum SHAP values by selecting the highest value from n random samples within $\mathcal{X}$.

Considering the Number of Changes

We can make a small alteration to the previously formulated problem to further reduce computing costs. This is done by considering that the generation of counterfactuals is usually done with a constraint on how many features at max can be changed. Let k be the number of features that can be further altered from the partial solution x. $\mathcal{X}_{(x,\mathcal{D},k)} = \{\tilde{x} \in \mathcal{X}_{(x,\mathcal{D})} \mid \sum_{i=1}^{n} \mathbb{I}[x_i \neq \tilde{x}_i] \leq k\}$ i.e., is the set of elements that have at most k open features different from x. Let be $O_k \subseteq D^C$ be the set of k open features that have the k-biggest $\overline{\phi}_i$, i.e., if $i \in O_k \implies \overline{\phi}_i \geq \overline{\phi}_j \forall j \notin O_k$. Using the same idea from the previous proof, we can obtain a similar upper-bound:

$$
\max_{\tilde{x}\in\mathcal{X}_{(x,\mathcal{D},k)}} r(\tilde{x}) \leq r(x) - \sum_{i\in O_k} \phi_i(x) + \sum_{i\in O_k} \overline{\phi}_i \tag{4.2}
$$

As $O_k \subseteq D$ this formulation gives an upper bound that is lower than the previous one obtained, and more frequently will result in values smaller than τ, reducing the number of sets that need to be searched by the algorithm.

4.3 Algorithm

Algorithm 2 presents the pseudocode for MAPOFCEM. Changes made to MAPOCAM are colored blue within the code. The algorithm functions as follows: it iterates through the search grid by modifying one feature at a time. For each potential feature value being changed, it evaluates the impact on secondary features, continuing this process until the maximum number of changes is reached. The vector $\mathcal{A}$ is used to record the solutions found; any value greater than 0 means a change.

Algorithm 2: MAPOFCEM

Input: A sample x, an objective function c, an outlier detection function $\mu(\cdot)$, a decision rule $r(\cdot)$, a threshold τ and a number of allowed changes k.

```
1  procedure ENUMERATE(a, D, A)
2      if |i : a_i ≠ 0 ∀i ∈ D| > k or ∃a' ∈ A : c(a) ⪰ c(a') then
3          return;
4      end
5      x' ← copy of x stating unchanged values as missing D;
6      if μ(x') = 1 then
7          return
8      end
9      r̄ ← maximum probability of x + a with fixed values D;
10     if r̄ < τ then
11         return;
12     end
13     if r(x + a) ≥ τ then
14         if μ(x + a) ≠ 1 then
15             A = A ∪ {a} return;
16         end
17     end
18     i = SELECT_FEATURE(∀i : i ∉ D);
19     for ∀a' : a'_i ≥ a_i do
20         ENUMERATE(a', D ∪ {i}, A)
21     end
22 end procedure
23 Algorithm MAPOFCEM(x, c, μ, r(·), τ, k)
24     D = {}, A = {};
25     a_i = 0 ∀a_i ∈ {1,...,d};
27     ENUMERATE(a, D, A);
29     return A;
```

The search process begins by checking if the maximum allowable changes have been reached or if the new solution surpasses the cost of a previous solution (lines 2-4). In the following, the algorithm verifies if the partial solution is already an outlier; if it is, further actions are not considered in this partial solution (lines 5-8). Not all outlier detection algorithms can evaluate an observation with missing features. Our experiments utilized the Isolation Forest algorithm, which, due to its modeling in a tree, can deal with missing values by walking along multiple branches of the trees. The algorithm calculates the

maximum probability of $x + a$ and checks whether this value exceeds the threshold τ established (lines 9-12). If τ has not been exceeded, it is unnecessary to continue the search. Subsequently, the algorithm validates whether the current sample constitutes a solution and, if so, whether it qualifies as a feasible solution (not an outlier). The current solution is added to the collection of obtained solutions upon meeting these criteria (lines 13-17). In cases where none of the previous return commands has stopped the function, the algorithm recursively selects the next feature for evaluation (lines 18-21).

Algorithm 3, called by Algorithm 2, presents the pseudocode for estimating the maximum probability of a partial solution based on the previous presentation. It starts by calculating the model probability $r(x)$ and the feature attributions of x. Then, features are iterated for the decreasing ordering of $\overline{\phi}$ (SHAP maximum feature attributions). If such a feature is not fixed, the maximum probability is increased with the difference $(\overline{\phi}_i - \phi_i)$. A counter j is utilized to consider only k features.

Algorithm 3: Estimate Maximum Probability

Input: Partial solution x, a decision rule $r(\cdot)$, fixed features $\mathcal{D}$, number of maximum allowed changes k, maximum feature attributions $\overline{\phi}$

```
1  r̄ ← r(x);
2  φ ← SHAP feature attributions of x and r;
3  S ← features index i in decreasing order of φ̄;
4  j ← 0;
5  for i ∈ S do
6  |  if i ∈ D then
7  |  |  continue;
8  |  end
9  |  r̄ ← r̄ + (φ̄_i − φ_i);
10 |  j ← j + 1;
11 |  if j ≥ k then
12 |  |  break
13 |  end
14 end
15 return r̄
```

Chapter 5

Material and Methodology

In this section, we will introduce the hardware, software, datasets, classifier algorithms, and outlier detection algorithm utilized in the research. In addition, we will outline the general experimental setup and the specific configurations of the experiments.

5.1 Hardware

In the course of this research, the primary computing device employed was a Dell Vostro 15 laptop whose specifications include an Intel Core i7 processor, 8GB of RAM and 256GB SSD, which collectively ensured efficient processing and storage capabilities.

5.2 Software

The operating system installed on the Dell Vostro 15 was Linux Ubuntu, specifically the latest version available at the time of the research, and the key programming language used in the research was Python.

5.3 Datasets

Addressing the Credit Analysis challenge, experiments were conducted using two widely recognized datasets commonly referenced in literature: German Credit Risk and Taiwan Default of Credit Card Clients.

German Credit Risk

The German Credit Risk[1] dataset (Appendix A - Table A.1) has a total of 1000 rows, and it is composed by two targets, *1: good customers* and *-1: not good customers*, in the *Good Customer* column. In the data sample, 700 rows are related to *good customers* and 300 to *not good customers*. The nature of this data is tabular, counting with 30 representative features, where two of them are categorical.

[1] https://github.com/ustunb/actionable-recourse/tree/master/examples/paper/data/german

Taiwan Default of Credit Card Clients

The Taiwan Default of Credit Card Clients[2] dataset (Appendix B - Table B.1) has a total of 30.000 rows, where the *default.payment* column has two labels, *1: Yes* and *0: No*. The label 0 has a total of 23.364 samples and the label 1 has 6.636 samples. This is also a tabular dataset, which has 25 columns of numerical data.

5.4 Classifier Algorithms

In the experiments, the Credit Risk Assessment Machine Learning Classifiers were constructed by employing two well-established algorithms, LightGBM and MLPClassifier.

LightGBM

The LightGBM[3] (LGBM) algorithm is a high-performance gradient-boosting framework developed by Microsoft, known for its efficiency, scalability, and accuracy [17].

The following LGBM hyperparameters were used in the experiments:

- *n_estimators*
- *learning_rate*
- *max_depth*
- *colsample_bytree*
- *reg_alpha*
- *verbose*
- *random_state*

MLPClassifier

The MLPClassifier[4] algorithm is a neural network that optimizes the log-loss function [4]. The following MLPClassifier hyperparameters were used in the experiments:

- *hidden_layer_sizes*
- *learning_rate_init*
- *epochs*
- *class_weight*
- *batch_size*
- *random_state*

[2]https://www.kaggle.com/datasets/uciml/default-of-credit-card-clients-dataset
[3]https://lightgbm.readthedocs.io/en/latest/pythonapi/lightgbm.LGBMClassifier.html
[4]https://scikit-learn.org/stable/modules/generated/sklearn.neural$_n etwork.MLPClassifier.html$

5.5 Outlier Detection Algorithm

Isolation Forest

The Isolation Forest[5] algorithm returns the anomaly score of each sample requested [18]. The following Isolation Forest hyperparameters were used in the experiments:

- *ndim*
- *sample_size*
- *sample_size*
- *max_depth*
- *ntrees*
- *missing_action*
- *contamination*

5.6 General Setup

A set of experiments to evaluate MAPOFCEM were performed in different settings and compared with baselines and competitors' techniques.

Data Pre-Processing

For the German Credit Risk dataset, the Gender variable was changed to *is_male*, a binary variable that checks whether the user is $1 - Male$ or $0 -$ *Not Male*. The variable *Purpose Of Loan* was excluded due to the inability to categorize it due to the wide range of possibilities it has. For simplicity, the target was inverted, so $Good - 1$ represents that the credit is lent and $Bad - 0$ that the individual is default.

For the Taiwan Default of Credit Card Clients dataset, the features that start with "BILL_AMT", "PAY_AMT" and "LIMIT_BAL" have their values converted from NTD to USD (1 USD - 32.75 NTD). Also, the target *default.payment* was inverted to *No Default*, so $No - 1$ represents that the credit is lent and $Yes - 0$ that the individual is default. The *MARRIAGE* feature was categorized by $1 - Married$, $2 - Single$ and $3 - Other$. Also, the feature *AGE* was categorized by < 25, *from* 25 *to* 39, *from* 40 *to* 59 and $>= 60$. The categories of the feature *EDUCATION* were inverted to $Graduate - 3$, $University - 2$ and $HighSchool - 1$. Also, new features were created, such as: *MaxBillAmountOverLast6Months*, *MaxPaymentAmountOverLast6Months*, *MonthsWithZeroBalanceOverLast6Months*, *MonthsWithLowBalanceOverLast6Months*, *MonthsWithHighBalanceOverLast6Months*, *MostRecentBillAmount*, *MostRecentPaymentAmount*, *TotalOverdueCounts*, *TotalMonthsOverdue*, *HistoryOfOverduePayments*.

[5]https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.IsolationForest.html

Classifier Model Training

To train classifiers with each data set, the data sets were divided into 40% for training, 50% for testing and 10% for validation. The hyperparameters described in the previous section were tuned to obtain the best performance of each dataset and algorithm. The hyperparams optimization of the LGBM model was performed with the validation set using Optuna[6].

Outlier Detection Model Training

Furthermore, two outlier detection algorithms were trained. The first one was trained with the training subset, and the second one was trained on the testing data for evaluation. Using two outlier detection models is important as MAPOFCEM utilizes one trained on the training data subset. To prevent data leaking, counterfactual explanations will be evaluated with an outlier detection model trained specifically on the test subset.

5.7 Evaluation Metrics

Outlier Detection Percentage (ODP) Metric

Definition 15 (The Outlier Detection Percentage)**.** *ODP is a quantitative metric used to evaluate the effectiveness of different algorithms in detecting outliers within a dataset. It measures the proportion of outliers identified by an algorithm relative to the total number of solutions generated by that algorithm.*

Let $N_{outliers}$ be the number of outliers identified by the algorithm, and $N_{total\_solutions}$ be the total number of solutions generated by the algorithm. The Outlier Detection Percentage (ODP) is calculated using the formula:

$$ODP = \left(\frac{N_{outliers}}{N_{total_solutions}} \right) \times 100$$

Score Samples Metric

Definition 16 (Score Samples)**.** *The Score Samples metric is a measure used to evaluate the anomalousness of data points within a dataset using the Isolation Forest algorithm. It provides a score indicating how isolated a data point is relative to others in the dataset, with higher scores suggesting greater anomalies.*

Let $PathLength(x)$ be the average path length for a data point x across all trees in the Isolation Forest. $H(n)$ be the average path length of unsuccessful searches in Binary Search Trees, which is approximately $2 \cdot \log(n-1) + 0.5772156649$ (Euler's constant), where n is the number of samples in the dataset. The anomaly score for a data point x, denoted as $Score(x)$, is calculated using the formula:

$$Score(x) = 2^{-\frac{PathLength(x)}{H(n)}}$$

[6]https://optuna.readthedocs.io/en/stable/

Where:

- A score close to 1 indicates that the data point is an anomaly (more isolated).
- A score close to 0 indicates that the data point is not an anomaly (less isolated).

Execution Time (ET) Metric

Definition 17 (Execution Time)**.** *ET metric measures the duration required by a counterfactual search algorithm to complete its task of exploring alternative scenarios.*

Let: T_{start} be the timestamp when the counterfactual search algorithm begins its execution, and T_{end} be the timestamp when the algorithm completes its task. The Execution Time (ET) is calculated using the formula:

$$\text{ET} = T_{end} - T_{start}$$

Where ET is expressed in seconds and represents the total duration taken by the algorithm to perform its task.

Cost Metric

Definition 18 (Cost)**.** *The Cost metric quantifies the distance between the solutions obtained and the original data point in terms of a percentile. This normalized measure provides an interpretable assessment of how far the solutions deviate from the original data point in the feature space.*

Let $\mathbf{x}_{origin}$ be the origin data point in the feature space, and $\mathbf{x}_{solution}$ be the solution data point obtained by the algorithm. $d(\mathbf{x}_{origin}, \mathbf{x}_{solution})$ be the distance metric used to quantify the difference between the original and solution data points. This could be a Euclidean distance, Manhattan distance, or any other suitable distance measure. The Cost metric is calculated using the formula:

$$Cost = \frac{d(\mathbf{x}_{origin}, \mathbf{x}_{solution})}{P_d} \times 100$$

Where P_d is a percentile threshold or normalization factor that represents the typical range or spread of distances in the dataset.

Number of Changes (NC) Metric

Definition 19 (Number of Changes)**.** *The NC metric counts the number of feature modifications proposed by an algorithm to achieve a counterfactual solution. It serves as a measure of the effort or complexity involved in transitioning from the original instance to the counterfactual.*

Let: $\mathbf{x}_{origin} = (x_1, x_2, \ldots, x_n)$ be the origin data point with n features. $\mathbf{x}_{solution} = (x'_1, x'_2, \ldots, x'_n)$ be the counterfactual solution data point.

The Number of Changes (NC) is calculated using the formula:

$$NC = \sum_{i=1}^{n} \delta(x_i, x'_i)$$

Where $\delta(x_i, x'_i)$ is a function that returns 1 if the feature x_i in the origin data point is different from the feature x'_i in the solution, and 0 otherwise. It can be expressed as:

$$\delta(x_i, x'_i) = \begin{cases} 1, & if x_i \neq x'_i \ 0, \\ if x_i = x'_i \end{cases}$$

5.8 Experiments Setup

Baselines Comparison

In the experiments, the following open source strategies that seek counterfactual explanations were used as benchmarking: NICE, DiCE, MAPOCAM and BruteForce. Related works in the literary review did not have codes available for use in the experiments.

Experiment I - Outlier Detection 2D View

- **Goal**: The Experiment I was conducted in a 2D feature space to illustrate the impact of MAPOFCEM on the feasibility of solutions compared to baselines.
- **Dataset**: German - Features: *LoanAmount* and *Age*.
- **Algorithm**: LGBM Classifier.
- **Baselines**: NICE, DiCE, MAPOCAM and BruteForce.
- **Methodology**: Two Isolation Forest algorithms were trained using *contamination* : 0.05. The first was trained using training data, and it was used inside MAPOFCEM's outlier detection function, while the second was trained using test data to identify outliers in solutions from each baseline methodology.
- **Metric**: Visual Evaluation.

Experiment II - Outlier Detection Percentage

- **Goal**: The Experiment II was conducted to show the percentage of outliers detected for each baseline compared to MAPOFCEM strategy.
- **Datasets**: German and Taiwan.
- **Algorithm**: LGBM and MLP Classifiers.
- **Baselines**: NICE, DiCE, MAPOCAM and BruteForce.

- **Methodology**: This experiment also uses two Isolation Forest algorithms, one trained with training data to be used within the MAPOFCEM outlier detection function and other trained with test data to check outliers in the solutions proposed by the baselines. The *contamination* hyperparameter used by the Isolation Forest algorithms was set as 0.05 to German dataset and 0.01 to Taiwan dataset. The percentage of outliers was calculated for the total number of solutions found by each baseline. The simulations were performed to search for counterfactuals in 50 samples and the average results found were listed in tables with the maximum range of variation found.
- **Metric**: Outlier Detection Percentage (ODP) and Score Samples.

Experiment III - Time, Cost and Number of Changes

- **Goal**: The Experiment III was conducted to show the time and cost calculated for each baseline to obtain solutions compared to MAPOFCEM, and the number of proposed changes considered.
- **Datasets**: German and Taiwan.
- **Algorithm**: LGBM and MLP Classifiers.
- **Baselines**: NICE, DiCE, MAPOCAM and BruteForce.
- **Methodology**: The simulations were performed to search for counterfactuals in 50 samples and the average results found were listed in tables with the maximum range of variation found for execution time in seconds, cost and number of changes.
- **Metric**: Execution Time (ET), Cost and Number of Changes (NC).

Experiment IV - Multi-objective Analysis

- **Goal**: The Experiment IV was conducted with the aim of verifying the impact of using multi-objectives compared to Experiments II and III.
- **Datasets**: German.
- **Algorithm**: LGBM and MLP Classifiers.
- **Baselines**: NICE, DiCE, MAPOCAM and BruteForce.
- **Methodology**: The same of Experiment II and Experiment III.
- **Metric**: Outlier Detection Percentage (ODP) and Score Samples, Execution Time (ET), Cost and Number of Changes (NC).

Experiment V - Analysis of Contamination Hyperparameter

- **Goal**: The Experiment V was conducted to show the impact of varying the *contamination* hyperparameter of the Isolation Forest algorithm used in MAPOFCEM.
- **Datasets**: German and Taiwan.
- **Algorithm**: LGBM Classifier.
- **Baselines**: NICE, DiCE, MAPOCAM and BruteForce.
- **Methodology**: The simulations were performed to search for counterfactuals in 50 samples and the average results found were listed in tables with the maximum range of variation found, using 5 *contamination* hyperparameters values: 0.005, 0.01, 0.05, 0.075 and 0.1.
- **Metric**: Execution Time (ET), Cost, Outlier Detection Percentage (ODP) and Score Samples.

Chapter 6

Results and Discussions

6.1 Experiment I - Outlier Detection 2D View

In this experiment, each strategy presents a 2D visualization in which the individual who wants an explanation is represented by a black dot, individuals with accepted credit are green, and individuals with rejected credit are red. The strategy identifies orange elements as counterfactual solutions, and the individuals represented with an "x" are outliers found by the Isolation Forest trained with the test data.

NICE. Figure 6.1 illustrates the results of Experiment I for the NICE algorithm. NICE (Nearest Instance Counterfactual Explanations) is a cutting-edge algorithm developed by Brughmans et al. (2023) [6] that typically offers a single solution for each selected individual. In this case, the counterfactual solution identified by NICE was flagged as an outlier despite the proximity of the identified counterfactual to the individual, as outliers detected by Isolation Forest encompass its neighborhood.

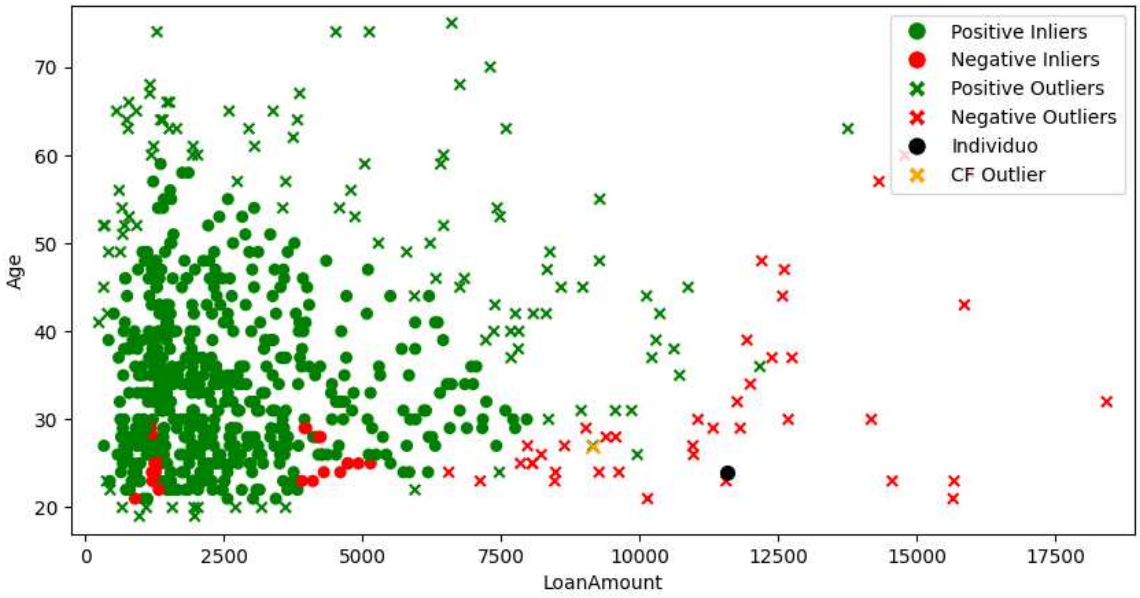

Figure 6.1: NICE - Experiment I Results.

NICE uses the nearest-neighbor strategy to find solutions closest to the individual. For this particular example, the only solution found was detected as impractical by the Isolation Forest, which could have been classified as a feasible solution if the contamination hyperparameter used in the experiment was more flexible. Experiment IV will showcase how the variation of this hyperparameter can impact the solutions found.

DiCE. Figure 6.2 illustrates the results of Experiment I for the DiCE algorithm. Diverse Counterfactual Explanations (DiCE) is a framework introduced by Mothilal et al. (2020) [22] that presents an approach that provides multiple solutions within the feature space for each selected individual. In this case, three solutions were found and two were identified as outliers by the Isolation Forest algorithm. For these solutions identified as outliers, it is possible to verify whether both have a significant distance from the data distribution while the inlier solution found is on the edge of the data distribution.

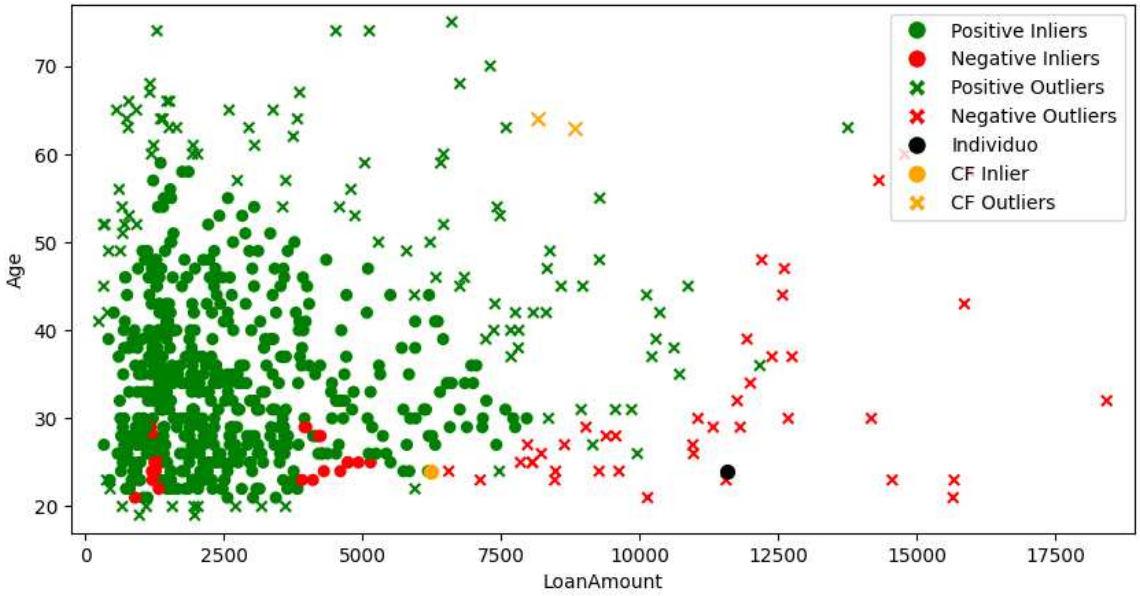

Figure 6.2: DiCE - Experiment I results.

DiCE uses the determinant of the kernel matrix to guarantee the diversity of counterfactuals, but this experiment showed that this strategy also presents unfeasible solutions.

MAPOCAM. Figure 6.3 illustrates the results of Experiment I for the MAPOCAM algorithm, which uses multiobjective analysis based on Pareto-optimal solutions. In this case, MAPOCAM found 3 edge counterfactuals, where Isolation Forest recognized them as outliers. As MAPOCAM uses branch-and-bound techniques that use the number of changes as a criterion for stopping the search, the algorithm ended the search after identifying 3 changes for the individual. This meant that potential solutions were not found within the data distribution, which did not achieve the feasibility of its solutions.

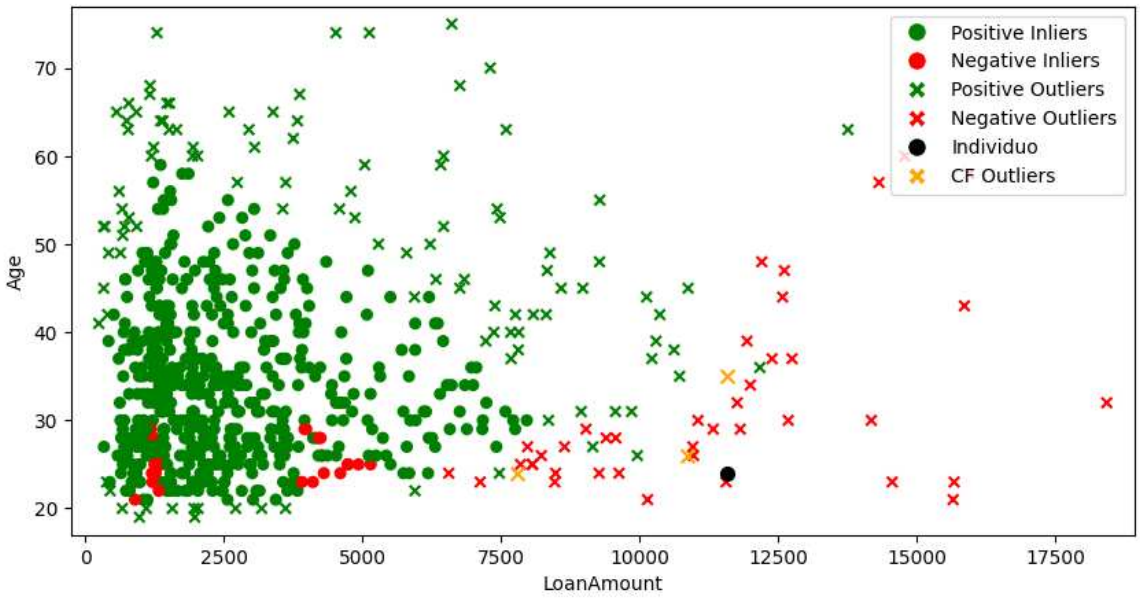

Figure 6.3: MAPOCAM - Experiment I Results.

BruteForce. Figure 6.4 illustrates the results of Experiment I for the BruteForce algorithm. BF operates by iterating possible changes to selected features and evaluating the impact of these changes on the outcome of interest. The strategy adopted by this algorithm calculates all possible solutions during the counterfactual search. The result found for this experiment is very similar to the result obtained by MAPOCAM but with a processing time much higher than that of MAPOCAM, which uses cuts to optimize the counterfactual search.

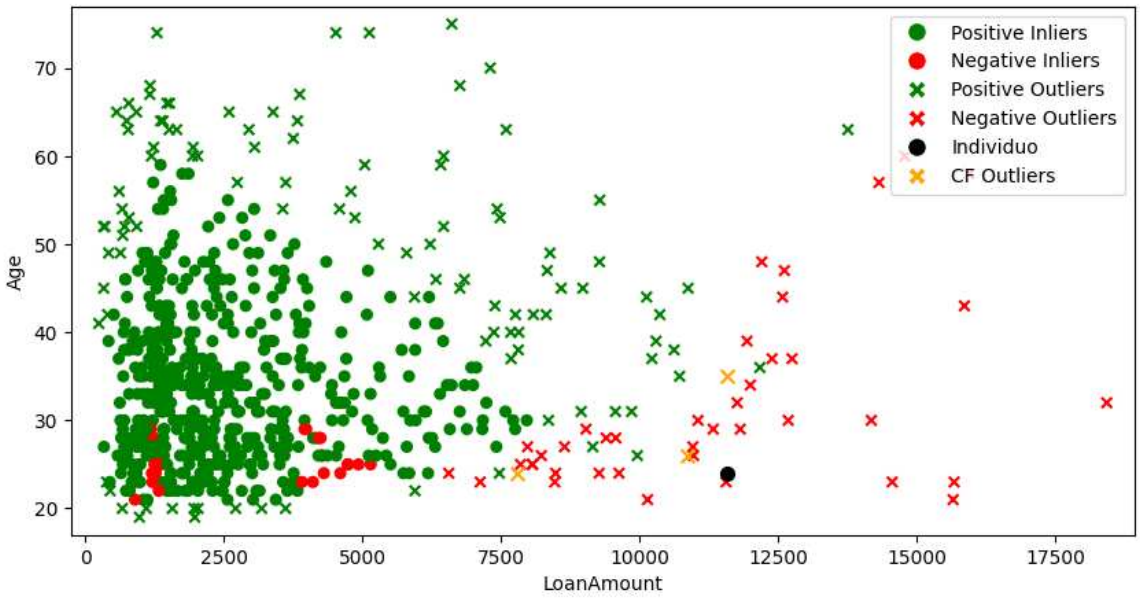

Figure 6.4: BruteForce - Experiment I Results.

MAPOFCEM. Figure 6.5 illustrates the results of Experiment I for the MAPOFCEM algorithm, where one inlier counterfactual was found at the edge of the data distribution. Analyzing this result, the breach-and-bound strategy used by MAPOCAM was employed by MAPOFCEM to find the optimal solution. However, the search was not interrupted after MAPOFCEM's internal strategy that uses the Isolation Forest trained with the training data identified that the first counterfactual search solutions were unfeasible. This is the main difference between the results of MAPOCAM and MAPOFCEM in this experiment. The feasibility of the solutions found has a significant impact on the strategies used by the algorithms. It is also possible to note that although the maximum number of solutions is up to 3 possible solutions, MAPOFCEM considered only 1, while MAPOCAM presented 3 solutions. This is also because this variable is part of MAPOFCEM's maximum probability function, where pruning will restrict the range of other options considered in MAPOCAM.

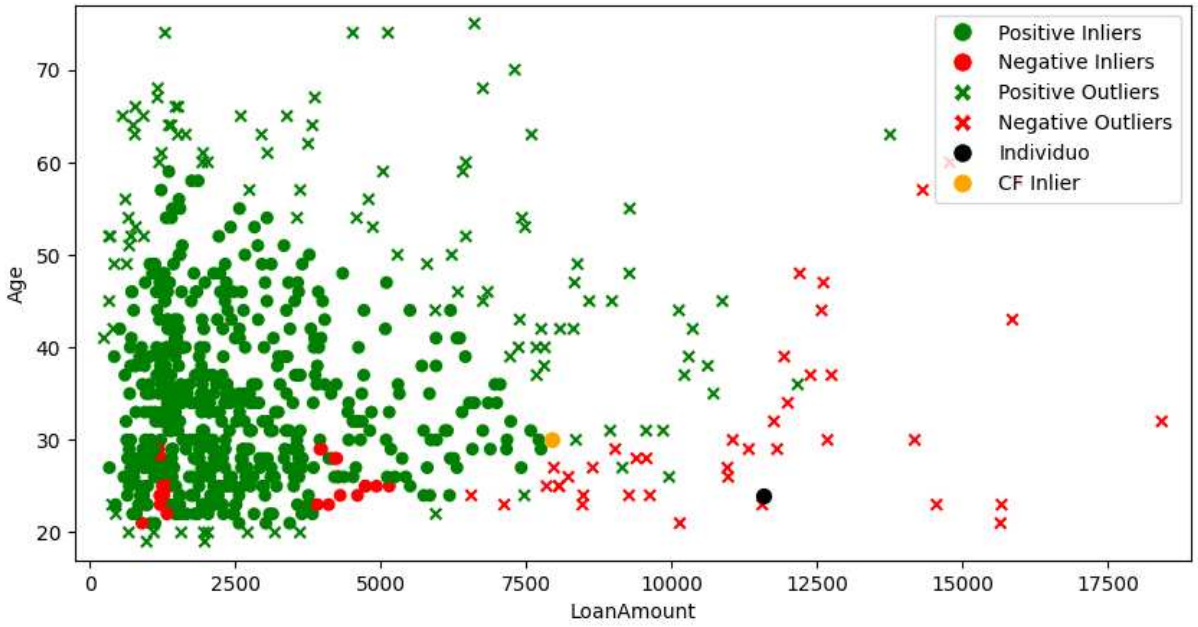

Figure 6.5: MAPOFCEM - Experiment I Results.

Although MAPOFCEM found an inlier individual in this particular case, it could also have found outlier individuals due to the detection being done by an Isolation Forest algorithm that uses the test data. In contrast, the Isolation Forest used by MAPOFCEM uses the training data. This variation in cases will depend on how much the training data distribution is different from the distribution of the test data.

In summary, the results obtained in this experiment illustrated the impact of considering the feasibility of the solutions proposed for users. In the real world, algorithms are trained with dozens or hundreds of features, increasing this 2D feature space even more. Therefore, Experiment II was conducted to illustrate the scope of these impacts in a larger feature space and with different databases.

6.2 Experiment II - Outlier Detection Percentage

In this experiment, each strategy presents outlier percentage values and score samples, which is the metric used by Isolation Forest to determine the probability of a sample being an outlier. This parameter is calculated based on the number of partitions needed to isolate a sample. Samples isolated quickly (with few partitions) are considered anomalous, while those requiring more partitions are seen as normal. A threshold of 0.5 is commonly used to differentiate between normal and anomalous samples, as the algorithm assumes that half of the data is normal and the other half may contain anomalies. In this scenario, individuals with a score above 0.5 are classified as outliers. The following tables present the average values found in the search for counterfactuals from 50 samples.

6.2.1 LGBM

The Table 6.1 presents the results found by using the LBGM Classifier.

Table 6.1: Percentage of outliers detected by each strategy using LGBM classifier

Datasets	**German**		**Taiwan**	
Algorithm	**Outliers**	**Score Samples**	**Outliers**	**Score Samples**
NICE	12%	0.481	6%	0.457
DiCE	14%	0.502	20%	0.504
MAPOCAM	22%	0.521	6%	0.454
BruteForce	20%	0.523	6%	0.461
MAPOFCEM	6%	0.509	4%	0.450

Upon analyzing the results for the German dataset, it is possible to see that the MAPOFCEM strategy of using an Isolation Forest trained on the training data achieved the best outlier percentage ratio of just 6%. In contrast, the MAPOCAM strategy exhibited the highest outlier percentage of 22%. This disparity arises because MAPOCAM employs a strategy that identifies Pareto optimal values without considering the underlying data distribution.

Similarly, when analyzing the results for the Taiwan dataset, it is possible to see that the strategy used by MAPOFCEM achieved the lowest percentage of outliers. Meanwhile, NICE, MAPOCAM, and BruteForce exhibited comparable outlier percentages, also closely aligning with MAPOFCEM. A noticeable general decrease in the percentage of outliers is observed compared with the German dataset. This is attributed to the different *contamination* values used for the German (0.05) and Taiwan (0.01) datasets. The lower *contamination* index for the Taiwan dataset is justified by the optimal performance of MAPOFCEM with this value in Experiment V. This Experiment underscores the significant impact of the *contamination* hyperparameter across these datasets.

Another noteworthy aspect is that, ideally, the strategy employed by MAPOFCEM should yield outlier percentages very close to 0%. However, the reason these values are higher is due to the Isolation Forest used in the counterfactual search, which considers the distribution of the training data, whereas the experiment relies on the distribution of the

test data. The degree of divergence between these distributions influences the number of outliers identified by the MAPOFCEM strategy.

6.2.2 MLP

Table 6.2: Percentage of outliers detected by each strategy using MLP classifier

Datasets	**German**		**Taiwan**	
Algorithm	**Outliers**	**Score Samples**	**Outliers**	**Score Samples**
NICE	4%	0.468	0%	0.441
DiCE	20%	0.519	4%	0.483
MAPOCAM	24%	0.524	2%	0.437
BruteForce	24%	0.526	4%	0.462
MAPOFCEM	15%	0.514	2%	0.437

Table 6.2 presents the results found using the MLP Classifier. The results obtained for the German dataset showed that the NICE strategy achieved the best percentage of detected outliers with just 4%, while MAPOFCEM, the second best, presented a percentage of 15%. It is interesting to note the impact of changing the type of classifier on MAPOFCEM's performance. However, in general, the other algorithms also present higher percentages of outliers than with LGBM, highlighting the advantage of NICE when used with MLP models. For the Taiwan dataset, the *contamination* index of 0.01 significantly contributed to reducing outlier identification across all strategies. However, NICE identified no outliers, and MAPOFCEM exhibited the same outlier percentage as MAPOCAM.

6.3 Experiment III - Time, Cost and Number of Changes

In this experiment, each strategy presents values for time, cost and number of proposed changes. The cost measure was proposed to represent the distance, in percentile, of the solutions obtained from the individual. The following tables present the average values found in the search for counterfactuals from 50 samples.

6.3.1 LGBM

Table 6.3: Time, cost and number of changes for each strategy using LGBM classifier

Datasets	**German**			**Taiwan**		
Algorithm	**Time**	**Cost**	**Changes**	**Time**	**Cost**	**Changes**
NICE	0.01 (± 0.00)	0.385	1.42	0.02 (± 0.00)	0.238	2.22
DiCE	0.20 (± 0.04)	0.520	1.72	0.24 (± 0.01)	0.608	1.64
MAPOCAM	0.46 (± 1.26)	0.065	2.14	7.66 (± 21.23)	0.054	2.56
BruteForce	46.92 (± 1.30)	0.064	1.56	313.27 (± 22.82)	0.322	1.64
MAPOFCEM	0.36 (± 0.60)	0.099	2.08	4.37 (± 25.37)	0.065	2.49

Table 6.3 presents the results found by using the LGBM Classifier. Evaluating the first variable, time (seconds), for the German dataset showed that NICE and DiCE exhibited the best computational times, while BruteForce showed the worst. This is because the counterfactual search employed by NICE and DiCE does not require exhaustive mining of the entire feature space to find a solution. However, the solutions identified by these methods may not align closely with the data distribution, indicating a necessary trade-off between search time and the feasibility of the results. Additionally, MAPOFCEM demonstrated better computational times than MAPOCAM, despite utilizing a SHAP-based feature importance strategy, which is typically more computationally expensive. This improvement is due to the adaptation of the maximum probability calculation, which was modified to optimize the counterfactual search further. For the Taiwan dataset, it is possible to see that the computational times for MAPOFCEM and MAPOCAM significantly increase. This can be attributed to the complexity of this dataset, which contains a substantially more significant number of samples than the German dataset.

MAPOCAM and MAPOFCEM strategies demonstrated the best average costs for the German and Taiwan datasets, with values of 0.065 and 0.099 for the German dataset and 0.054 and 0.065 for the Taiwan dataset, respectively. In other words, these strategies yield solutions closest to the individual in terms of percentile. This is because these strategies perform a thorough grid search to find optimal solutions for individuals, aiming for the lowest implementation cost. Additionally, it can be observed that MAPOCAM achieves lower costs compared to MAPOFCEM. This difference is related to the Outlier Detection strategy used in MAPOFCEM, which discards samples closer to the individual but distant from the data distribution.

The other variable considered is the number of changes proposed by each algorithm. DiCE and BruteForce demonstrated favorable results for both datasets, proposing fewer changes across more variables than the other strategies. Notably, MAPOCAM presents more changes than MAPOFCEM for both datasets. This difference arises from one of the adaptations made to the MAPOFCEM maximum probability function, which incorporates the number of changes.

6.3.2 MLP

The Table 6.4 presents the results found by using the MLP Classifier.

Table 6.4: Time, cost and number of changes for each strategy using the MLP classifier

Datasets	**German**			**Taiwan**		
Algorithm	**Time**	**Cost**	**Changes**	**Time**	**Cost**	**Changes**
NICE	0.01 (± 0.00)	0.486	2.10	0.02 (± 0.01)	0.510	5.48
DiCE	0.21 (± 0.04)	0.460	2.04	0.27 (± 0.18)	0.662	1.48
MAPOCAM	1.44 (± 6.38)	0.082	1.78	121.68 (± 275.44)	0.202	2.96
BruteForce	57.60 (± 0.83)	0.082	1.42	381.81 (± 32.81)	0.674	2.06
MAPOFCEM	0.41 (± 0.63)	0.081	1.73	60.23 (± 60.41)	0.204	2.96

With MLP, it is possible to see that the computational time for both MAPOCAM and MAPOFCEM increases considerably for the Taiwan dataset. MAPOCAM's time is

twice that of MAPOFCEM. This occurs due to the principle of monotonicity employed by MAPOCAM, which does not constrain the counterfactual search for black-box algorithms. Despite this, both strategies continue to offer low-cost solutions that are closely aligned with the individuals in both datasets. Meanwhile, NICE and DiCE are the strategies that require the most effort from the user. The DiCE and BruteForce strategies remain the most stable regarding the number of changes. For the Taiwan dataset, the number of changes proposed by NICE increased significantly.

6.4 Experiment IV - Multi-objective Analysis

Each strategy presents outlier percentage values, score samples, time, cost, changes, and many counterfactual solutions found (CFs) in this experiment. The following tables present the average values found in the search for counterfactuals from 50 samples. Only strategies that present multiple solutions were considered, and MAPOCAM2 and MAPOFCEM2 are versions of MAPOCAM and MAPOFCEM that present multi-objective analysis, respectively.

6.4.1 LGBM

Table 6.5: Multi-objective results for LGBM algorithm in German Dataset

Algorithm	Outliers	Score Samples	Time	Cost	Changes	CFs
DiCE	19%	0.508	0.30 (± 0.07)	0.508	1.77	22
MAPOCAM	19%	0.511	0.93 (± 2.67)	0.112	1.29	1
MAPOCAM2	21%	0.514	15.14 (± 13.88)	0.436	2.09	36
MAPOFCEM	1%	0.496	0.51 (± 0.77)	0.150	1.36	1
MAPOFCEM2	7%	0.503	2.12 (± 1.92)	0.434	2.05	23

Table 6.5 presents the results found using the LGBM Classifier. It can be observed in Table 6.5 that MAPOCAM2's computational time grows exponentially compared to MAPOCAM when employing the multi-objective strategy, whereas MAPOFCEM2 demonstrates control over this effect. This is due to the set of new innovations adopted, specifically the changes within the calculation of the maximum probability function. Furthermore, the implementation of the outlier detection approach significantly contributes to the results, as MAPOCAM2 presented approximately 36 changes, whereas MAPOFCEM2 presented only 23 changes, and MAPOFCEM2 presented 14% fewer outliers compared to MAPOCAM2.

6.4.2 MLP

Table 6.6 presents the results found using the MLP Classifier. From Table 6.6, it can be seen that unlike LGBM, with MLP, the computational time of MAPOFCEM2 grows significantly, similar to MAPOCAM2, but to a lesser extent than MAPOCAM2. Additionally, it is interesting to note that the percentage of outliers decreased for both the MAPOFCEM2

Table 6.6: Multi-objective results for MLP algorithm in German Dataset

Algorithm	Outliers	Score Samples	Time	Cost	Changes	CFs
DiCE	18%	0.509	0.42 (± 0.11)	0.516	1.89	41
MAPOCAM	23%	0.524	2.00 (± 7.00)	0.087	1.37	1
MAPOCAM2	18%	0.509	38.97 (± 20.56)	0.514	2.29	47
MAPOFCEM	14%	0.514	1.94 (± 3.71)	0.094	1.38	1
MAPOFCEM2	6%	0.499	25.89 (± 15.41)	0.517	2.30	40

and MAPOCAM2 multi-objective strategies. However, the MAPOFCEM2 strategy still showed the best performance with just 6%.

6.5 Experiment V - Analysis of Contamination Hyperparameter

In this experiment, each strategy presents values for time, cost, outlier percentage values, score samples and the index of *contamination*, which represents the Isolation Forest's sensitivity to anomaly detection. The following tables present the average values found in the search for counterfactuals from 50 samples.

6.5.1 German

Table 6.7: Contamination Hyperparameter results for German Dataset

Algorithm	Time	Cost	Outliers	Score Samples	Contamination
	0.19 (± 0.25)	0.077	18%	0.519	0.0
	0.27 (± 0.35)	0.091	20%	0.517	0.005
MAPOFCEM	0.28 (± 0.37)	0.095	12%	0.511	0.01
	0.35 (± 0.51)	0.105	4%	0.507	0.05
	0.60 (± 1.46)	0.136	0%	0.499	0.075
	0.59 (± 1.41)	0.140	0%	0.493	0.1

Table 6.7 presents the results found for the German dataset. It shows that the computational time also rises as the *contamination* index increases from 0.0 to 0.1. This occurs because MAPOFCEM must consider more samples during the counterfactual search, as those closest to the individual may be outliers. Consequently, the cost of solutions also increases for the same reason.

Furthermore, it is observable that as the *contamination* index increases, the percentage of outliers identified by MAPOFCEM decreases, eventually reaching 0%. This occurs because MAPOFCEM becomes progressively less flexible towards values at the edge of the distribution, increasingly focusing on the core of the distribution. Therefore, it is crucial to study the flexibility of the *contamination* hyperparameter used, as an excessively high value may unnecessarily disregard feasible samples. Because of this, in previous experiments the value of 0.05 was used for German dataset, which is the default value

of this hyperparameter, as it does not increase the computational time considerably, but manages to establish a percentage of 4% outliers found.

6.5.2 Taiwan

Table 6.8: Contamination Hyperparameter results for Taiwan Dataset

Algorithm	Time	Cost	Outliers	Score Samples	Contamination
	0.72 (± 0.87)	0.062	6%	0.453	0.0
	4.58 (± 24.91)	0.059	6%	0.454	0.005
MAPOFCEM	4.63 (± 25.37)	0.060	4%	0.450	0.01
	7.08 (± 30.62)	0.064	4%	0.449	0.05
	9.50 (± 34.45)	0.065	0%	0.446	0.075
	13.41 (± 42.90)	0.073	0%	0.444	0.1

Table 6.8 presents the results found for the Taiwan dataset. It can be observed that the computational time grows significanlty compared to the German experiment, as the Taiwan dataset exhibits greater complexity due to the higher number of features present. Due to the increase observed between the indices 0.01 and 0.05, the value 0.01 was selected for the *contamination* index in the previous experiments. It was also observed that the cost of the solutions found remains relatively more stable compared to the German dataset and that the percentage of outliers found for this database is slightly lower than in German's experiment.

6.6 Discussion

The findings of this thesis demonstrate that integrating counterfactual explanations with an outlier detection mechanism considerably improves the feasibility and usability of these explanations for end users. The experimental results underscore the superiority of the proposed method over existing techniques in producing feasible counterfactual explanations. As illustrated in Figures 6.1 to 6.5 and detailed in Tables 6.1, 6.5, and 6.6, the strategy employed by MAPOFCEM consistently results in a lower percentage of outliers compared to other methods. This is true across both single and multi-objective verification scenarios, highlighting MAPOFCEM's effectiveness in maintaining solution integrity.

Furthermore, Tables 6.3, 6.4, 6.5, and 6.6 illustrate that the MAPOFCEM approach significantly improved execution time compared to MAPOCAM by implementing innovative bound strategies. These strategies are designed to be more agnostic, making them particularly effective for complex models. Despite these improvements in execution time, MAPOFCEM maintained the proximity of the generated solutions to the origin data points, ensuring that the quality and relevance of the counterfactuals were not compromised. This balance between efficiency and solution proximity highlights the versatility of MAPOFCEM in handling intricate modeling scenarios.

Chapter 7

Conclusion and Next Steps

In this work, the framework Model-Agnostic Pareto-Optimal Feasible Counterfactual Explanations Mining (MAPOFCEM) was presented. Aspects related to the feasibility of counterfactual solutions found by counterfactual search algorithms were investigated. To this end, a strategy based on Outlier Detection was proposed, utilizing the Isolation Forest algorithm trained with the training dataset to refine the feasibility of the solutions proposed. Additionally, the base algorithm of this study, Model-Agnostic Pareto-Optimal Counterfactual Antecedent Mining (MAPOCAM), was optimized to consider agnostic models such as LGBM and MLP, along with updating the maximum probability calculation function using the number of changes and a limit based on feature importance with SHAP values. It was found that MAPOFCEM presents a lower percentage of identified outliers compared to open-source algorithms in the literature, and a lower cost of implementing the proposed solutions, despite higher computational time. Moreover, MAPOFCEM demonstrated the following advantages over the MAPOCAM strategy: solutions closer to data distribution and shorter computing time. Furthermore, the importance of considering the *contamination* index of the Isolation Forest algorithm for the performance of MAPOFCEM was analyzed. This hyperparameter needs to be considered and evaluated when applying outlier detection strategies in the counterfactual search. The results presented in this work contribute to a better understanding of the feasibility of counterfactual solutions proposed, aiming to enhance the adoption of this type of tool by users with solutions that are more applicable and consistent with the real world.

We found throughout this research that exploring causality in counterfactual explanations presents a promising avenue for future research. Current studies increasingly focus on understanding causal relationships between variables to provide more meaningful and actionable counterfactual explanations. This approach not only improves the interpretability of models, but also ensures that explanations are based on underlying causal mechanisms, thus improving their reliability and usefulness in real-world applications. Despite these opportunities, significant challenges remain, particularly in dealing with categorical data during the counterfactual search process. Categorical data often introduce complexities due to their discrete nature and the lack of inherent order among categories. Proposing a robust method for handling categorical data in counterfactual explanations is crucial for ensuring that the generated explanations are both feasible and interpretable.

Bibliography

[1] Daniel W Apley and Jingyu Zhu. Visualizing the effects of predictor variables in black box supervised learning models. *arXiv preprint arXiv:1612.08468*, 2016.

[2] André Artelt and Barbara Hammer. Convex density constraints for computing plausible counterfactual explanations. In *Artificial Neural Networks and Machine Learning–ICANN 2020: 29th International Conference on Artificial Neural Networks, Bratislava, Slovakia, September 15–18, 2020, Proceedings, Part I 29*, pages 353–365. Springer, 2020.

[3] Robert Bartlett, Adair Morse, Richard Stanton, and Nancy Wallace. Consumer-lending discrimination in the fintech era. *Journal of Financial Economics*, 143(1):30–56, 2022.

[4] Christopher M Bishop. *Neural networks for pattern recognition*. Oxford university press, 1995.

[5] Leo Breiman. Random forests. *Machine learning*, 45:5–32, 2001.

[6] Dieter Brughmans, Pieter Leyman, and David Martens. Nice: an algorithm for nearest instance counterfactual explanations. *Data mining and knowledge discovery*, pages 1–39, 2023.

[7] Susanne Dandl, Christoph Molnar, Martin Binder, and Bernd Bischl. Multi-objective counterfactual explanations. In *Parallel Problem Solving from Nature–PPSN XVI: 16th International Conference, PPSN 2020, Leiden, The Netherlands, September 5-9, 2020, Proceedings, Part I*, pages 448–469. Springer, 2020.

[8] Aaron Fisher, Cynthia Rudin, and Francesca Dominici. All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously. *J. Mach. Learn. Res.*, 20(177):1–81, 2019.

[9] Maximilian Förster, Philipp Hühn, Mathias Klier, and Kilian Kluge. Capturing users' reality: A novel approach to generate coherent counterfactual explanations. 2021.

[10] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

[11] Jerome H Friedman and Bogdan E Popescu. Predictive learning via rule ensembles. *The annals of applied statistics*, pages 916–954, 2008.

[12] Andreas Fuster, Paul Goldsmith-Pinkham, Tarun Ramadorai, and Ansgar Walther. Predictably unequal? the effects of machine learning on credit markets. *The Journal of Finance*, 77(1):5–47, 2022.

[13] Leilani H Gilpin, David Bau, Ben Z Yuan, Ayesha Bajwa, Michael Specter, and Lalana Kagal. Explaining explanations: An overview of interpretability of machine learning. In *2018 IEEE 5th International Conference on data science and advanced analytics (DSAA)*, pages 80–89. IEEE, 2018.

[14] Alex Goldstein, Adam Kapelner, Justin Bleich, and Emil Pitkin. Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation. *journal of Computational and Graphical Statistics*, 24(1):44–65, 2015.

[15] Cheng-Lung Huang, Mu-Chen Chen, and Chieh-Jen Wang. Credit scoring with a data mining approach based on support vector machines. *Expert systems with applications*, 33(4):847–856, 2007.

[16] Kentaro Kanamori, Takuya Takagi, Ken Kobayashi, and Hiroki Arimura. Dace: Distribution-aware counterfactual explanation by mixed-integer linear optimization. In *IJCAI*, pages 2855–2862, 2020.

[17] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30, 2017.

[18] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *2008 eighth ieee international conference on data mining*, pages 413–422. IEEE, 2008.

[19] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

[20] Divyat Mahajan, Chenhao Tan, and Amit Sharma. Preserving causal constraints in counterfactual explanations for machine learning classifiers. *arXiv preprint arXiv:1912.03277*, 2019.

[21] Christoph Molnar. A guide for making black box models explainable. *URL: https://christophm. github. io/interpretable-ml-book*, page 3, 2018.

[22] 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*, pages 607–617, 2020.

[23] W James Murdoch, Chandan Singh, Karl Kumbier, Reza Abbasi-Asl, and Bin Yu. Definitions, methods, and applications in interpretable machine learning. *Proceedings of the National Academy of Sciences*, 116(44):22071–22080, 2019.

[24] Daniel Nemirovsky, Nicolas Thiebaut, Ye Xu, and Abhishek Gupta. Countergan: Generating counterfactuals for real-time recourse and interpretability using residual gans. In *Uncertainty in Artificial Intelligence*, pages 1488–1497. PMLR, 2022.

[25] Axel Parmentier and Thibaut Vidal. Optimal counterfactual explanations in tree ensembles. In *International Conference on Machine Learning*, pages 8422–8431. PMLR, 2021.

[26] Martin Pawelczyk, Klaus Broelemann, and Gjergji Kasneci. Learning model-agnostic counterfactual explanations for tabular data. In *Proceedings of The Web Conference 2020*, pages 3126–3132, 2020.

[27] Rafael Poyiadzi, Kacper Sokol, Raul Santos-Rodriguez, Tijl De Bie, and Peter Flach. Face: feasible and actionable counterfactual explanations. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 344–350, 2020.

[28] Marcos M Raimundo, Luis Gustavo Nonato, and Jorge Poco. Mining pareto-optimal counterfactual antecedents with a branch-and-bound model-agnostic algorithm. *Data Mining and Knowledge Discovery*, pages 1–33, 2022.

[29] LKPJ Rdusseeun and P Kaufman. Clustering by means of medoids. In *Proceedings of the statistical data analysis based on the L1 norm conference, neuchatel, switzerland*, volume 31, 1987.

[30] 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*, pages 1135–1144, 2016.

[31] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Anchors: High-precision model-agnostic explanations. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[32] Xiaoting Shao and Kristian Kersting. Gradient-based counterfactual explanations using tractable probabilistic models. *arXiv preprint arXiv:2205.07774*, 2022.

[33] Lloyd S Shapley et al. A value for n-person games. 1953.

[34] Alexander Spangher, Berk Ustun, and Yang Liu. Actionable recourse in linear classification. In *Proceedings of the 5th workshop on fairness, accountability and transparency in machine learning*, 2018.

[35] Arnaud Van Looveren and Janis Klaise. Interpretable counterfactual explanations guided by prototypes. In *Machine Learning and Knowledge Discovery in Databases. Research Track: European Conference, ECML PKDD 2021, Bilbao, Spain, September 13–17, 2021, Proceedings, Part II 21*, pages 650–665. Springer, 2021.

[36] Kush R. Varshney. *Trustworthy Machine Learning*. Independently Published, Chappaqua, NY, USA, 2022.

[37] Sahil Verma, Varich Boonsanong, Minh Hoang, Keegan E Hines, John P Dickerson, and Chirag Shah. Counterfactual explanations and algorithmic recourses for machine learning: A review. *arXiv preprint arXiv:2010.10596*, 2020.

[38] Sahil Verma, Keegan Hines, and John P Dickerson. Amortized generation of sequential algorithmic recourses for black-box models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 8512–8519, 2022.

[39] Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harv. JL & Tech.*, 31:841, 2017.

[40] Xintao Xiang and Artem Lenskiy. Realistic counterfactual explanations by learned relations. *arXiv preprint arXiv:2202.07356*, 2022.

[41] Fan Yang, Sahan Suresh Alva, Jiahao Chen, and Xia Hu. Model-based counterfactual synthesizer for interpretation. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pages 1964–1974, 2021.

Appendix A

German Credit Dataset

Table A.1: German Credit Risk Dataset.

Feature	**Possible Values**	**Description**
Gender	$Male$ or $Female$	Sex
ForeignWorker	$Yes = 1$ or $No = 0$	Foreign Worker
Single	$Yes = 1$ or $No = 0$	Single Status
Age		Age
LoanDuration		Loan Duration
PurposeOfLoan	$Education, NewCar, etc.$	Purpose Of Loan
LoanAmount		Loan Amount
LoanRateAsPercentOfIncome		Loan Rate
YearsAtCurrentHome		Years At Current Home
NumberOfOtherLoansAtBank		Other Loans At Bank
NumberOfLiableIndividuals		Liable Individuals
HasTelephone	$Yes = 1$ or $No = 0$	Has Telephone
CheckingAccountBalance_geq_0	$Yes = 1$ or $No = 0$	Account Balance
CheckingAccountBalance_geq_200	$Yes = 1$ or $No = 0$	Account Balance
SavingsAccountBalance_geq_100	$Yes = 1$ or $No = 0$	Account Balance
SavingsAccountBalance_geq_500	$Yes = 1$ or $No = 0$	Account Balance
MissedPayments	$Yes = 1$ or $No = 0$	Missed Payments
NoCurrentLoan	$Yes = 1$ or $No = 0$	No Current Loan
CriticalAccountOrLoansElsewhere	$Yes = 1$ or $No = 0$	Critical Account
OtherLoansAtBank	$Yes = 1$ or $No = 0$	Other Loans
HasCoapplicant	$Yes = 1$ or $No = 0$	Has Coapplicant
HasGuarantor	$Yes = 1$ or $No = 0$	Has Guarantor
OwnsHouse	$Yes = 1$ or $No = 0$	Owns House
RentsHouse	$Yes = 1$ or $No = 0$	Rents House
Unemployed	$Yes = 1$ or $No = 0$	Unemployed
YearsAtCurrentJob_lt_1	$Yes = 1$ or $No = 0$	Years At Current Job
YearsAtCurrentJob_geq_4	$Yes = 1$ or $No = 0$	Years At Current Job
JobClassIsSkilled	$Yes = 1$ or $No = 0$	Job Class Is Skilled
Good Customer	$Good = 1$ or $Bad = -1$	Target

Appendix B

Taiwan Credit Dataset

Table B.1: Taiwan Default of Credit Card Clients Dataset.

Feature	**Possible Values**	**Description**
ID		ID of Each Client
LIMIT_BAL		Amount of Given Credit
SEX	$Male - 1$ or $Female - 2$	Gender
EDUCATION	$Graduate - 1$, $University - 2, etc.$	Educational Level
MARRIAGE	$Married - 1$, $Single - 2, etc.$	Marital Status
AGE		Age in Years
PAY_0	$PayDuly - 1$, $PaymentDelay - 2, etc.$	Repayment Status
PAY_2	$PayDuly - 1$, $PaymentDelay - 2, etc.$	Repayment Status
PAY_3	$PayDuly - 1$, $PaymentDelay - 2, etc.$	Repayment Status
PAY_4	$PayDuly - 1$, $PaymentDelay - 2, etc.$	Repayment Status
PAY_5	$PayDuly - 1$, $PaymentDelay - 2, etc.$	Repayment Status
PAY_6	$PayDuly - 1$, $PaymentDelay - 2, etc.$	Repayment Status
BILL_AMT1		Amount of Bill Statement
BILL_AMT2		Amount of Bill Statement
BILL_AMT3		Amount of Bill Statement
BILL_AMT4		Amount of Bill Statement
BILL_AMT5		Amount of Bill Statement
BILL_AMT6		Amount of Bill Statement
PAY_AMT1		Amount of Previous Payment
PAY_AMT2		Amount of Previous Payment
PAY_AMT3		Amount of Previous Payment
PAY_AMT4		Amount of Previous Payment
PAY_AMT5		Amount of Previous Payment
PAY_AMT6		Amount of Previous Payment
default.payment	$Yes = 1$ or $No = 0$	Default Payment