Universidad
Rey Juan Carlos

Tesis doctoral

Explanation Sets: A framework for Machine Learning explicability

Autor:

Rubén Rodríguez Fernández

Directores:

Isaac Martín de Diego

Javier Martínez Moguerza

Programa de Doctorado en Tecnologías de la Información y las Comunicaciones

Escuela Internacional de Doctorado

Noviembre de 2023

If we knew what it was we were doing, it would not be called research, would it?.

— Albert Einstein

Agradecimientos

En primer lugar, quiero expresar mi más profundo agradecimiento a Javier e Isaac, mis directores de tesis, por ayudarme y guiarme durante este proceso. Sin las incontables sesiones de pizarra discutiendo infinidad de ideas, cada cual más loca, todo esto no habría sido posible.

También me gustaría agradecer a mi familia, en especial a mis padres, Isa y Miguel, a mi hermana Paula, y a Buddy. A mis primos favoritos, Irene y Hugo, y a mi tía Patri, que me ayudó a comprar mi primer ordenador. A Geli, por apoyarme durante todo el tiempo que he pasado en Madrid, a pesar de mis distracciones. Gracias a vosotros he tenido la motivación necesaria para seguir adelante y culminar este proceso. A mi abuelo, Onésimo, por toda la paciencia que tuvo ayudándome a construir mis "inventos" de pequeño, y a mi abuela, Benicia, por encubrirme cuando hacía trastadas. Gracias por todo el tiempo que pude pasar con vosotros. Y, por supuesto, a mi abuela Carmina, que siempre me insta a seguir adelante para poder presumir con sus amigas.

Tamara, gracias por ser la motivación y el ánimo detrás de mi decisión de hacer el doctorado, todo porque me diste envidia. Me alegra mucho haber compartido todo este proceso contigo, y espero con ansia nuestro viaje de fin de tesis a Japón.

A todos mis compañeros del DSLAB, y en especial a Víctor, mi mejor amigo. Te prometería que no te llevaría a más carreras que te dejen medio muerto, pero sería mentira. A Carmen y Marina, las mejores compañeras de despacho que he tenido: me habéis apoyado enormemente durante este proceso y espero que, tras este logro, pueda reunirme nuevamente con vosotras. Alberto, aunque difícil de creer, encontraste a alguien más cabezón que tú. Gracias por enseñarme a escribir artículos, sé que no fue fácil.

Por último, a toda la gente con la que he trabajado durante mi aventura en SpikaTech estos últimos meses: Alex, César, Laura[1], M&Ms, Paula, 'Sara' y Víctor. Han sido meses intensos y llenos de altibajos, pero el simple hecho de haberos conocido (o un poco más) ha hecho que mereciese la pena.

[1] Aunque no te contratasen.

Abstract

The term *Machine Learning* (*ML*) was coined by Arthur Samuel in 1959. Since then, more than sixty years have passed, and *ML* has evolved enormously, especially in the last decade. From the early days of *ML*, when it was primarily a research topic, to today, when we interact with *ML* systems on a daily basis, often without even realizing it, we have come a long way. Although the explainability of these *ML* systems has been considered since their inception, it has become more important than ever due to their integration into our daily lives. Explainable *ML* addresses this issue, aiming to make predictive models and their decisions understandable to humans.

There are several Explainable *ML* techniques, each with its own goals and scopes. For example, the scope of a technique can be either global, addressing the entire model, or local, focusing on a specific region of interest. While the choice of the technique depends on several factors, the main driving factor is the user, specifically their cognitive biases and what they expect from the system. These preferences and the different types of explanations have been extensively studied in the social sciences. Among these techniques, we emphasize counterfactuals and semifactuals, which have also been incorporated into Explainable *ML*. They are a contrastive explanation where the user reasons about the differences between the observation of interest and a hypothetical observation that led to the same prediction (semifactual) or a different prediction (counterfactuals). However, within the context of *ML*, they face some limitations. Both are mainly defined in a classification context and lack a standardized mechanism to enforce user preferences. Counterfactuals typically rely on a single observation, whereas semifactuals do not have a general definition and are associated with different terms.

This thesis introduces the Explanation Set framework to address these limitations. The Explanation Sets framework is an approach that unifies counterfactuals and semifactuals through similarity measures and provides users with mechanisms to specify their preferences via a feasible set. Besides providing a unified framework, the definitions based on similarity measures enable the seamless extension of counterfactuals and semifactuals to other tasks, like regressions, by using appropriate similarities. A review of how various techniques from the literature fit this framework is incorporated. The proposed approach was successfully validated in regression and classification tasks, showing how different feasible sets and similarity measures produce different explanations.

We also introduce two methods to extract Explanation Sets: *Anchor_ES* and *Random Forest Optimal Counterfactual Set Extractor* (*RF-OCSE*). *Anchor_ES* expands upon the *Anchor* method, allowing for user-defined similarity measures and including a feasible set. On the other hand, *RF-OCSE* is a method to extract counterfactual Explanation Sets from a *Random Forest* (*RF*). It involves a partial fusion of *Decision Tree*s (*DT*s) from a *RF* into a single *DT* using a modification of the *Classification and Regression Trees* (*CART*) algorithm. The proposed extraction methods were validated through several experiments against existing alternatives on several well-known datasets. The evaluation metrics measure aspects correlated with the quality of the explanations, including the percentage of valid counterfactuals, distance to the factual sample, method stability, and counterfactual set quality. *RF-OCSE* was the only method supporting set explanations that always yielded valid explanations and took, on average, significantly less time than the alternatives. Conversely, *Anchor_ES* obtained a good compromise between the fidelity and the coverage, and it emerges as a viable alternative, especially when full access to the model is not possible.

In conclusion, we introduce a novel explainability framework that empowers users to tailor explanations to their preferences. Explanation Sets pave the way for incorporating new preferences not currently recognized in the literature in a unified and standardized manner. This simplifies their eventual incorporation into extraction methods. Regarding the extraction methods, we noticed a significant disparity in quality between methods that utilize the internal structure of the model and those that use models as black-boxes, motivating the benefits of the former approach when possible.

Resumen

Antecedentes. El término *ML*, del inglés "*Machine Learning*" (Aprendizaje Automático), fue acuñado por Arthur Samuel en 1959. Desde entonces, han pasado más de sesenta años y el *ML* ha evolucionado enormemente, especialmente en la última década. Desde los primeros días del *ML*, cuando era principalmente un tema de investigación, hasta hoy, cuando interactuamos con sistemas de *ML* a diario, hemos recorrido un largo camino. Aunque la explicabilidad de estos sistemas de *ML* ha sido considerada desde su creación, hoy en día se ha vuelto más crucial que nunca debido a su integración en nuestra vida cotidiana. El *ML* explicable es un campo que aborda la explicabilidad de estos sistemas, con el objetivo de hacer que los modelos predictivos y sus decisiones sean comprensibles para los humanos.

Existen varias técnicas de *ML* explicable, cada una con sus propios objetivos y ámbitos. Por ejemplo, el ámbito de una técnica puede ser global, abordando el modelo completo, o local, centrándose en una región específica de interés. Si bien la elección de la técnica depende de varios factores, el principal es el usuario, específicamente sus sesgos cognitivos y el objetivo que se busca con la explicación. Estas preferencias y los diferentes tipos de explicación han sido ampliamente estudiados en las ciencias sociales. Entre estas técnicas, destacamos los contrafácticos y semifácticos, que también tienen aplicaciones en el *ML* explicable. Los contrafácticos son un tipo de explicación contrastiva que compara el escenario real con un escenario hipotético en el cual el resultado esperado es diferente del original. En cambio, los semifácticos también son explicaciones contrastivas, pero en este caso, el resultado del escenario hipotético coincide con el escenario real. A pesar de la similitud en sus representaciones, tienen un efecto diferente sobre nuestra percepción de las explicaciones, y la elección entre usar contrafácticos, semifácticos o ambos, depende del contexto particular.

Los contrafácticos y semifácticos presentan algunas limitaciones en el contexto de *ML*. Ambos tipos de explicación están principalmente definidos en un contexto de clasificación y carecen de un mecanismo estandarizado para expresar las preferencias del usuario. Además, los semifácticos no tienen una definición estándar, y se denotan con diferentes terminologías, lo que dificulta su adopción.

Objetivos. El principal objetivo de esta tesis es estudiar técnicas de explicabilidad de *ML*, extendiendo las técnicas de contrafácticos y semifácticos centrándonos en las preferencias del usuario, y validar su utilidad y rendimiento. Para abordar este propósito se han establecido los siguientes objetivos:

O1) Proporcionar una nueva metodología de explicabilidad que unifique contrafácticos y semifácticos basada en medidas de similitud, enfatizando su complementariedad, así como una metodología estándar para expresar las preferencias del usuario (conjunto factible).

O2) Elaborar una taxonomía de las representaciones basadas en conjuntos de la literatura para contrafácticos y semifácticos.

O3) Desarrollar un método agnóstico para extraer estas nuevas explicaciones basado en *Anchor*.

O4) Desarrollar un método para extraer estas nuevas explicaciones de un Bosque Aleatorio aprovechando su estructura interna y superficie de decisión paralela a los ejes.

O5) Validar la metodología de explicabilidad propuesta y comparar los métodos de extracción propuestos con alternativas en la literatura.

Metodología. Esta tesis introduce el marco de explicabilidad denominado Explanation Sets con el objetivo de abordar las limitaciones presentes en contrafácticos y semifácticos. El enfoque de Explanation Sets define contrafácticos y semifácticos mediante medidas de similitud y proporciona a los usuarios herramientas para definir sus preferencias a través de un conjunto factible (Objetivo O1). Las definiciones basadas en medidas de similitud subrayan la complementariedad entre contrafácticos y semifácticos, motivando su uso conjunto como método explicativo. La adaptabilidad de contrafácticos y semifácticos a tareas como regresión o detección de anomalías es natural, siempre que se pueda establecer una medida de similitud en la salida del modelo. Respecto al conjunto factible, se proporcionan ejemplos de cómo diferentes preferencias de contrafácticos en la literatura pueden expresarse mediante un método genérico de construcción de conjuntos factibles. Asimismo, se ofrece una taxonomía de las representaciones de contrafácticos y semifácticos según si imponen restricciones sobre la información a representar y si son aproximaciones (Objetivo O2).

Dentro del marco Explanation Sets, se propone un método llamado *Anchor_ES* para extraer conjuntos de contrafácticos y semifácticos basados en la técnica de explicabilidad *Anchor* (Objetivo O3). *Anchor* está orientado a generar conjuntos de semifácticos, y esta propuesta incorpora una adaptación para incluir restricciones del conjunto factible y distintas medidas de similitud. Para obtener conjuntos de contrafácticos, se añade un paso preliminar: primero, se extrae un contrafáctico utilizando técnicas de optimización bayesianas. Después, se emplea *Anchor* para producir una explicación para ese contrafáctico. Gracias a la relación complementaria entre contrafácticos y semifácticos, el resultado es un conjunto de contrafácticos.

Se introduce un método adicional, *RF-OCSE*, para extraer conjuntos de contrafácticos de un Bosque Aleatorio (Objetivo O4). Esta técnica se basa en fusionar parcialmente un Bosque Aleatorio en un Árbol de Decisión, aprovechando una adaptación del método *CART* para construir Árboles de Decisión. Esta técnica garantiza la obtención de un conjunto de contrafácticos que incluye el contrafáctico más cercano.

Finalmente, para evaluar el marco Explanation Sets y los métodos propuestos, se diseñan dos secciones de experimentos (Objetivo O5). La primera sección valida el marco a través de un caso de uso de regresión y otro de clasificación, empleando distintas preferencias para el conjunto factible. La segunda sección compara los métodos de extracción propuestos con técnicas alternativas para la obtención de contrafácticos y conjuntos de contrafácticos. Las métricas de evaluación incluyen el porcentaje de contrafácticos válidos, la proximidad a la muestra factual, la calidad de los conjuntos contrafácticos (cobertura y fidelidad), la estabilidad del método, y el tiempo requerido.

Resultados. En la primera sección de experimentos, se valida el marco de explicabilidad. Estas pruebas nos permiten investigar el impacto de diferentes medidas de similitud y conjuntos factibles. Se observa que la implementación de medidas de similitud puede transformar el escenario en un problema de clasificación desbalanceado, y esto afecta de manera notable a la calidad de las explicaciones. Esto no representa una limitación del enfoque propuesto, pero la mayoría de los métodos y métricas de calidad están definidas para escenarios equilibrados. Otra observación interesante es que, dependiendo de la medida de similitud seleccionada, a veces no es posible explicar un modelo mediante contrafácticos. En dicho caso, la ausencia de una explicación se convierte en la explicación en sí. Además, se identifica una distinción clara en la calidad entre conjuntos de semifácticos y conjuntos de contrafácticos generados por *Anchor_ES*. Esta diferencia radica en que los conjuntos de contrafácticos se generan en dos pasos, y su calidad no se optimiza a lo largo del proceso, a diferencia de lo que ocurre con los semifácticos.

En la segunda sección de experimentos, se compara la eficacia en la extracción de contrafácticos de los métodos propuestas con técnicas ya establecidas, utilizando conjuntos de datos reales. El método *RF-OCSE* destaca como el único enfoque basado en conjuntos que siempre genera explicaciones válidas y, en promedio, toma significativamente menos tiempo que las alternativas. En contraste, *Anchor_ES* logra un equilibrio entre fidelidad y cobertura, presentándose como una opción factible, particularmente cuando no se tiene acceso completo al modelo.

Conclusiones. La presente tesis se enfoca en la explicabilidad de modelos, y en ella se presenta un nuevo marco de explicabilidad denominado Explanation Sets. Este marco permite a los usuarios adaptar las explicaciones a sus propias preferencias y presenta además dos métodos específicos para extraer dichas explicaciones. Los Explanation Sets abren las puertas para la incorporación de nuevas preferencias, no abordadas en la literatura actual, de manera unificada y estandarizada. También posibilitan la extensión de contrafácticos y semifácticos a otros campos. Además, este enfoque unificado facilita su integración en futuros métodos de extracción. En cuanto a los métodos de extracción, se ha identificado una diferencia significativa entre los métodos que aprovechan la estructura interna del modelo y los que tratan a los modelos como cajas negras. Este hallazgo destaca las ventajas del primer enfoque, siempre que sea aplicable. Finalmente, el método que transforma Bosques Aleatorios en Árboles de Decisión destaca que los Árboles de Decisión son directamente explicables solo cuando tienen una complejidad reducida.

Contents

List of Figures

List of Tables

List of acronyms

AUC *Area under the ROC Curve*.

CART *Classification and Regression Trees*.

CLEAR *Counterfactual Local Explanations for Any Classifier*.

DT *Decision Tree*.

FBT *Forest-based Tree*.

FT *Feature-Tweaking*.

GAN *Generative Adversarial Networks*.

HS *Hot Start*.

ID3 *Iterative Dichotomiser 3*.

LORE *LOcal Rule-based Explanations*.

MACE *Model-Agnostic Counterfactual Explanation*.

ML *Machine Learning*.

MO *Minimum Observable*.

RF-OCSE *Random Forest Optimal Counterfactual Set Extractor*.

RF *Random Forest*.

SHAP *SHapley Additive exPlanations*.

SVM Support Vector Machine.

TPE *Tree Parzen Estimators*.

XAI *Explainable Artificial Intelligence*.

YADT *Yet Another Decision Tree builder*.

Chapter 1

Introduction

In today's data-driven society, *Machine Learning* (*ML*) plays an increasingly pivotal role in our daily activities, supporting decisions and tasks such as advertisement selection, route planning, and credit scoring. This automated decision-making has enhanced numerous aspects of our lives but also brings some notable challenges (Pentland, 2013; Wolff *et al.*, 2016). Biases against minorities, objective mismatches, and opacity of *ML* systems have raised concerns about their fairness and trustworthiness (Molnar, 2018; Doshi-Velez & Kim, 2017).

These challenges become especially pronounced in domains such as healthcare, where an incorrect prediction can have a significant impact. However, it also poses challenges in *ML* systems that might initially seem harmless. In systems where the output of the *ML* models affects the collected data, a phenomenon known as feedback loops might occur (Van Giffen *et al.*, 2022; Malik, 2020). These feedback loops can magnify initially small biases, and sometimes, they can make the model overconfident because it is trained on its predictions.

Numerous well-known cases highlight the consequences of deploying models without first applying appropriate Explainable *ML* techniques. For example, Google's image tagger mistakenly classified black people as gorillas, Facebook's advertisement model unintentionally displayed offensive content, Amazon's resume filter was found to be biased against women, and a healthcare system widely used in the U.S. required black patients to be significantly more ill to receive comparable care recommendations. These are just a few examples, with many more potentially remaining undisclosed.

The field of Explainable *ML* provides the tools to address these concerns, aiming to make *ML* models and their decisions comprehensible by humans (Arrieta *et al.*, 2020; Zhu *et al.*, 2018). The benefits of Explainable *ML* impact all the people involved with these systems, from those developing them to those using them (Belle & Papantonis, 2020). In essence, the goal is to ensure these systems work as expected and act in our best interests.

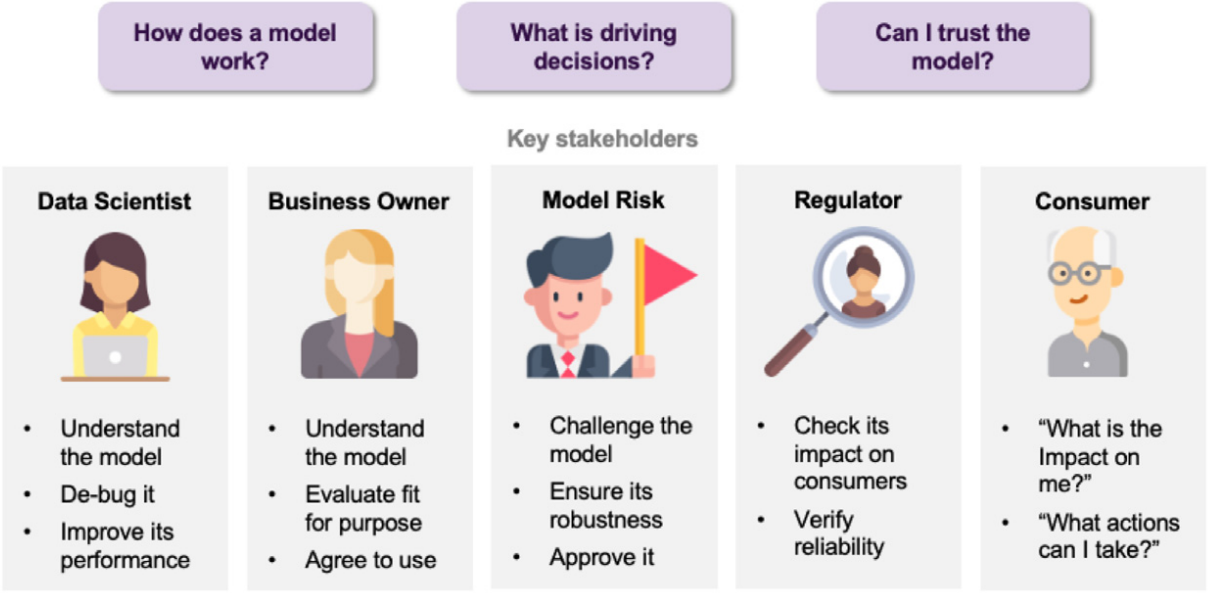

Figure 1.1: *Explainability requirements and questions faced by the different stakeholders. Reproduction from (Belle & Papantonis, 2020).*

This chapter is organized as follows. Section 1.1 provides the context and motivation for this thesis. Section 1.2 introduces the objectives that this thesis addresses. Lastly, Section 1.3 details the organization of this document.

1.1. Context and motivation

In the previous section, we showed examples where *ML* systems behave incorrectly and unethically, motivating the need to understand *ML* models properly. This section contextualizes how the different stakeholders can benefit from Explainable *ML*. Then, we provide an overview of Explainable *ML* techniques.

Figure 1.1 shows an illustration of the main stakeholders involved with *ML* models and the main question they wish to address by using Explainable *ML* (Belle & Papantonis, 2020). Data scientists, business owners, and model risk stakeholders are involved in the life cycle of a data science project, from the design of the requirements and validation to the actual development and testing of the system. The explainability goals can be aligned with each phase of a data science project, taking as a reference the life cycle proposed by Kelleher & Tierney (2018):

- **Business understanding**: It sets the problem and objectives and assesses the impact of the *ML* models and possible risks. As such, it orchestrates the explainability requirements for the other phases. For instance, these requirements might be based on business field regulations.

- **Data preprocessing**: It integrates and curates the data for the following tasks. Although Explainable *ML* does not explicitly target this phase because there might not be a model yet, certain data transformations might make it difficult to understand the models. Also, the mitigation measures for problems detected in other phases, like harmful biases, might be integrated here.

- **Modeling**: It builds *ML* models for the particular *ML* task. These models might be restricted to being "simple" because of interpretability concerns or have explainability constraints, for instance, in the training step (Plumb *et al.*, 2019; Krishnan & Wu, 2017). Also, explainability techniques can help to debug these models.

- **Evaluation**: It evaluates the models on unseen data using the metrics defined in the business understanding phase. These metrics can consider the complexity of the model, making a compromise between performance and explainability. For instance, in a *Decision Tree* (*DT*), it could be a trade-off between the depth of the *DT* and its performance.

- **Storytelling**: It communicates the results from the data analysis and the *ML* model. As such, it is the phase where Explainable *ML* shines. The target of this communication might not be technical people, and they are often the people who decide when a model is production-ready or detect domain-related problems (Krishnan & Wu, 2017), so this information must be conveyed appropriately and succinctly.

- **Deployment**: Is the integration of the *ML* model pipeline into production. Similar to storytelling, appropriate explainable techniques should be used so that end-users understand the output of the model and use it correctly. For instance, when the end-users do not understand the meaning of the output, we could incur deployment bias, where a model is optimized for a certain goal and then used to make unrelated decisions (Collins, 2018; Larson *et al.*, 2016).

Outside the data science life cycle, we have the remaining stakeholders: regulators and end-users. The regulators define the rules these *ML* systems must adhere to. The European Union is pioneering these regulations, proposing a legal framework on Artificial Intelligence (European Commission, 2023). In this regulatory framework, they propose a set of strict obligations that Artificial Intelligence systems must adhere to based on their risk level. In this context, Explainable *ML*, or more generally, *Explainable Artificial Intelligence* (*XAI*), can check if a system adheres to a given regulation by inspecting how the model works (Wachter *et al.*, 2017), or fulfill the end-user right to understand automated-decisions.

End-users are the actual people using the *ML* systems. Firstly, they have the right to know that they interact with a *ML* system (European Commission, 2023). Then, end-users should be able to understand the rationale behind an automated decision and not take it as given, particularly when the consequences are negative (Wachter *et al.*, 2017). Understanding these automated decisions can help them achieve better decision-making or even complain if they prejudice them.

In some scenarios, users can use this understanding to game (deceive) the system, "What do I have to change to obtain the outcome X?", this is not a limitation of Explainable *ML* but the *ML* model itself (Molnar, 2018). If a *ML* model is *gameable*, it indicates that it is relying on proxies (correlations) to make the predictions, not the actual causal relation. Therefore, if the user changes some of these proxy measures upon examining the explanation, it might lead to outcomes not aligned with the modeled process.

Recognizing the different necessities and goals of each stakeholder is crucial to provide an appropriate explanation method. Among these necessities and goals are the domain, education, or even the age of the target user. We have to emphasize that the goal of these techniques is to communicate, and as such, the user should be part of this conversation. An introduction to the most common explanation types to address this communication is presented in the next section.

1.1.1. Explanations in Machine Learning

Before delving into the different types of explanations, we will try to answer the question "What is an explanation in *ML*?". An explanation in *ML* is a mechanism to convey information about a *ML* model or how it operates in a particular scenario. It is usually an answer to a "why" question, which is a contrastive explanation (Miller, 2018). Research suggests that people do not explain the causes of why an event occurred (causal attribution). Instead, they explain the causes relative to another, possibly similar, event that did not occur (Miller, 2018).

The explanations often ought to answer the question: "Why P rather than Q?", where P is the fact, and Q is the foil (Lipton, 1990). For instance, "Why was the loan rejected rather than accepted?". This question, along with its answer, is called a counterfactual explanation. In some cases, the foil is not explicitly stated: "Why P?", "Why was the loan rejected?", in which the foil is all the available alternatives. Among these alternatives, those similar to P will provide a better explanation. In a binary classification problem, the choice is simple since we only have two classes, the foil is the class not predicted. In a multi-class problem, it would make more sense to use those classes that obtained the highest probability as foils (excluding the highest label, which would be the fact).

Another major finding from (Miller, 2018) is that users expect a small set of explanations (causes) and iterate through them based on their own biases. In other words, we do not generally expect all the causes leading to an event, but a few of them based on our biases, and we consider them to be *the explanation.*

The last major finding from (Miller, 2018) is that explanations are social. They are a conversation between the *explainer* and the *explainee* (person receiving the explanation). As such, the *explainer* must acknowledge the social context and target audience to determine the content and how it is presented to the *explainee.* Regarding the content, probability relationships to explain why an event occurred are unsatisfying unless they include a causal explanation (Miller, 2018). More information is not always better, and a few small explanations should be preferred (Molnar, 2018; Miller, 2018). However, in some situations, complex explanations are needed, for instance, if the regulation requires causal explanations.

Miller concludes these findings by stating that they converge to a single point, which is the core of this thesis:

> The *explainee* only cares about a small subset of explanations based on their preferences and context, and *explainer* and *explainee* might interact through these explanations.

These interactions might involve asking for more explanations or modifying the preferences for those explanations, among others.

After emphasizing how each stakeholder requires different explanations and what questions these explanations usually address, we move to the Explainable *ML* methods. Up to this point, it should bear no doubt that there is no one-size-fits-all solution for explaining *ML* models (Arya *et al.*, 2019). Each explanation method emphasizes different aspects of *ML* models and has different use cases. It is crucial to note that Explainable *ML* techniques work independently of the quality of such decisions. Thus, the accuracy of the *ML* model is not a necessary condition for a "good" explainability of the model: we are explaining the ML model independently of its accuracy (although it is desirable for *ML* models to describe reality accurately).

Following the taxonomy presented by Molnar (2018), explainability in *ML* can be approached on two levels: transparent models and post-hoc interpretability (see Figure 1.2 for a graphical depiction). For a more detailed taxonomy, we refer to (Arrieta *et al.*, 2020; Schwalbe & Finzel, 2023). Transparent models, the opposite of black-box models, refer to models considered interpretable due to their simple structure, such as Linear Regression, *DT*s, or Rule-based systems. However, transparent models become black-boxes as their complexity grows (Molnar, 2018; Adhikari *et al.*, 2019). On the other hand, post-hoc techniques try to explain *ML* models. Based on their output type, they can be categorized into surrogate models, feature statistics, or explanations based on examples.

Explanation techniques in *ML* can also be categorized based on three criteria (Molnar, 2018):

- **Scope**: *Global or local.* Global explanations target the whole model, and local explanations target a region of the feature space.

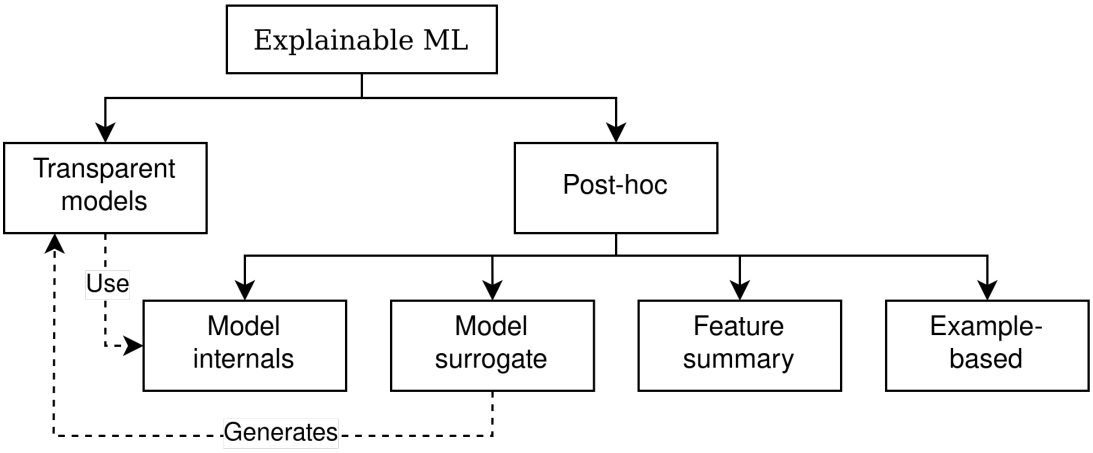

Figure 1.2: *A taxonomy of Explainable Machine Learning techniques. Adapted from Molnar (2018).*

- **Origin**: *Intrinsic or post-hoc.* Intrinsic explanations imply that the model can be directly understood (transparent models), and post-hoc explanations mean we use extra tools to help explain it.
- **Applicability**: *Agnostic or specific.* Agnostic explanation techniques can be used for any model, whereas specific explanation techniques target a specific model (or type of *ML* model).

The scope and origin of the explanations are often correlated. Global explanations are usually limited to transparent *ML* models, and local explanations are generally generated using post-hoc techniques. The rationale behind this correlation is that as the complexity of the *ML* model increases, it becomes harder to explain large regions of the input space, and we focus on small regions. There are exceptions to this correlation, such as global feature importance or global surrogates that approximate a complex *ML* model using a simpler one. However, sometimes, these aggregations discard relevant information to some (probably underrepresented) instances and generate misleading insights. Therefore, global explanations should be used with caution when they are generated using approximations. Examples of well-known post-hoc explanation techniques are:

- **SHapley Additive exPlanations (SHAP)** (Lundberg & Lee, 2017): *SHAP* explanations quantify the contribution of the individual features on the output. Specifically, Shapley values indicate how much a particular feature value deviates the output from the average prediction. Thus, the sum of these deviations makes up the difference between the prediction and the average prediction. *SHAP* is based on cooperative game theory and offers a theoretically rigorous approach to explanations. In addition, global (or group) explanations can be generated by combining individual Shapley values. In the previous taxonomy, *SHAP* is a post-hoc feature statistic technique with global and local scopes. *SHAP* can be generated using model-agnostic methods, such as *KernelShap* (Lundberg & Lee, 2017), and model-specific like *TreeShap* (Lundberg *et al.*, 2018).

- **Influential observations**: They are observations that had a big effect on the *ML* models. In other words, its absence in the training set would have changed the model significantly. For instance, they might be outliers or errors. These influential observations help us to understand the model parameters through the data, make robust models, and detect problems in the data. Examples of influential observation techniques are the Cook's distance (Cook, 1977) and influence functions (Koh & Liang, 2017). Cook's distance is a technique specific to linear models, and it measures the effect of removing an instance by retraining the model and comparing the output with the original model. Influence functions are a technique specific to differentiable models, which estimates the influence without retraining the model, making it apt for larger models such as deep learning models. Influential observations are an observation-based global post-hoc technique.

- **Forest-based Tree (FBT)** (Sagi & Rokach, 2020): *FBT* approximates a *Random Forest* (*RF*) with an interpretable *DT*, balancing between performance and explainability. It combines the conjunction sets (i.e., the rules) of each *DT* to form a unified representation of the *RF*. This unified representation is then organized hierarchically to yield a simplified and approximated *DT*. *FBT* is a global surrogate specific to *RF*.

- **Counterfactuals and semifactuals**: They are two example-based techniques whose goal is to explain the outcome of an observation of interest, often referred to as factual sample, using other observations. These explanation techniques are based on the comparison of the factual sample with another observation (or set of observations). In counterfactuals, the outcome for the factual sample and the other observation(s) is different, and in semifactuals, it is the same. Counterfactuals are one of the most widely used Explainable *ML* techniques because they resemble the human-thinking process (Molnar, 2018; Adhikari *et al.*, 2019). In contrast, semifactual-based techniques (although not often directly referred to like that in the *ML* literature) are gaining momentum (Dhurandhar *et al.*, 2018; Ribeiro *et al.*, 2018). Techniques that combine both approaches also exist (Dhurandhar *et al.*, 2018; Guidotti *et al.*, 2019).

This thesis focuses on counterfactuals and semifactuals because of their wide usage, complementarity, and robust foundation in the social sciences. In particular, it seeks to address existing limitations in their application. Most counterfactual explainability approaches face the following limitations: 1) they use only one observation, and 2) they are primarily defined in a classification context. In semifactuals, they lack a general definition and are primarily defined in a classification context. In both cases, they lack a standard approach for users to express their preferences, and although definitions of other tasks, such as regression, exist, they are ad-hoc (Stepin *et al.*, 2021). These limitations motivate the necessity of a user-oriented framework that formalizes counterfactuals and semifactuals to other *ML* tasks in terms of similarity measures, supports multiple observations, and enables users to express their preferences more effectively.

1.2. Objectives

The main goal of this thesis is to study *ML* explainability techniques, extending counterfactual and semifactual techniques with a focus on the target user, and validating their utility and performance. Based on this, the following objectives have been set:

O1) To provide a new explanation methodology unifying counterfactuals and semifactuals based on similarity measures, emphasizing their complementarity and a standard methodology to define the feasible sets.

O2) To provide a taxonomy of current set-based representations in the literature for counterfactuals and semifactuals.

O3) To develop an agnostic method to extract these new explanations based on *Anchor*, a well-known agnostic explanation method.

O4) To develop a method to extract these new explanations from a *RF* leveraging on its internal structure and axis-parallel decision surface.

O5) To validate the proposed explanation methodology and compare the extraction methods to alternatives in the literature.

From these goals arise the following research questions that we wish to address:

Q1) Can a *RF* be converted into a *DT*? Is it a valid mechanism to explain a *RF*?

Q2) From an explainability point of view, are sets of observations better than a single observation?

Q3) How do different notions of similarity affect the extracted counterfactual and semifactual explanations?

Q4) Are full-access explanation techniques better than black-box techniques?

1.3. Thesis organization

This document is organized into six chapters. The first two chapters introduce the objectives, motivation, and background of this document. Chapters 3, 4, and 5 detail how these objectives were addressed and their validation. The last chapter presents the conclusions and research opportunities.

Chapter 2 provides the background knowledge necessary to understand the proposal. It covers the notation, offers a brief introduction to RF and DTs, and presents a literature review of counterfactuals and semifactuals. This chapter sets the foundation for Chapter 3, where the Explanation Set framework is introduced, which integrates counterfactuals and semifactuals using user-defined similarity measures (Objective O1). It discusses the benefits of this similarity-based definition and proposes a feasible set definition for counterfactuals and semifactuals. In addition, it includes a taxonomy of representations from the literature that can represent Explanation Sets (Objective O2). Lastly, it presents an agnostic approach to extract Explanation Sets based on *Anchor* (Objective O3).

Chapter 4 presents *Random Forest Optimal Counterfactual Set Extractor* (*RF-OCSE*), a method for extracting counterfactual Explanation Sets from RF (Objective O4). The extraction approach is based on the fusion of a RF into a DT. Due to the exponential combinatory nature of this fusion, it introduces a partial fusion of the RF into a DT restricted to a particular region to reduce runtime.

Chapter 5 details the experiments to validate the proposed methodology and extraction techniques (Objective O5). It includes the evaluation of the Explanation Sets framework and examines the effect of various similarities and feasible sets on the explanations. It also compares the two proposed extraction techniques in counterfactual Explanation Sets extraction with existing methods in the literature.

Chapter 6 concludes this thesis by addressing the research questions posed in Section 1.2, listing the main contributions, and highlighting future research opportunities. Finally, a list of publications tied to this thesis is presented.

Chapter 2

Background

In this chapter, we lay the foundations for the proposed explanation methodology and extraction techniques discussed in subsequent chapters. It is structured as follows. Section 2.1 defines the notation conventions adopted throughout the document. Section 2.2 provides a brief introduction to *Decision Tree*s (*DT*s) and *Random Forest* (*RF*) *Machine Learning* (*ML*) models, emphasizing their construction. This foundational knowledge is crucial to understanding some counterfactual extraction methods, and it is the base for the proposal in Chapter 4. Finally, Section 2.3 presents a review of counterfactual and semifactual explanations in the literature, their desirable preferences, and methods to extract them.

2.1. Notation

Throughout this document, we adopt the following notation. Let f be a *ML* model:

$$f : \mathcal{X} \to \mathcal{Y} \tag{2.1}$$

that maps observations from the feature space, $\mathcal{X}$, to the output space, $\mathcal{Y}$. This *ML* model could represent various tasks such as classification, regression, clustering, anomaly detection, and more, provided they adhere to the above definition.

Typically, the feature space is defined as the real space, $\mathcal{X} = \mathbb{R}^p$, where p denotes the number of features. The output space of *ML* models is often either the real space with a single output, $\mathcal{Y} = \mathbb{R}$, or a discrete space representing the problem's labels (e.g., $\mathcal{Y} = \{-1, 1\}$ for binary classification or anomaly detection, or the identifier of each cluster in clustering).

We assume that the *ML* model is constructed using a set of training samples $X \subseteq \mathcal{X}$ that are independent and identically distributed. Note that more information might be needed to construct the model depending on the task, but we do not make assumptions about the availability of such information. For instance, if the problem is supervised classification, each element from the training set X has an associated label, such that the training set is $\mathcal{L} = \{(\mathbf{x}_i, \mathbf{y}_i) : i = 1, 2, \dots, n\}$. In such a case, these samples are drawn independently and identically from the joint distribution $(\mathcal{X}, \mathcal{Y})$.

Finally, lets consider $\hat{\mathbf{x}} \in \mathcal{X}$ as the sample whose explanation is of interest (factual sample). For clarity, we will assume that the model f returns a scalar. Thus, we will use $\hat{y} = f(\hat{\mathbf{x}})$ as the output of the model for the factual sample.

2.2. Machine Learning models

2.2.1. Decision Trees

*DT*s are one of the most well-known and used algorithms in the *ML* community. They are conceptually simple yet powerful; even non-experts can understand their inner workings. *DT*s provide a hierarchical structure where a split is made at each node based on a single feature, resulting in two or more partitions. While two-way splits are the most common, certain tree construction techniques allow multi-way trees. The objective of the splits is to maximize the homogeneity of the training data in the resulting partitions. The splitting process is repeated on the resulting partitions until a stopping criterion is met, obtaining a terminal node. Terminal nodes, or leaves, contain the predictions of the *DT*. For new instances, predictions are made by identifying the corresponding terminal node based on the node conditions.

Figure 2.1 illustrates a *DT*. This *DT* is presented from left to right, with the leftmost node being the root and the rightmost nodes representing the leaves (depicted as circles). The root node divides the data using the $f1 \leq 7.30$ condition. Observations not satisfying this condition (i.e., $f1 > 7.30$) proceed to the subsequent branch below. Here, another division occurs based on the feature $f1$, specifically $f1 \leq 7.75$. Continuing with observations not meeting this latter condition, we arrive at the bottom circle, a leaf (or terminal) node for the positive class. This implies that all observations following the described path are categorized as positive. The same reasoning can be applied to the other branches.

Among the various algorithms available for *DT* construction, *Iterative Dichotomiser 3* (*ID3*) (Quinlan, 1986), C4.5 (Quinlan, 2014), *Yet Another Decision Tree builder* (*YADT*) (Ruggieri, 2004), and *Classification and Regression Trees* (*CART*) (Breiman, 2017) are the most well-known. While each algorithm has its strengths and weaknesses, we focus on the *CART* algorithm due to its binary-splitting nature and its wide use in the field.

The *CART* algorithm starts with the entire training set and greedily searches for the optimal way to split the training set. All potential split values, the average of two

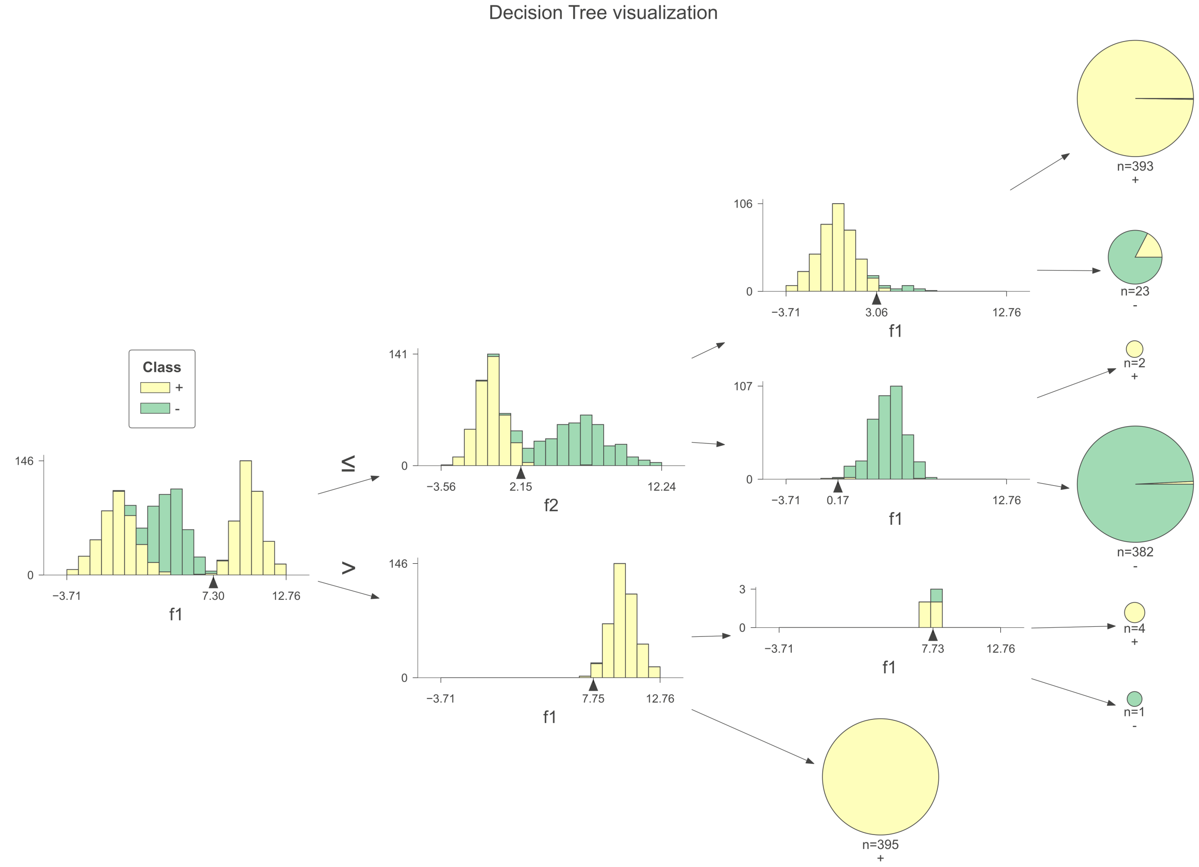

Figure 2.1: *Visualization of a Decision Tree using the dtreeviz library. The Decision Tree is displayed from left to right, based on the synthetic data discussed in Section 3.3. Nodes with a yellow color represent the positive class, while green ones represent the negative class. The terminal nodes, depicted as circles, are pie charts indicating the class distribution within that leaf. Each internal node specifies the feature used for the split (below the histogram) and the corresponding threshold value (indicated below the triangle).*

consecutive values from the sorted set of feature values, are considered to find the best split for a feature. Although the *CART* algorithm supports regression and classification tasks, this chapter focuses only on classification. In classification, the goodness of a split is calculated using the weighted average (by number of instances) of the Gini impurities of each partition. The Gini impurity, G, for a partition is computed as:

$$G = 1 - \sum_{i=1}^{l} p_i^2$$

where p_i denotes the probability of an instance being classified into class i, and l denotes the number of labels. This probability is determined by the relative frequency of the class within the partition.

After determining the best split across all features, the data is split, and the process is recursively applied to each partition. This process is repeated until a stopping criterion is met. The most common stopping criteria are:

- A node becomes pure (i.e., all instances from the same class).
- The number of instances in the partition falls below a certain threshold.
- A predefined maximum depth is reached.
- No significant improvement is observed in the splitting process.

Upon constructing the DT, a pruning step might be used to reduce its complexity. Given the susceptibility of DTs to overfitting, pruning serves as a regularization mechanism. Note that most of the current implementations are mixes of different construction methods. For instance, other criteria, such as Information Gain, are also used. Also, they do not usually perform an exhaustive search for the best split, resulting in less computing time and reducing the overfitting risk.

2.2.2. Random Forest

RF (Breiman, 2001) is a bagging method based on DTs. This section specifically addresses RF for classification. Essentially, a RF is defined as a set of DTs and a method to aggregate the outcomes of the individual DTs into a single result. The primary goal of using several DTs is to reduce the variance and the risk of overfitting inherent to DTs.

In the construction of a RF, multiple datasets are generated using bootstrapping on the original data. Subsequently, a DT is trained on each dataset. The DT construction sightly deviates from the $CART$ approach. Instead of greedily evaluating all features at each node, only a subset of m features is considered, being the most common option $\lceil m = \sqrt{p} \rceil$, where $\lceil \cdot \rceil$ is the ceiling function.

Another deviation is the stopping criteria. Individual DTs are grown to full depth until either node purity is achieved or the minimum number of instances in the partition is reached. Note that in most libraries, the maximum depth can also be specified.

These deviations ensure diversity among the DTs and can lead to overfitting of the individual DTs. However, the aggregation method counteracts this, reducing variance and enhancing the overall model performance (Aceña *et al.*, 2022). The most common aggregation methods are the most frequent class or the average probability, with the latter a generalization over the former.

2.3. Counterfactuals and semifactuals

In the previous chapter, we discussed the nature of an explanation, its desirable preferences, and provided a brief overview of counterfactuals and semifactuals. This section delves deeper into these topics. We begin by defining counterfactuals and semifactuals in the *ML* context. Then, we address their desirable preferences and extraction methods.

Counterfactuals have been thoroughly studied in social sciences, whereas semifactuals have gotten less attention (McCloy & Byrne, 2002). Both explanations share a similar representation but differ in their cognitive impact. Counterfactuals are represented by the conjecture "if only p, q" (e.g., study, pass the exam) and the presupposed fact "not-p, not-q" (e.g., not study, not pass the exam). Semifactuals present the conjecture as "even if p, not-q" and the presupposed fact "not-p, not-q" (e.g., not study, not pass the exam). In these conjectures, "p" is the predicate, and "q" is the consequent, corresponding to the observations and outcome in the *ML* literature.

According to McCloy & Byrne (2002), the inferences made about the relationship between the antecedent and the consequent may trigger different emotional responses. Counterfactuals amplify this response, and semifactuals reduce it. It has also been stated that a combination of counterfactuals (how to avoid an event) and semifactuals (hypothetical situations that would have led to the same event) help to evaluate the causal structure of an event correctly (McCloy & Byrne, 2002; Sherman & McConnell, 1995). Nevertheless, as previously mentioned, there is no silver bullet in explanation techniques, and the choice of counterfactuals, semifactuals, both, or other techniques depends on the problem and situation.

Similarly to the social sciences, counterfactuals have gotten more attention than semifactuals in the Explainable *ML* field. This is evidenced by the various terminologies that use semifactual-based techniques:

- **Anchors** (Ribeiro *et al.*, 2018): A sub-region (hyperbox) of the feature space around the factual sample where the prediction does not change.
- **Factual rules** (Guidotti *et al.*, 2019): The rule that classifies the factual sample from a surrogate *DT* built in the neighborhood of the factual sample.
- **Pertinent positives** (Dhurandhar *et al.*, 2018): Denote the minimum features needed to ensure the prediction. The other (unset) features can change, defining a set of observations where the minimum features are enforced, and the others can freely vary.
- **Prototypes** (Bien & Tibshirani, 2011): A set of similar instances that obtains the same classification. Also known as exemplars (Guidotti, 2022). They are not necessarily tied to a model and can be used directly over the training set.

Anchors and factual rules are conceptually the same: a hyperbox defined on the feature space containing the factual sample whose prediction is mostly constant. They only differ in their construction. In all these methods, the objective is the same: to draw inferences about an outcome by comparing the factual sample with other instances whose prediction is the same.

In contrast, most counterfactual-based techniques in the *ML* literature adhere to the definition provided by (Wachter *et al.*, 2018):

> "Score $\hat{y}$ was returned because variables $\hat{\mathbf{x}}$ had values $(\hat{x}_1, \hat{x}_2, \ldots)$ associated with them. If $\hat{\mathbf{x}}$ instead had values $(x'_1, x'_2, \ldots)$, and all other variables had remained constant, score y' would have been returned."

In this definition, $\hat{\mathbf{x}}$ is the observation whose explanation is of interest, and $\hat{y}$ is the prediction of a *ML* model f, such that $f(\hat{\mathbf{x}}) = \hat{y}$. The outcome $\hat{y}$ is known as fact, which is the event that occurs in the reality of the model f under the parameters $\hat{\mathbf{x}}$. The foil, y', is the event that did not occur and was the expected outcome. The parameters $\mathbf{x}'$ represent one of the possible scenarios where the outcome y' would have occurred.

We propose to adapt this definition to semifactuals by having $y' = \hat{y}$ and replacing "if" by "even if":

> "Score $\hat{y}$ was returned because variables $\hat{\mathbf{x}}$ had values $(\hat{x}_1, \hat{x}_2, \ldots)$ associated with them. Even if $\hat{\mathbf{x}}$ instead had values $(x'_1, x'_2, \ldots)$, and all other variables had remained constant, score $\hat{y}$ would have also been returned."

The changes between the counterfactual and semifactual definitions are marked in blue. Under this definition, a semifactual describes a scenario, $\mathbf{x}'$, that is different from the actual one, $\hat{\mathbf{x}}$, but leads to the same outcome, $\hat{y}$.

In *ML*, techniques similar to counterfactuals but with different goals also exist. The most well-known instance is adversarial examples (Mittelstadt *et al.*, 2019; Karimi *et al.*, 2020). While both aim to find observations close to the target but with a different predicted class, their purpose differs. The purpose of adversarial attacks is to make the model misclassify the sample. In most cases, the difference between the observation of interest and the adversarial attack is not easily perceptible by humans (e.g., one-pixel attacks in images (Su *et al.*, 2019; Alatalo *et al.*, 2022)). Conversely, the purpose of counterfactuals is to make the user reason about the difference between the observation of interest and the counterfactual observation. This difference will help the user to understand the outcome of the classifier and, therefore, should be noticeable to users.

Another technique similar to adversarial attacks is flip points (Yousefzadeh & O'Leary, 2019). They are defined as the points that lie on the decision surface (i.e., the positive and negative classes have the same score). Given that flip points lie on the decision surface, whether this technique generates counterfactuals or semifactuals is determined by the implementation of the *ML* models.

Though adversarial attacks and flip points can be viewed as counterfactuals, they usually would not be helpful to explain the *ML* model. This explainability is achieved in counterfactuals through a series of desirable preferences explained in the following section.

Similarly to the semifactual methods mentioned above, counterfactuals extend to multi-instance scenarios. For instance, counterfactual rules (Guidotti *et al.*, 2019) and diverse counterfactuals (Dhurandhar *et al.*, 2018). These methods are detailed in the following sections.

2.3.1. Desirable preferences

In a *ML* model defined in a continuous input space, the number of potential counterfactuals and semifactuals is usually infinite. For the sake of simplicity, we will refer to counterfactuals and semifactuals as explanations throughout this section. Further, most of them will not provide a satisfactory explanation, and we cannot consider all the alternatives (McCloy & Byrne, 2002; Fernández *et al.*, 2019). Consequently, the goal is to provide a few explanations based on specific desirable preferences, allowing the user to understand the problem from different perspectives.

Within the social sciences literature, the desirable preferences of counterfactuals have been deeply explored. In contrast, semifactuals have gotten less attention (McCloy & Byrne, 2002). Generally, certain factors in counterfactuals are seen as more mutable than others (McCloy & Byrne, 2002). Examples of such mutability preferences are voluntary over external changes, choosing actions over inactions, and prioritizing the most recent event in a series of related events.

The goal of the individual influences these preferences (Roese & Epstude, 2017). Drawing from (Roese & Epstude, 2017), even though the focus is on counterfactual thinking, the study offers insights into these preferences. For instance, the goal structure plays a significant role. Some people might be driven to feel better and, consequently, promote downward counterfactuals (situations less preferable than the actual outcome) over upward counterfactuals (situations better than the actual outcome) (K. White & Lehman, 2005; Roese & Epstude, 2017).

Another instance is the specific goal of the user. If the person seeks to improve the outcome, upward counterfactuals are more common. In contrast, if the person aims to maintain the status quo, downward counterfactuals are more fitting (Roese & Epstude, 2017). Also, if the user aims to achieve a goal, additive counterfactuals (adding new elements to the current scenario) are more common. Conversely, subtractive counterfactuals (remove elements from the current scenario) are more dominant if the aim is to prevent something.

The main conclusion from these counterfactual preferences underscores what we have emphasized throughout the document: the selection criteria for counterfactuals are deeply connected with the domain and target user. The abovementioned preferences are not quantifiable and thus cannot be implemented directly in the extraction methods. As a result, several studies have proposed metrics and restrictions as a proxy for these desirable preferences, ensuring that counterfactuals remain helpful. Metrics provide a quantifiable magnitude about the extent to which a preference is met. On the other hand, restrictions are binary, indicating whether a preference is met. We will use the term "preferences" to jointly refer to metrics and restrictions for simplicity.

A comprehensive review of these preferences for counterfactuals is provided in (Guidotti, 2022), building upon the previously rigorous works in (Mothilal *et al.*, 2020; Verma *et al.*, 2020). Since most preferences are defined for counterfactuals, we will indicate if the preference directly applies to semifactuals with (S✓) next to the preference name, (S✗) if it does not, or (S?) if any remark is made about the adaptation. Besides, in some preferences, we will discuss their relationship with the others. They discuss the following preferences:

- **Validity** (S✓): The validity is not a restriction per se but rather the indication of whether the observation is actually a counterfactual. In other words, it indicates whether the definition of counterfactual holds independently of its quality. For a given observation $\mathbf{x} \in \mathcal{X}$ it involves checking that $f(\mathbf{x}) \neq f(\hat{\mathbf{x}})$. For a semifactual to be valid, it corresponds to checking that an observation different from the factual sample $\mathbf{x} \in \mathcal{X} \cap \{\hat{\mathbf{x}}\}$ has the same prediction as the factual sample, $f(\mathbf{x}) = f(\hat{\mathbf{x}})$.

- **Similarity** (S?): For a counterfactual to be useful, it should be similar to the factual sample, as the explanation is based on the comparison between these two instances. The conclusions are likely to be irrelevant if these instances present no resemblance. As a proxy for the similarity, distance functions are considered. Thus, a closer counterfactual should be preferred. The most common distances considered in the literature are the Manhattan distance ($L1$ norm), the Euclidean distance ($L2$ norm), the median absolute deviance, and the Gower distance (Gower, 1971):

$$Gower(x, x') = \frac{\sum_{i \in \mathcal{F}} dist_i(x_i, x'_i)}{p} \tag{2.2}$$

  where $\mathcal{F} = \{1, \dots, p | x_i \neq x'_i\}$ represents the set of feature changes, p denotes the number of features, and $dist_i$ is a user-defined pairwise distance for the feature i, bounded between 0 and 1.

  While the same reasoning holds for semifactuals, emphasizing other preferences, such as diversity, becomes crucial to obtaining relevant instances in this scenario.

Since the prediction of semifactuals is the same as the factual sample, generating a semifactual observation near the factual observation is trivial, possibly with a high plausibility value. However, it is not useful since it is basically the factual sample.

- **Sparsity** (S✓): The sparsity, often referred to as the number of changes, is the number of features in which the counterfactual and the factual sample differ. A high sparsity should be preferred since few changes are easier to understand and less complex than a large number of changes.

  However, it is more complex. A high sparsity should be preferred while keeping the counterfactual closer to the factual sample. Otherwise, it is not useful. Consider two counterfactuals: the first suggests increasing the net income from $\$50k$ to $\$10M$, while the second suggests reducing the expenses by 1% and increasing the net income by $\$10k$. Although the first has a higher sparsity, the second would be preferable in most cases due to its more feasible changes.

  To balance between the cost of the changes and sparsity, we can consider distances such as the Manhattan distance and *sGower* (Fernández *et al.*, 2019):

  $$sGower(x, x') = \frac{\sum_{i \in \mathcal{F}} dist_i(x_i, x'_i)}{1 + p - |\mathcal{F}|}$$

  where $\mathcal{F} = \{1, \ldots, p | x_i \neq x'_i\}$ represents the set of feature changes, p denotes the number of features, $| \cdot |$ is the cardinality of the set, and $dist_i$ is a user-defined pairwise distance for the feature i, bounded between 0 and 1.

- **Plausibility** (S✓): Plausibility takes several names in the literature, such as feasibility, reliability, and data manifold closeness. It is a measure of how realistic the proposed counterfactual is. Counterfactual reasoning is similar to performing a simulation process on those hypothetical scenarios, so it makes sense to consider only scenarios that could happen. This can be easily conceptualized through domain-related restrictions (e.g., negative age or weight), which do not make sense in those simulations.

  Plausibility is mainly measured using the observed data (training set) and assumptions about the data distribution. Instances of methods to enforce plausibility in the literature are: generative models (Barredo-Arrieta & Del Ser, 2020), autoencoders (Dhurandhar *et al.*, 2018), density estimation techniques (Artelt & Hammer, 2020; Poyiadzi *et al.*, 2020), and ϵ-chain distances (Laugel *et al.*, 2019).

- **Discriminative Power** (S✗): The discriminative power is a proxy of the user's ability to recognize the reasons for the counterfactual outcome (Guidotti *et al.*, 2020; Guidotti, 2022). Therefore, it measures how the cognitive biases of the user align with the counterfactual. To be meaningful, discriminative power should be combined with a similarity criteria. This is crucial since a counterfactual with high discriminative power but far from the factual sample is not useful in the comparison process. Discriminative power is sometimes measured using a simpler model that is expected to correlate with the user's ability to recognize the counterfactuals (Guidotti *et al.*, 2020).

For semifactuals, this restriction is not directly applicable. The reason is that the user would classify observations close to the factual sample as being of the same class as the factual sample, and they do not help in the explanation process.

- **Actionability** (S✓): The actionability is a restriction over the counterfactual search space and is often referred to as recourse (Ustun *et al.*, 2019; Von Kügelgen *et al.*, 2022). Usually, a set of attributes is defined as non-actionable and cannot be changed (e.g., age and race). It is a method to enforce preferences over what can be mutated and what cannot. Under the same definition, actionability restrictions can be inequality constraints rather than equality (e.g., the value of a feature can only increase).

  The selection of actionable features completely depends on the domain and user preferences. For instance, age is usually mentioned as an instance of a non-actionable feature. However, in a *ML* system to determine an individual's eligibility for a driver's license, a relevant counterfactual might suggest that if the individual were above the required driving age, they could obtain the license.

- **Causality** (S✓): This casualty restriction refers to the causal relationship among the features, not between the input and the output (Guidotti, 2022). The counterfactuals should preserve any existent relationship between the features. This restriction aligns with plausibility, as plausible instances inherently keep causal relationships among the features.

  To illustrate this concept, consider a *ML* system with three features, with the third being the product of the first two. In this scenario, modifying the first feature in the counterfactual search should reflect on the third feature to preserve the causal relationship.

  As a side note, the causal relationship between the input and output is essential for a good counterfactual explanation. We assume the model provides a reliable approximation of the underlying process. If this assumption does not hold, the counterfactuals will not be useful to explain the underlying process.

Not all of these preferences are essential for a robust explanation; rather, they serve as general guidelines that can be tailored based on the goal. Further, most research on these preferences focuses on end-users, overlooking all users interacting with the system before deployment. For instance, actionability restrictions might hide harmful biases in the *ML* model. Another instance is plausibility. It could prevent us from identifying ill-defined regions in the *ML* model. If the goal is to reason about the *ML* model, minimizing the biases in the explanation generation is crucial, or at least being aware of them.

Although the aforementioned preferences are defined for individual explanations, they can be easily extended to a set of finite explanations. For instance, they can be calculated for each observation and then aggregate their results using the mean or maximum. In addition, we can define more preferences that relate to multiple instances (Guidotti, 2022):

- **Size** (S✓): It is the number of instances that contain the counterfactual set. The number of counterfactuals should be moderate based on the domain and the target user. Additionally, these instances should be diverse because otherwise, they do not provide new insights. While it may seem intuitive that having a larger number of counterfactuals would result in a better explanation, it could be counterproductive by increasing the user's cognitive load.

- **Diversity** (S✓): This metric quantifies the heterogeneity within a set of counterfactuals. A set with higher diversity should be preferred because users can approach counterfactual thinking from different perspectives. This also accounts for the preferences of the user; a diverse set allows users to pick the counterfactuals that align with their preferences and biases (Miller, 2018).

  Given a finite set of counterfactuals C, the diversity can be measured as the average pairwise distance between the counterfactuals (Guidotti, 2022; Mothilal *et al.*, 2020):

$$div_{dist} = \frac{1}{|C|^2} \sum_{\mathbf{x},\mathbf{x}' \in C} d(\mathbf{x}, \mathbf{x}') \tag{2.3}$$

  and also can be measured as the average pairwise number of feature changes (Guidotti, 2022):

$$div_{feat} = \frac{1}{p(|C|^2)} \sum_{\mathbf{x},\mathbf{x}' \in C} \phi(\mathbf{x}, \mathbf{x}') \tag{2.4}$$

  where p denotes the number of features, $\phi : \mathcal{X} \times \mathcal{X} \mapsto \mathcal{N}_0$ measures the number of values in which the two vectors differ, and d is a user-defined distance.

Finally, some preferences are defined for explanations that delineate regions of the space. The previous preferences are not directly applicable because these regions encompass infinite observations. Another significant difference is that region-based explanations often include non-valid observations. Thus, a counterfactual region might contain non-valid counterfactuals, and semifactual regions might contain non-valid semifactuals. These non-valid observations are considered exceptions and should be infrequent. Allowing exceptions enables broader and more generic explanations that are usually easier to understand. Another reason is that ensuring that all explanations in a region are valid is a challenging endeavor.

In contrast with the previous preferences, these preferences are used for counterfactuals and semifactuals within the literature. Let $C \subseteq \mathcal{X}$ be the observations inside the explanatory region. The region-based preferences are the following:

- **Fidelity** (S✓): The fidelity is the proportion of observations that are valid explanations within the set C. If the explanation is a counterfactual, it is the proportion of counterfactuals in the set C. A higher fidelity should be preferred since it indicates that most observations, except for a few exceptions, are valid explanations.

  Since the set C might contain infinite elements, the fidelity is often approximated using sampling procedures. Specifically, we sample observations from the set C obtaining the set C', and then we estimate the fidelity as the average number of valid explanations within these representative samples (Ribeiro *et al.*, 2018; Guidotti *et al.*, 2018; Bodria *et al.*, 2023):

$$fidelity_{\hat{\mathbf{x}},f}(C) = \frac{1}{|C|} \sum_{\mathbf{x}' \in C} \mathbb{1}[f(\hat{\mathbf{x}}) \neq f(\mathbf{x}')] \tag{2.5}$$

- **Coverage** (S✓): The coverage is the proportion of instances from the training set that are contained in the set S:

$$coverage(S) = \frac{|X \cap C|}{|X|} \tag{2.6}$$

  where $|\cdot|$ denotes the cardinality of the set. The coverage is a measure of the genericity of the explanation. As a result, explanations with higher coverage tend to be more comprehensible, provided that these instances are similar to the factual sample.

  This definition should not be mistaken with the coverage defined in (Mohammadi *et al.*, 2021), which refers to the number of cases where the method yielded a valid counterfactual.

The previous preferences for individual explanations can be extended to region-based explanations. In fact, the fidelity is the generalization of the valid restriction. Thus, the preferences can be estimated by sampling observations within the region and then aggregating the individual values. Note that the size preference does not make sense in region-based explanations because the number of observations is infinite.

The preferences introduced are primarily implemented using ad-hoc approaches tailored for each extraction method. The absence of a standardized methodology for representing these preferences makes their reuse and incorporation into extraction methods difficult. Furthermore, it complicates the combination of multiple preferences, particularly when they are metrics due to their continuous nature.

2.3.2. Extraction methods

After defining the preferences for counterfactual explanations in single, multi, and region-based settings and their extensions and applicability to semifactuals, we turn our attention to the extraction methods that enforce these preferences. In this section, we use the term "cost" to denote the objective function optimized during counterfactual and semifactual extraction. In most methods, this cost function incorporates the preferences under consideration. For an extensive review of counterfactual extraction methods, we refer to (Guidotti, 2022).

The extraction methods can be categorized based on their applicability (agnostic or specific) (Adadi & Berrada, 2018) and their extraction strategy (Guidotti, 2022). While this categorization is initially defined for counterfactuals, it is also applicable for semifactuals. The most common extraction strategies are:

- **Optimization** (OPT): Represents the cost function using an existing optimization framework and solves it.
- **Heuristic Search Strategy** (HSS): Uses local heuristics to minimize the cost function.
- **Instance-Based** (IB): Selects the instance from a dataset that minimizes the cost function.
- **Decision Tree** (DT): Constructs a surrogate DT and then exploits its structure to extract the counterfactual that minimizes the cost function.

The literature has also examined the preferences of counterfactual extraction methods. These preferences help to narrow down the available techniques based on the problem at hand (Guidotti, 2022). These preferences are directly applicable to semifactual extraction methods. The preferences are the following:

- **Efficiency**: Measures the time the counterfactual extraction method takes. The method should be fast enough to ensure practical applicability in real-world scenarios.
- **Stability**: Measures the variability in the counterfactual extraction. Close observations with the same prediction should have closer counterfactuals. This term has also been called robustness (Alvarez-Melis & Jaakkola, 2018).

  A method to quantify the robustness based on a local definition for the Lipschitz continuity is introduced in (Alvarez-Melis & Jaakkola, 2018). The robustness is defined as the maximum difference between the explanation for the factual sample and other observations in a ϵ-ball near the factual sample. Small differences in output imply more robust methods.

- **Fairness**: Promotes counterfactuals valid both in the "counterfactual world" and the "actual world" (Guidotti, 2022). Fairness is not always a desirable preference, mainly depending on the objective. For instance, if the aim is to understand the model from a developer's perspective, fairness constraints could complicate the debugging process. However, when providing explanations to end-users, ensuring fairness becomes imperative.

- **Validity**: Guarantees that a counterfactual can be found whenever it exists.

Method	**Aplicability**	**Strategy**	**Validity**	**Multiple**	**Target**
Anchor (Ribeiro *et al.*, 2018)	Agnostic	HSS		✓	S
(Barredo-Arrieta & Del Ser, 2020)	Neural Networks	HSS			C
(Blanchart, 2021)	Tree-based	DT	✓	✓	C
CLEAR (A. White & Garcez, 2019)	Agnostic	HSS	✓		C
FBT (Sagi & Rokach, 2020)	RF	DT			C
FT (Tolomei *et al.*, 2017)	RF	DT			C
Growing spheres (Laugel *et al.*, 2017)	Agnostic	HSS	✓		C
LORE (Guidotti *et al.*, 2019)	Agnostic	DT		✓	C,S
MACE (Karimi *et al.*, 2020)	Agnostic	OPT	✓		C
MO	Agnostic	IB	✓		C,S

Table 2.1: *Methods for extracting counterfactuals, semifactuals, and both. The applicability denotes the target* ML *model of the extraction method, and the strategy column details how the extraction problem is posed. In the validity column, a ✓indicates that the method guarantees that the counterfactual is valid. Similarly, a ✓denotes that the method returns more than one counterfactual or semifactual in the multiple column. The target column indicates if the method returns counterfactuals (C), semifactuals (S), or both (C,S).*

Table 2.1 introduces well-known methods from the literature for extracting counterfactuals and semifactuals. We classify them using the taxonomy proposed by (Guidotti, 2022), including whether they offer validity guarantees and indicating if they produce single or multiple explanations. Additionally, we specify whether these methods produce counterfactuals, semifactuals, or both. The inner workings of these methods will be elaborated upon later.

Methods specifically addressing semifactuals are not common, being *Anchor* its most prominent example. Some methods target counterfactuals and semifactuals, such as *LOcal Rule-based Explanations* (*LORE*) and *Minimum Observable* (*MO*), frequently employed as a baseline. Although not common, there are agnostic methods like *Model-Agnostic Counterfactual Explanation* (*MACE*) that come with validity guarantees.

DT-based approaches are common due to the relative simplicity of extracting counterfactuals and semifactuals from them. The process involves selecting the nearest sample from a rule whose prediction aligns with the desired outcome (Fernández *et al.*, 2019). Additionally, extracting multiple explanations is relatively easy, either by sampling observations from the rule or using the rule itself as an explanation. In addition, Tree-based approaches are often the target of extraction techniques that capitalize on their rule-based structures. We will provide a brief discussion about techniques to simplify *RF* to a *DT* to extract explanations when discussing *Forest-based Tree* (*FBT*).

In the following paragraphs, we will delve into each method, explaining their extraction procedure and output explanation.

MO: It is the closest instance from a set of observations satisfying the counterfactual or semifactual requirements. If these requirements only include validity, then *MO* is always defined as long as at least one instance is correctly classified for each class. If the observations are real (e.g., training set), then the method enforces plausibility and causality by default. The sparsity and similarity quality depend on the set S, larger sets are correlated with a higher quality.

Anchor: It is a method to extract semifactual sets, specifically, a hyperbox in the feature space that contains the factual sample. The set of minimal features that define the hyperbox is called anchor, and it ensures that most instances in the anchor have the same prediction. We use the term "*Anchor*" to refer to the extraction method and "anchor" to refer to the explanation. The construction of the anchor starts without restrictions (whole feature space), and it iteratively adds restrictions until a fidelity requirement is met (by default, 0.90). As a consequence of adding restrictions, the coverage is optimized indirectly. *Anchor* is an agnostic method that uses the model in black-box setting. Specifically, it uses the model to calculate the fidelity of the candidate anchors, using a multi-armed bandit approach that enables the reduction of the number of predictions made while having statistical guarantees.

(Barredo-Arrieta & Del Ser, 2020): It is a method to generate plausible image counterfactuals. The generation involves training a *Generative Adversarial Networks* (*GAN*) network, which is then used to generate new images based on a perturbation and attribute vector. They use metaheuristics search algorithms such as NSGAII (Deb *et al.*, 2002) to generate counterfactual images (through the perturbation vector) balancing between the validity (adversarial success), similarity (the magnitude of the changes), and the plausibility in terms of Pareto optimality.

(Blanchart, 2021): It is a method that takes the set of regions that define the tree ensemble (i.e., hyperboxes), and extracts the pure hyperboxes, regions where prediction is constant, from the fused tree ensemble. This approach is impractical in large tree ensembles because of the exponential combinatory nature of the process. Therefore, they propose a branch and bound optimization for extracting counterfactuals that iteratively shrinks the region of interest as closer counterfactuals are found. This optimization overcomes the exponential combinatory problem and keeps the search space within a reasonable space. However, the former only requires one conversion, while this approach is run for each new factual sample. In addition, it generates counterfactual sets from the hyperbox that classifies the closest counterfactual.

CLEAR: *Counterfactual Local Explanations for Any Classifier* (*CLEAR*) enriches explanations by providing a b-counterfactual and the regressions coefficients that explain a neighborhood that contains both the factual sample and the b-counterfactual. B-counterfactuals (boundary counterfactuals) are counterfactuals close to the decision boundary, similar to adversarial attacks and flip points. This technique improves LIME (Ribeiro *et al.*, 2016) and LEAFAGE (Adhikari *et al.*, 2019) by providing more information (b-counterfactual) and a better technique to generate the neighborhood. It improves existing counterfactual methods by providing the regressions coefficients, which has proven helpful in the literature (Ribeiro *et al.*, 2016).

FBT: It simplifies a *RF* into an approximate *DT*, keeping a high predictive power and resulting in a more interpretable model. From this *DT*, we can easily extract counterfactuals by selecting the closest rule to the factual sample. This distance between a rule and the factual sample is the smallest distance from the observations that satisfy the rule to the factual sample. Counterfactual sets are easily generated by taking the rule that classifies the counterfactual, similarly to *LORE*.

The approximation of the *RF* using a *DT* starts by pruning *DT*s from the *RF* based on the *Area under the ROC Curve* (*AUC*). Then, the *DT* are merged using a left fold reduction (i.e., first with the second, then with the third, and so on). At each fold reduction step, only the top L conjunctions by probability are kept, avoiding the combinatory explosion of the merging process. Then, the ruleset is combined into a single *DT*.

There are other similar methods that could be used in the same way to generate counterfactuals:

- (Bastani *et al.*, 2017): it greedily builds a *DT* similarly to the *CART* method using active sampling to calculate the splits. Specifically, it samples from a mixture of axis-aligned Gaussian fitted on the training data. It has convergence (to the target model) guarantees on the limit.

- (Deng, 2019): it is an evolved version of the work (Deng, 2014). It first extracts the rules from each *RF* and reassigns their outcome based on the whole dataset. Then, it applies a series of pre-processing steps: pruning based on the rule length, coverage, and precision, selection (Deng *et al.*, 2014; Deng & Runger, 2013), and frequent pattern extraction. Finally, it summarizes the rule set into a rule-based learner.

- (Zhou & Hooker, 2016): It is similar to the approach in (Bastani *et al.*, 2017), but it is specifically designed for *RF*. The major difference is that it only considers the split points of the original *RF* for candidates in the split selection process, significantly reducing the complexity of the method.

The quality of these techniques is mostly evaluated on the testing sets. While this choice is normal, it might generate explanation disagreements, specifically in the case of counterfactuals. Counterfactuals mostly live in the vicinity of the decision surface, and two models might have different decision surfaces while providing the exact same results in the testing set. This difference magnifies if the counterfactuals are extracted in low-density regions of the feature space. However, we hypothesize that these differences should be small if the counterfactuals are extracted with plausibility constraints because the regions where the counterfactuals are extracted will likely contain observations from the testing set. In summary, depending on the goal of the explanation, model-cloning techniques should be used with caution.

FT: It is a model-specific approach that extracts counterfactuals from tree-based ensembles using model internals. The counterfactual generation starts by first gathering all counterfactual rules from the *RF*. Then, for each of those rules, they update the instance to meet conditions. That is, for each condition not met in the rule, the feature value is set to the node value and adding ($>$) or subtracting ($\leq$) a small value ϵ. Then, they take the closest counterfactuals (if any). In some cases, the method might not be able to return a valid counterfactual (i.e., a counterfactual cannot be derived from a single rule).

Growing spheres: It is a method that searches counterfactuals within a sphere by shrinking or expanding its radius. The search within the sphere is performed using the *YPHL* algorithm (Harman & Lacko, 2010), which can efficiently sample within a sphere. Using the sampled points, the method checks if any of them is a counterfactual. The algorithm considers two cases, given a radius η. If a counterfactual is found in the η-ball centered in the factual sample, the radius is reduced to $\eta/2$. This process is repeated until a counterfactual is not found, returning the closest counterfactual. In the second case, if a counterfactual is not found, the method considers the range $(\eta_t, \eta_{t+1}]$, where $\eta_{t+1} = \eta_t + \eta$. The process is repeated until a counterfactual is found. In both cases, when a counterfactual is found, the method uses an approach to maximize its sparsity and returns it as the explanation.

LORE: It builds a surrogate DT using the YaDT algorithm (Ruggieri, 2004) in the neighborhood of the factual sample and then extracts a counterfactual by taking the closest counterfactual rule. The method returns the counterfactual rule that satisfies the counterfactual and also the factual rule, which is the rule that satisfies the factual sample, as an explanation. In the construction of the DT, a genetic algorithm is used to efficiently generate a representative dataset in the neighborhood of the factual sample. The genetic algorithm is run separately for each class, ensuring that both classes are well-represented.

MACE: It represents the counterfactual extraction as a satisfiability problem and extracts the closest counterfactual within an arbitrary tolerance. Both the model and the distance are expressed as a logic formulae, checking if there is a counterfactual closer than η. The process is repeated using a bisect method until the difference between the search extremes is lower than a user-given tolerance. This method works with any distance or convex combination of distances and has been tested using plausibility, actionability, and diversity preferences. *MACE* is defined as an agnostic method, but it requires a specific interface to encode the *ML* model into the formulae. Therefore, we argue that extensible is a better fit than agnostic because it does not work off-the-shell in all *ML* models such as *Anchor* or *CLEAR*. In their work, they provide such interfaces for RF, DT, logistic regression, and neural networks.

Among all the explored methods, *MACE* is the only one designed to easily incorporate other preferences. *MACE* can generate single or finite sets of counterfactuals using diversity preferences. However, no technique in the literature possesses such preference flexibility in a region-based setting, motivating the research of such techniques.

Chapter 3

Explanation Sets

This chapter presents Explanation Sets, a framework that unifies counterfactuals and semifactuals. Explanation Sets inherit the explanation properties of counterfactuals and semifactuals, which have been thoroughly studied in the social sciences and provide mechanisms to adapt them to different scenarios and preferences. This chapter addresses the following objectives:

O1) To provide a new explanation methodology unifying counterfactuals and semifactuals based on similarity measures, emphasizing their complementarity and a standard methodology to define the feasible sets.

O2) To provide a taxonomy of current set-based representations in the literature for counterfactuals and semifactuals.

O3) To develop an agnostic method to extract these new explanations based on *Anchor*, a well-known agnostic explanation method.

The chapter is structured as follows. The core concepts of this proposal are introduced in Section 3.1 (Objective O1). Section 3.2 includes examples of counterfactuals and semifactuals based on similarity measures. Section 3.3 elaborates on expressing explanation preferences (feasible set) within this unified framework and provides examples of common preferences in the literature. Section 3.4 introduces a taxonomy of Explanation Sets representations (Objective O2). Lastly, Section 3.5 outlines *Anchor_ES*, a method for extracting Explanation Sets from *Machine Learning* (*ML*) models (Objective O3).

3.1. Concepts

In this section, an example-based explanation framework called Explanation Sets is presented. Explanation Sets encompasses counterfactuals and semifactuals, and they are based on three fundamental ideas:

1. Define counterfactuals and semifactuals using user-defined similarities.

2. Specify a feasible set containing only the observations relevant to the user using a standard methodology.

3. Employ a set of observations, rather than a single observation, to provide more information.

We will explore these concepts individually before formally defining Explanation Sets.

Counterfactuals and semifactuals based on similarities.

Semifactuals are observations distinct from the factual sample that yield identical predictions. In this definition, the notion of "identical" refers to the identity similarity (or Kronecker delta):

$$\delta(y, y') = \begin{cases} 1 & \text{if } y = y' \\ 0 & \text{otherwise} \end{cases} \tag{3.1}$$

Thus, an observation $\mathbf{x}$ qualifies as a semifactual for the ML model, f, and factual sample $\hat{\mathbf{x}}$ if:

$$\mathbf{x} \in \mathcal{X} \cap \{\hat{\mathbf{x}}\} \ ; \ \delta(f(\hat{\mathbf{x}}), f(\mathbf{x})) = 1 \tag{3.2}$$

Conversely, counterfactuals require the outputs to be different. Using the identity similarity, an observation $\mathbf{x}$ qualifies as counterfactual if:

$$\mathbf{x} \in \mathcal{X} \ ; \ \delta(f(\hat{\mathbf{x}}), f(\mathbf{x})) = 0 \tag{3.3}$$

Excluding the factual sample, any observation from the feature space can be categorized as either a counterfactual or a semifactual. Thus, the set of all counterfactuals is complementary to the set of all semifactuals when the factual sample is excluded.

This relationship leads to two interesting scenarios. In the first scenario, consider a constant model. In such a case, all observations except the factual sample are semifactuals, leaving no counterfactuals. Second, consider a model where the factual sample is classified as negative and all other observations as positive. Then, all observations except the factual sample are counterfactuals, with no semifactuals present. In both cases, the absence of counterfactuals or semifactuals, respectively, becomes the most significant explanation. This interdependence between counterfactuals and semifactuals motivates their generalization and combined use in explanations.

Alternatively, counterfactuals and semifactuals can be expressed in terms of dissimilarities. For instance, the identity similarity can be transformed into a dissimilarity as:

$$\delta'(y, y') = \begin{cases} 1 & \text{if } \delta(y, y') = 0 \\ 0 & \text{if } \delta(y, y') = 1 \end{cases} \tag{3.4}$$

Then, the counterfactual definition can be reformulated as:

$$\mathbf{x} \in \mathcal{X} \; ; \; \delta'(f(\hat{\mathbf{x}}), f(\mathbf{x})) = 1 \tag{3.5}$$

The implicit usage of the identity similarity in counterfactuals and semifactuals is a natural choice in problems like binary classification. However, alternative notions of similarity can be employed for tasks with continuous outputs or multi-class/multi-label problems. Consequently, custom similarity definitions tailored to specific problems and user requirements can be used, extending semifactuals and counterfactuals to tasks where a similarity can be defined in the output space.

To unify the definitions of counterfactuals and semifactuals, we introduce a surjective mapping called the grouping measure, $m : Y{\times}Y \mapsto \{0, 1\}$, which can be either a similarity or dissimilarity. This mapping indicates whether two elements should be grouped (1) or not (0). Similarity-based grouping measures group similar observations, while dissimilarity-based measures group dissimilar ones.

Similarity and dissimilarity measures can be adapted to meet the grouping measure requirements by introducing a cut-off threshold. Let u be a similarity or dissimilarity, and $\epsilon \in (0, \infty)$ the cut-off threshold, then:

$$m_\epsilon(y, y') = \begin{cases} 1 & \text{if } u(y, y') > \epsilon \\ 0 & \text{otherwise} \end{cases}$$

Values above ϵ are considered equal in a similarity and distinct in a dissimilarity. Furthermore, they offer a straightforward conversion between them, similar to Eq. 3.4.

Then, we can rewrite the previous counterfactual and semifactual definitions into a single definition using the grouping measure m as follows:

$$\mathbf{x} \in \mathcal{X} \cap \{\hat{\mathbf{x}}\} \; ; \; m(f(\hat{\mathbf{x}}), f(\mathbf{x})) = 1$$

In this definition, the choice between a similarity or a dissimilarity based grouping measure determines the derivation of semifactuals or counterfactuals. Notice that the factual sample is also excluded compared to the counterfactual definition. However, since a dissimilarity between an entity and itself is 0, this exclusion does not impact the definition.

Section 3.2 offers examples of similarity and dissimilarity measures, highlighting their advantages, applications, and complementarity.

Feasible set.

The feasible set is introduced to express preferences over counterfactuals and semifactuals. This helps to reduce the observations to those of interest (e.g., close instances) and keep only those relevant to the domain (e.g., positive age and weight). The feasible set, S, is a subset of the feature space:

$$S \subseteq \mathcal{X}$$

Using the feasible set, we can rewrite our previous unified definition of counterfactuals and semifactuals as follows:

$$\mathbf{x} \in S \cap \{\hat{\mathbf{x}}\} \ : \ m(f(\hat{\mathbf{x}}), f(\mathbf{x})) = 1$$

Only observations from the feasible set S, excluding the factual sample, are considered candidates for semifactuals or counterfactuals.

Section 3.3 provides a compact representation for the feasible sets and illustrates how various examples from the literature are expressed with this framework.

Employ a set of observations.

Finally, using sets of counterfactuals and semifactuals rather than a single observation. The idea is simple: using more observations will provide more information than a single observation. However, this only applies if this set of observations can be presented in an easy-to-understand manner. Also, depending on the topology and representation of the set, it could have more properties. For instance, rule-based representations contain infinite observations using a simple representation (Guidotti *et al.*, 2018; Ribeiro *et al.*, 2018). They are particularly useful when features have a high variability, such as luminosity and temperature, or when referring to measurements like height or weight. Because of the measurement noise, these features are more effectively represented using a range rather than a fixed value.

These sets of observations might include observations that do not meet the grouping measure criteria, but their inclusion result in a more generic explanation. We can think of those non-compliant observations as a small group of exceptions that do not influence the overall explanation. This approach is also considered in methods like *Anchor* (Ribeiro *et al.*, 2018) and *LOcal Rule-based Explanations* (*LORE*) (Guidotti *et al.*, 2018). To quantify the proportion (which can also be expressed as a percentage) of these compliant observations within a set, we can define a measure called fidelity as follows:

$$fidelity_{m,\hat{\mathbf{x}},\mu,f}(U) = \frac{\mu(\{\mathbf{x} \in U \ : \ m(f(\hat{\mathbf{x}}), f(\mathbf{x})) = 1\})}{\mu(U)}$$

where μ is a measure of U. The fidelity of an empty set is 0. Note that this definition is similar to *Anchor* fidelity.

A **constant set** is a set whose fidelity is either 0 or 1. Conversely, a *ML* is said to be constant under the grouping measure m and factual sample $\hat{\mathbf{x}}$ if it is constant in the set $\mathcal{X}$, i.e., if $fidelity_{m,\hat{\mathbf{x}},\mu,f}(\mathcal{X}) \in \{0, 1\}$. When the grouping measure and factual sample are clear from the context, we will use only the term **constant ML model**.

The calculation of the fidelity is often intractable because it involves sets with infinite elements (e.g., a set represented by feature ranges with continuous variables). In such a case, an alternative is to estimate the fidelity by sampling observations from the given set. The fidelity is then calculated over this sampled set, with μ set as its cardinality. As an alternative to this sampling process, we can estimate it using the training feasible set, which is the intersection of the training set and the feasible set, or a hybrid approach combining both methods.

Section 3.4 delves further into the representations of Explanation Sets using various examples.

Explanation Set definition.

Now, we can move to the definition of Explanation Sets. An Explanation Set, E_α, is a subset of the feasible set S, whose fidelity is equal or higher than a user-given value $\alpha \in (0, 1]$:

$$E_\alpha \subseteq S \ : \ fidelity_{m,\hat{\mathbf{x}},\mu,f}(E) \geq \alpha \tag{3.6}$$

Explanation Sets are a technique for explaining the outcome of the model f for the observation $\hat{\mathbf{x}}$. They are defined as a subset of the feasible set, S, that contains observations whose comparison with the sample $\hat{\mathbf{x}}$ might be illustrative to explain the outcome $f(\hat{\mathbf{x}})$. The parameter α determines the maximum proportion of observations that do not meet the grouping measure, m, in the Explanation Set. It is a trade-off between the size and composition of the Explanation Set. A high value of α (e.g., 0.90) should be preferred since it indicates that the explanation is faithful to its purpose. However, if the resulting explanations are too specific, the α values could be slightly reduced (e.g., from 0.90 to 0.85). "Explanation Sets" where the fidelity requirement is not met are considered pseudo Explanation Sets. An approach to generate Explanation Sets is described in Section 3.5.

If the grouping measure is a dissimilarity, then the Explanation Set is a Counterfactual Explanation Set (in short, a counterfactual set). Conversely, if it is a similarity, it is a Semifactual Explanation Set (in short, a semifactual set). Explanation Sets inherit the properties of counterfactuals and semifactuals but also equip users with tools to refine them based on their preferences (feasible set) and apply them to other tasks (similarities). Utilizing sets facilitates considering multiple scenarios simultaneously, resulting in a more informed decision-making and mitigating the influence of noisy regions of the feature space. To utilize counterfactual and semifactual Explanation Sets, their existence is the primary prerequisite. However, for more meaningful explanations, it is essential to check whether one of the two scenarios mentioned earlier applies: a constant *ML* model (yielding only semifactuals) or if there are only counterfactuals. In these instances, the absence of counterfactuals and semifactuals, respectively, becomes the most meaningful explanation.

Explanation Sets allow non-compliant (i.e., not meeting the grouping measure) observations for two main reasons. First, guaranteeing that all observations within a set satisfy the grouping measure is complex and expensive when the number of observations in the Explanation Set is infinite. For instance, in Explanation Sets represented by feature intervals in a tree-based ensemble, it could be calculated by modifying the approach (Blanchart, 2021). However, this calculation might be impractical with models without axis-parallel decision surfaces. Second, allowing a low (and controlled) number of non-compliant observations results in broader and less specific explanations.

Consequently, the goal is usually to find an Explanation Set (or several Explanation Sets) satisfying the requirements while providing a broad and simple explanation (similarly to *Anchor* (Ribeiro *et al.*, 2018)). For explanations that define regions of the input space, the broadness can be measured by the coverage (Ribeiro *et al.*, 2018; Guidotti *et al.*, 2018). This metric indicates the percentage of instances from the training set contained in the region. Explanations with higher coverage are generally easier to understand as they can be contrasted with more real observations from the dataset.

In certain situations, the methods to generate Explanation Sets (also referred to as extraction methods) might yield an explanation that fails to meet the requirements. This can happen for two reasons: either the feasible set is constant and does not contain the required explanation (e.g., it contains only semifactuals, and we target counterfactuals), or the method used to generate Explanation Sets cannot find a valid explanation.

To determine if we are in the first case, we can encode the problem requirements into *Model-Agnostic Counterfactual Explanation* (*MACE*) (Karimi *et al.*, 2020). If we extract counterfactuals, these requirements include the feasible set and the dissimilarity. Conversely, if the target is semifactuals, the requirements are the feasible set and the similarity. In both cases, if *MACE* identifies an explanation, the problem is related to the extraction method because the feasible set is not constant. However, this approach has some drawbacks: it requires full access to the ML model, is computationally expensive, and is only available for some ML models.

In some cases, this verification can also be done manually using the properties of the model. For example, a *Random Forest* (*RF*) or *Decision Tree* (*DT*) cannot make predictions higher or lower than the maximum and minimum value of the label in the training set, respectively (see proof in Annex A). In most other scenarios, determining that we are in the first case is not currently feasible.

Determining the second case is trivial, for instance, checking if the explanation meets the minimum fidelity constraint. In both cases, this outcome also gives us valuable information. First, the observations in the feasible set that satisfy the grouping measure are rare (or sparse), or they do not exist. In the particular case when the grouping measure is a similarity, this fact also suggests that the outcome of the factual sample is not a common scenario. Second, the restrictions given by the user (grouping measure, feasible set, or both) might be very restrictive and should be relaxed.

3.2. Counterfactuals and semifactuals based on similarity measures

This section delves into the advantages of utilizing similarity and dissimilarity measures to generalize counterfactuals and semifactuals, emphasizing their distinct explanatory properties. Through two illustrative examples, we aim to show the practical implications of these measures.

Example 1: House Price Estimation.

Consider a *ML* system designed to predict house prices. This system evaluates attributes such as location, size, and the number of bedrooms. Given an estimated price of v, a user might wonder why the house is worth v rather than more than $v + o$. For this scenario, the dissimilarity measure $gt_{b,o}$ can be formulated as:

$$gt_{b,o}(v, v') = \begin{cases} 0 & : \text{otherwise} \\ 1 & : \min(v, v') \leq b \wedge \max(v, v') > b + o \end{cases} \tag{3.7}$$

Here, $o > 0$ and b (set to $b = v$) ensures symmetry in the dissimilarity. This dissimilarity is 1 when one of v or v' is equal or less than b, and the other is greater than $b+o$. Otherwise, the dissimilarity is 0. If symmetry is not required, the measure simplifies to:

$$gt_{o}(v, v') = \begin{cases} 0 & : \text{otherwise} \\ 1 & : v' > v + o \end{cases} \tag{3.8}$$

The simplified dissimilarity gt_o returns 1 when v' is greater than v plus o, and 0 otherwise. Since we rely on the order of the parameters, this dissimilarity is not symmetric.

Using this dissimilarity, counterfactual explanations will spotlight houses priced above $v + o$. These insights can guide potential modifications to increase the price of the house.

Alternatively, if a user wishes to identify features that can be altered without deviating the price from the range $[v-k,\ v+k]$, a similarity measure sr_k can be employed:

$$sr_k(v,v') = \begin{cases} 0 & : \text{otherwise} \\ 1 & : |v-v'| \leq k \end{cases} \tag{3.9}$$

The similarity sr_k returns 1 when the absolute difference $|v - v'|$ is equal or less than k and 0 otherwise. Semifactual explanations in this scenario might suggest actions like selling the furniture, which might not significantly impact its valuation, or indicate that the estimation is solely based on the location.

Example 2: Disease Risk Estimation.

Consider a *ML* system predicting the likelihood of contracting a specific disease based on lifestyle and preventive measures. Upon receiving a risk estimation of r, a user might ponder the rationale behind this assessment. Given that risk is a continuous metric, equating r to r' (another observation's risk) might yield limited similar observations. Hence, a similarity measure s_{risk} is defined analogously to Eq. 3.9.

Semifactual explanations here can spotlight activities or precautions that do not influence the risk estimation, offering insights into potential lifestyle changes without altering the risk. Conversely, counterfactual explanations highlight factors that could significantly modify the risk, necessitating caution.

While counterfactual and semifactual could be interchangeably used in both examples, they are more apt for their respective scenarios. For instance, using semifactuals in the first example might lead to discarding potential explanations, whereas counterfactuals would directly provide them. The reverse logic applies to the second scenario. By leveraging both types of explanations, we can comprehensively understand influential and non-influential factors in predictions.

3.3. Feasible set

Explanation Sets are defined within a subset of the feature space, known as the feasible set, which allows us to restrict the explanations to specific regions of the feature space. We employ the term "restrictions" to describe the preferences from Section 2.3.1, regardless of whether they are metrics or actual restrictions because they operate in a binary manner: they are either met (belong to the feasible set) or not. As previously mentioned, these preferences can serve various purposes, such as enforcing actionability, promoting sparsity, ensuring data manifold closeness, enhancing diversity, and providing a local context (i.e., instances close under a given distance).

We propose a unified approach to implementing these preferences using a "smaller than" inequality restriction. The restriction function, g, is defined as:

$$g : \mathcal{R}^p \times \mathcal{R}^p \mapsto \mathcal{R}_{\geq 0}$$

The first parameter of g corresponds to the factual sample, while the second parameter is an observation we wish to evaluate against the feasible set.

The feasible set, denoted as $S_{g,r}$, is defined using the "smaller than" inequality restriction as follows:

$$S_{g,r} = \{g(\hat{\mathbf{x}}, \mathbf{x}) < r : \mathbf{x} \in \mathcal{X}\}$$

where $\hat{\mathbf{x}}$ is the factual sample, and $r \in (0, \infty)$ is the restriction value. All observations from the feature space, $\mathcal{X}$, whose restriction function against the factual sample is less than r, belong to the feasible set. The decision to employ this type of inequality restriction arises from the observation that most restrictions will be distances.

We propose a simple taxonomy of feasible sets in Explanation Sets based on the type of restriction: user-defined or model-induced. A user-defined restriction can be any restriction function. Model-induced restrictions are based on the properties of the *ML* model. Examples of model-induced restrictions are the kernel in Support Vector Machines (*SVM*s) and the proximity measure (Breiman, 2002) in *RF*. Model-induced restrictions have the additional property of grouping similar individuals in the space where the *ML* model projects the observations.

We hypothesize that model-induced restrictions are beneficial when debugging and developing *ML* models. In this scenario, Explanation Sets contain individuals similar under the model reality (model-induced metric) that should be grouped based on the criteria of the user (grouping measure). Ideally, the grouping measure in this scenario should group similar outcomes like the similarity in Eq. 3.9 for continuous outcomes or the identity similarity for discrete outcomes. Thus, developers can determine if the way the model projects the instances makes sense in the scenario defined by the grouping measure. On the other hand, user-defined restrictions introduce a bias in the instance selection process. Consequently, user-defined restrictions might group instances that are not similar under the model perspective. This bias might provide better explanations from a user perspective, but it might hide relevant details in other scenarios like model debugging.

In the following paragraphs, we provide illustrative examples of these restrictions, including actionability, sparsity, data manifold closeness, and diversity, to demonstrate their practical applications and how they can be combined to form more complex constraints. We will guide them through a synthetic binary classification example to illustrate the restrictions better.

The dataset is sampled from three bivariate normal distributions, $\mathcal{N}_1(\mu_1, \Sigma_1)$, $\mathcal{N}_2(\mu_2, \Sigma_2)$,

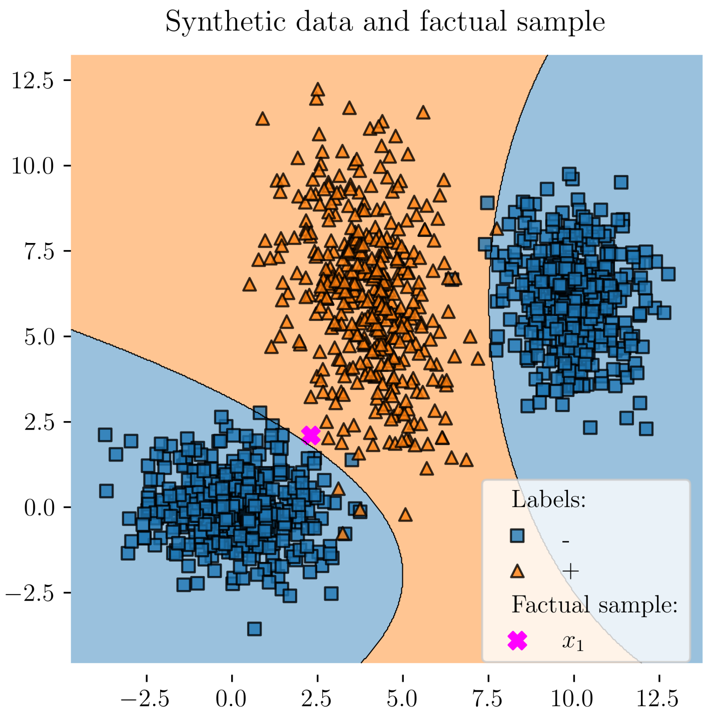

Figure 3.1: *Binary classification problem with synthetic data. The orange triangles are the data from the positive class, while the blue squares are the data from the negative class. Using a Bayes classifier, the orange and blue regions are classified as positive and negative, respectively. The pink cross is the factual sample.*

$\mathcal{N}_3(\mu_3, \Sigma_3)$. The data sampled from $\mathcal{N}_1$ belongs to the positive class, while the data from $\mathcal{N}_2$ and $\mathcal{N}_3$ to the negative class. The data is generated with Python using the *NumPy* package (Harris *et al.*, 2020) with seed 1234. A total of 400 points is sampled from each distribution, with parameters:

$$\Sigma_1 = \begin{bmatrix} 2 & -4 \\ 0 & 4 \end{bmatrix}, \ \Sigma_2 = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}, \ \Sigma_3 = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}, \ \mu_1 = [4, 6], \ \mu_2 = [0, 0], \ \mu_3 = [10, 6]$$

We consider an observation to be explained (factual sample), $\hat{\mathbf{x}} = [2.3, 2.1]$, located proximate to the decision boundary. The observation is classified as positive (+). Figure 3.1 depicts the sampled data, the theoretical decision frontier, and the point to be explained. The separation between the observations of each sampling distribution can be clearly seen since they have minimal overlap. This separation accounts for the accurate classification of most observations.

First, we consider a base restriction. In the literature, it is usually the Manhattan distance ($L1$ norm). This distance is used to enforce sparsity (few feature changes), which makes explanations easier to understand. Figure 3.2 shows the Manhattan distance to the factual samples.

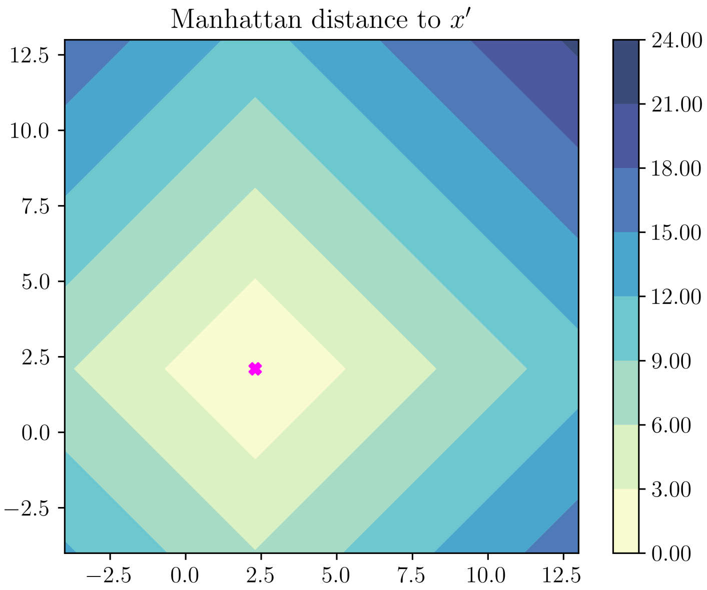

Figure 3.2: *The contour plot illustrates the Manhattan distance to the factual sample (pink cross). The color gradient from yellow (closest) to dark blue (furthest) indicates the distance.*

Actionability restricts observations meeting a condition. Consequently, it can be modeled as having the distance to those individuals that do not meet the condition higher than the restriction value r. For instance, we could define g_lp_i to restrict those instances whose feature x_i is equal or lower than ξ as follows:

$$g\_lp_i(\hat{\mathbf{x}}, \mathbf{x}) = \begin{cases} -x_i + 2\xi : \text{if } x_i < \xi \\ 0 : \text{otherwise} \end{cases} \tag{3.10}$$

or to restrict those whose feature x_i is equal or greater than ξ, we could define g_gp_i:

$$g\_gp_i(\hat{\mathbf{x}}, \mathbf{x}) = \begin{cases} +x_i - 2\xi : \text{if } x_i > \xi \\ 0 : \text{otherwise} \end{cases} \tag{3.11}$$

and then setting $\xi \in (0, \infty)$ as the restriction value. The expression $-x_i + 2\xi$ in the g_lp_i restriction yields a monotonically increasing restriction as the value progressively gets smaller than ξ, which might be helpful when minimizing the restriction functions. A similar reasoning applies to g_gp_i. If the restriction feature is categorical, we could define g_ep_i to restrict those instances whose feature x_i is different than v as follows:

$$g\_ep_i(\hat{\mathbf{x}}, \mathbf{x}) = \begin{cases} 0 : \text{if } x_i = v \\ 1 : \text{otherwise} \end{cases} \tag{3.12}$$

This restriction function yields 0 when $x_i = v$, and 1 otherwise. The restriction value in

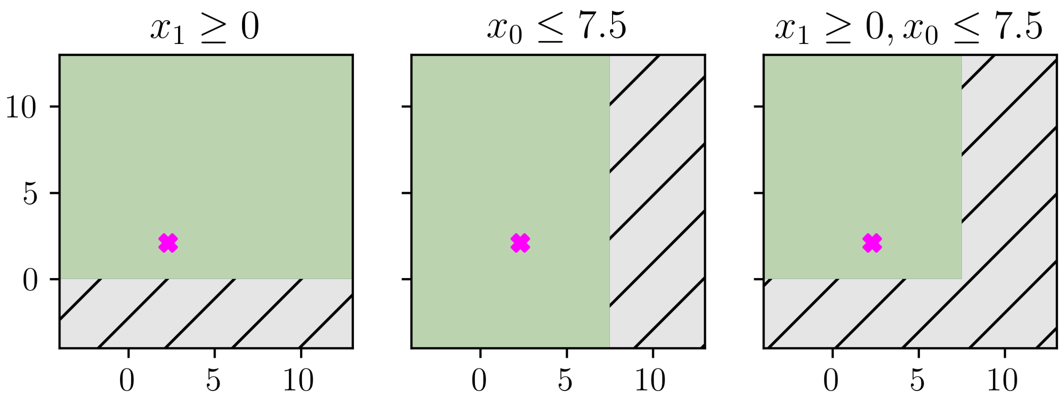

Figure 3.3: *Actionability illustration example. Green regions denote areas where the given restriction is satisfied, whereas gray-shaded areas indicate non-compliance. The pink cross represents the factual sample. Each figure title details the specific restriction.*

this case should be 1. Observe that these constraints can be made relative to the factual sample by setting ξ or ν to the value associated with the factual sample ($\hat{x}_i$).

In Figure 3.3, we can see three different feasible sets that result from applying three restrictions to the factual sample. The first restriction forces $\hat{x}_1 \geq 0$, the second $\hat{x}_0 \leq 7.50$, and the third combines these two restrictions.

Diversity can be enforced by penalizing observations close to specific regions of the input space (e.g., a region where an explanation was previously extracted). The diversity could be defined as follows:

$$g\_d(\hat{\mathbf{x}}, \mathbf{x}) = \omega(\mathbf{x}) \tag{3.13}$$

where ω is a function that penalizes observations close to a penalization point. Three examples of diversity restrictions are shown in Figure 3.4. The penalization function is the inverse of the Euclidean distance between an observation and a penalization point plus one (Mothilal *et al.*, 2020). The points used in penalization are $\mathbf{p}_1 = [5, 5]$, $\mathbf{p}_2 = [1, 1]$, and $\mathbf{p}_3 = [7, 5]$. From left to right, the first plot penalizes instances close to $\mathbf{p}_1$, the second to $\mathbf{p}_1$ and $\mathbf{p}_2$, and the third to $\mathbf{p}_1$, $\mathbf{p}_2$, and $\mathbf{p}_3$. It can be seen that the color darkens as we approach the penalization points, with the penalization points themselves having the maximum value. The central image shows two penalization points that are not close, but their combination increases the penalization in the line between them compared to surrounding regions (e.g., in opposite directions). The rightmost image demonstrates that closely positioned penalization points amplify the penalization value within their vicinity.

Finally, data manifold closeness can be defined similarly to the diversity measure:

$$g\_c(\hat{\mathbf{x}}, \mathbf{x}) = \rho(\mathbf{x}) \tag{3.14}$$

where ρ is a function that penalizes observations that are not close to the observations in

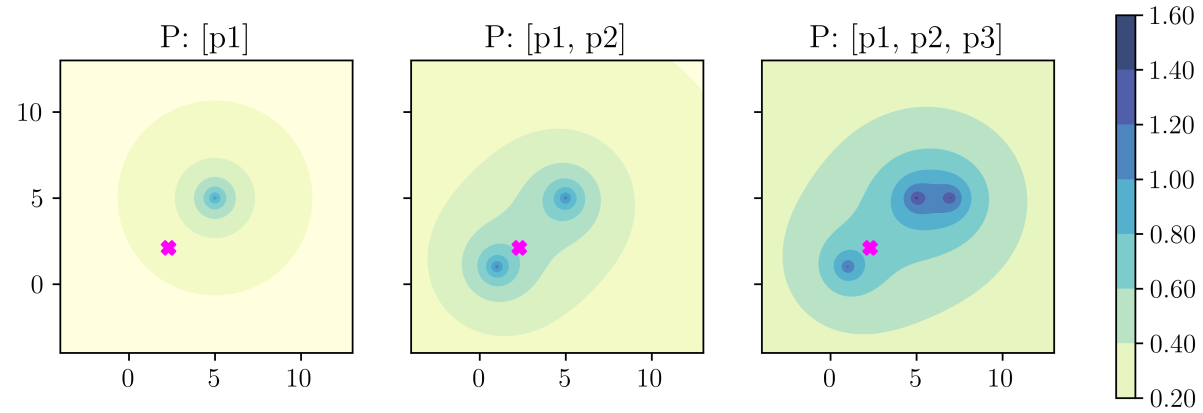

Figure 3.4: *Contour plot illustrating the diversity restriction. The pink cross represents the factual sample. The points penalized are shown in the title, being* $\mathbf{p}_1 = [5, 5]$*,* $\mathbf{p}_2 = [1, 1]$*, and* $\mathbf{p}_3 = [7, 5]$*. In the middle and right plots, the penalization is the cumulative effect of the individual penalizations. The color indicates the penalization value, from yellow (smallest penalization) to dark blue (biggest penalization).*

the training data. Examples of penalization functions in the literature are density estimation techniques (Poyiadzi *et al.*, 2020) and autoencoders (Dhurandhar *et al.*, 2018). In Figure 3.5, we show an example of data manifold closeness. Since we know the sampling distributions, we can use them directly to provide a data manifold closeness measure. Specifically, we use one minus the probability density function corresponding to the sampling distribution of the factual class. It can be seen how the value increases as we get far from the sampling distribution mean.

These previous definitions of restrictions make the composition of several restrictions straightforward by combining their individual effects. For instance, we can combine them by adding them together:

$$g\_a(\hat{\mathbf{x}}, \mathbf{x}) = g\_b(\hat{\mathbf{x}}, \mathbf{x}) + g\_c(\hat{\mathbf{x}}, \mathbf{x}) + g\_gp_i(\hat{\mathbf{x}}, \mathbf{x}) + g\_d(\hat{\mathbf{x}}, \mathbf{x}) \tag{3.15}$$

Another option is to combine them using the maximum:

$$g_m(\hat{\mathbf{x}}, \mathbf{x}) = max\{g\_b(\hat{\mathbf{x}}, \mathbf{x}),\ g\_c(\hat{\mathbf{x}}, \mathbf{x}),\ g\_gp_i(\hat{\mathbf{x}}, \mathbf{x}),\ g\_d(\hat{\mathbf{x}}, \mathbf{x})\}$$

which is equivalent to the intersection of their respective feasible sets. These combination procedures assume that the restriction value of the individual restrictions is the same. If it is different, this combination might yield unexpected results. To account for different values, we can multiply each of the restrictions by a scaling factor:

$$\alpha = \frac{r_{new}}{r_{old}}$$

where r_{old} is the original restriction value of the restriction and r_{new} is the restriction value

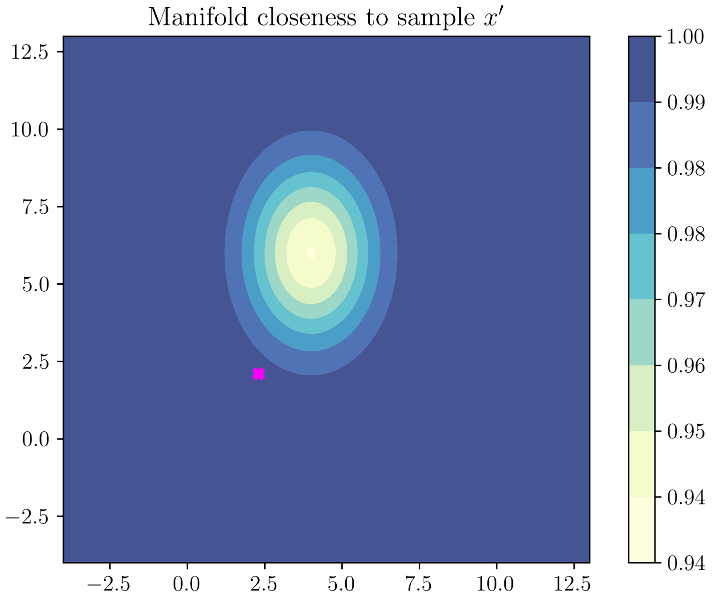

Figure 3.5: *Contour plot showing the data manifold closeness for the positive class. It is estimated directly using the probability density function of the sampling distribution. The color indicates the penalization value, from yellow (smallest penalization) to dark blue (biggest penalization).*

upon combined. Note that other scaling methods might be used.

Figure 3.6 shows the combination of the individual restrictions with the sum and max methods. All plots include the base distance and data manifold closeness restrictions. In addition, they include the penalization and actionability restrictions with the parameters in the column title.

The two approaches produce slightly different feasible sets. The sum combination, as visualized in the figure, is generally more difficult to understand since the exclusion of a point depends on the contributions of the individual restrictions. Conversely, a point is excluded in the max combination when one or more restrictions exclude it. Consequently, it is easier to understand and trace back to the individual restriction. In other words, it is easy to guess the max combination by looking at the individual restrictions, but the sum combination could be tricky.

Note that there is no one-size-fits-all solution. There are scenarios where the sum combination is more appropriate. For example, when combining multiple diversity penalizations, the max approach might not be ideal since it does not consider the collective effect of individual penalizations. In contrast, the sum combination does, as illustrated in Figure 3.4.

The combination procedure presented earlier, along with the concepts of diversity, actionability, and data manifold closeness, are examples to illustrate the benefits of the methodology. Other combination procedures (e.g., weighting the restrictions), new re-

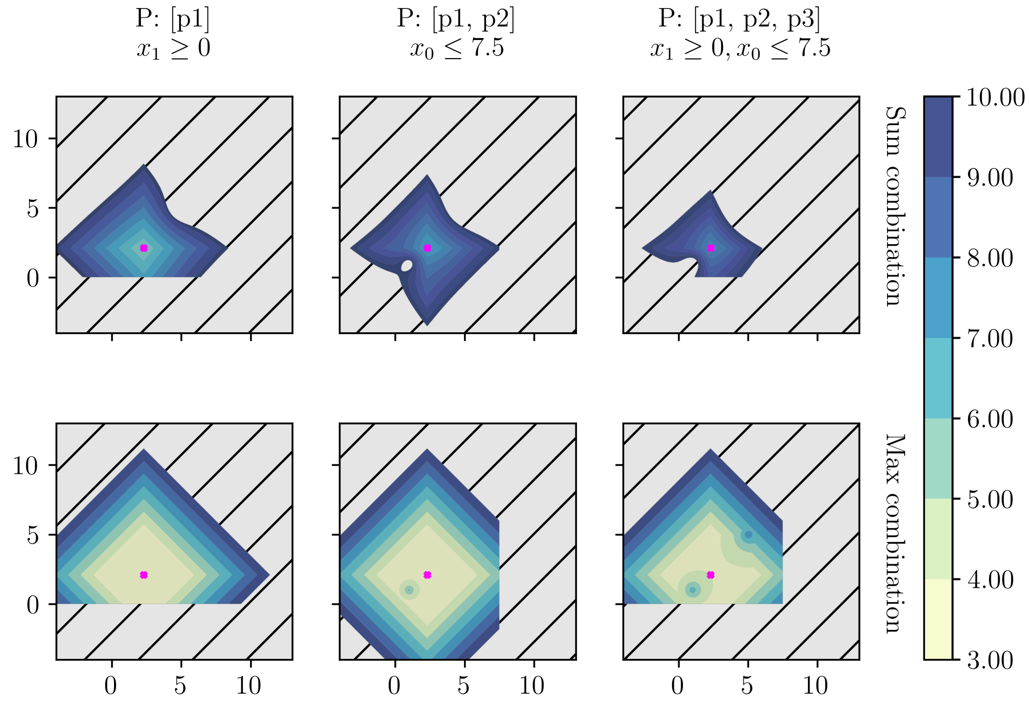

Figure 3.6: *Example of combining restrictions using the max and sum approaches. The first row shows the sum approach, while the second corresponds to the max approach. All figures incorporate both the base distance and manifold closeness restrictions. Subsequently, they integrate the diversity restriction, penalizing the points specified in the column title and the actionability restriction indicated in the column title. The penalization points are from the diversity example. The pink cross is the factual sample.*

strictions (e.g., transition penalization (Poyiadzi *et al.*, 2020)), or other implementations for diversity, actionability, and data manifold-closeness terms could be used.

3.4. Explanation Sets representation

There are different ways to represent the observations from an Explanation Set. For instance, enumerating their elements, but it becomes impractical when the number of elements is large or infinite. A more concise representation should be used for larger sets, which may require the Explanation Set to meet specific properties. The choice of representation depends on the problem at hand.

We present a simple taxonomy covering the representation of the current state-of-the-art observation-based explanation methods. These representations offer a more compact way of representing a set of observations than enumeration and can be used to represent an Explanation Set. The taxonomy is defined as follows:

- **Restrictive or non-restrictive**: Restrictive representations require the Explanation

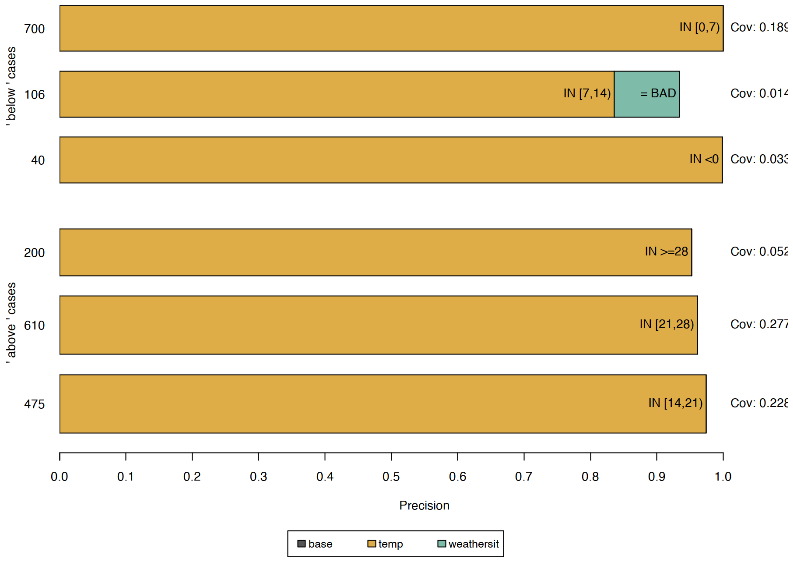

Figure 3.7: *Tabular anchor example. Each row provides an anchor explanation. The initial two rows correspond to instances where the number of rented bikes is below the average, while the next two rows represent above-average cases. Within each row, individual chunks signify restrictions on the features. The color indicates the feature. For continuous features, the restriction is denoted by a range, whereas for categorical features, it asserts the categories. The size of the chunk reflects the precision increment achieved by adding that restriction. Reproduced from Molnar (2018).*

Set to fulfill some properties (e.g., simply connected or restricted to a region of the input space), whereas non-restrictive representations can apply to any Explanation Set.

- **Exact or approximate**: Exact representations contain all the information to generate the original observations of the Explanation Set, while approximate representations do not.

Exact representations are often restrictive as they require the Explanation Set to satisfy certain properties. For example, rule-based explanations require the explanations to be represented by feature ranges, as seen in *LORE* (Guidotti *et al.*, 2018) and *Anchor* (Ribeiro *et al.*, 2018). Enumerating the elements of an Explanation Set is an exact representation that may be regarded as restrictive or non-restrictive, depending on whether finiteness is considered a restriction. Notably, by lifting some constraints, exact and restrictive representations can morph into approximate ones. For instance, in rule-based representations, the elements outside a rule can be discarded, or a rule that covers instances not initially defined in the Explanation Set can be used.

Figure 3.7 shows an anchor explanation example. It displays four cases: two cases where the number of bikes rented is below average and two above average. Taking as an example the second case, the anchor includes observations where the temperature

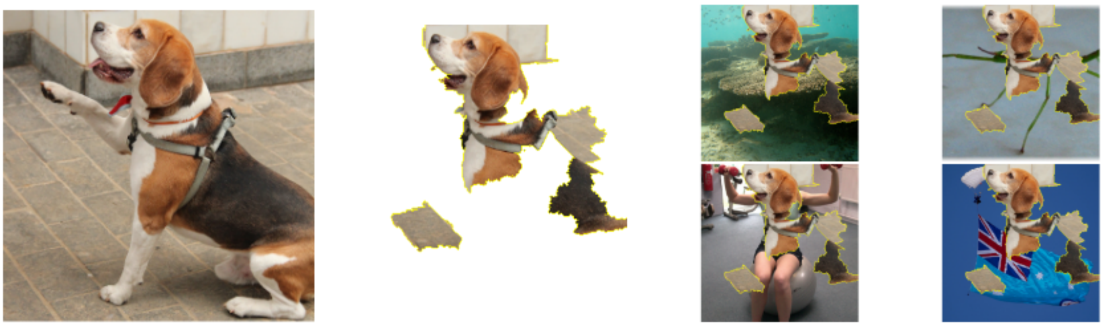

Figure 3.8: *Image anchors example. The left figure is the original image. The center image is the image anchor, with non-white pixels indicating fixed values. The white pixels can be changed, as shown in the right images, and the prediction will likely still be a beagle. Reproduced from Ribeiro et al. (2018).*

falls within $[7, 14)$ and the weather is bad, with no restrictions on the base feature. The precision (fidelity) is above 0.90, and 1.40% of observations in the dataset met the rule. The same reason can be applied to the other cases.

Image anchor (Ribeiro *et al.*, 2018) is an exact and restrictive representation for image explanations. In this method, some aspects of the image are fixed and cannot be changed, while others can take any value. This representation includes all valid fixed-pixel translations if the image anchors are invariant under translation.

Figure 3.8 displays an example of an image anchor. The image on the right is the original sample. The central image is the image anchor (center images), which shows the pixels used by the model to determine that the original image contains a beagle. If this image anchor is superimposed in a photograph, the model will predict "beagle" with high confidence. This behavior is exemplified in the images on the right. Note that the images might not make sense from our perspective, but the prediction remains the same.

Pertinent positives and pertinent negatives (Dhurandhar *et al.*, 2018) is a representation similar to image anchors, but in addition to the fixed attributes, it defines restrictions upon some unfixed attributes. The required factors to assert the outcome are pertinent positives, and those whose absence is required are pertinent negatives. While this representation can be defined with either a counterfactual Explanation Set or a semifactual Explanation Set, it is best defined using a counterfactual Explanation Set for the pertinent positives and a semifactual Explanation Set for the pertinent negatives. Thus, a single representation combines the information of two Explanation Sets.

Figure 3.9 showcases examples of pertinent positives and pertinent negatives. The leftmost image displays the prediction label above. The central image highlights the pertinent positive in light blue, representing the pixels required for the prediction. The image on the right depicts the pertinent negative; if the pink pixels were present, the prediction would change. For example, in the first row, the pixels of the pertinent positive

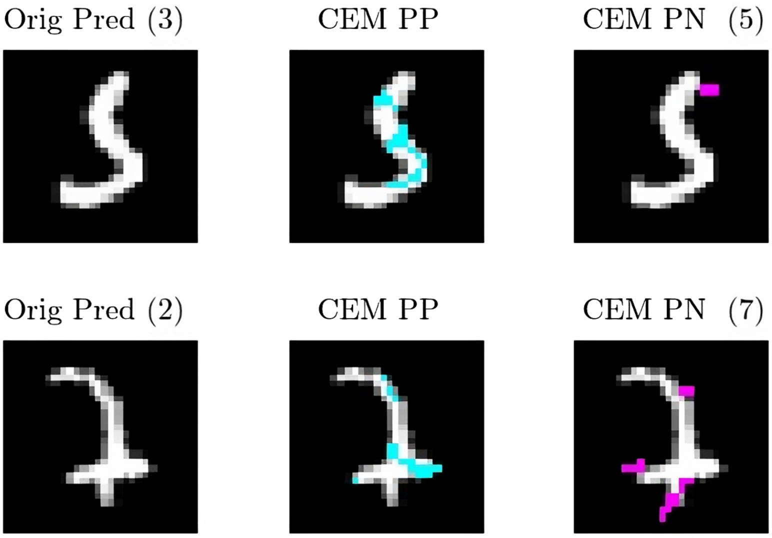

Figure 3.9: *Pertinent positives and negatives example. The leftmost column displays the original image, annotated with its prediction. The central column presents the pertinent positive, highlighting essential pixels in light blue. The rightmost column depicts the pertinent negative, marking in pink the pixels that, if present, would alter the prediction along with its prediction on top. Partial reproduction from Dhurandhar et al. (2018).*

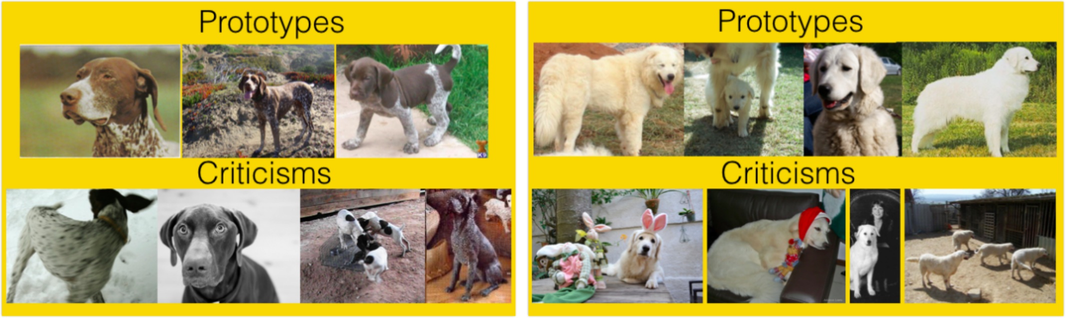

Figure 3.10: *Prototypes and criticisms example. The first row contains the prototypes, which represent standard dog photographs. The bottom row showcases criticisms, highlighting dog photographs that deviate from the dataset's norm. Reproduced from Kim et al. (2016).*

resemble the number 3, consistent with the prediction. Conversely, the pertinent negative pixels alter the original image to appear like the number 5.

Prototypes and criticisms (Kim *et al.*, 2016) are an example of non-restrictive and approximate representation. It explains a set of observations by providing the most average instances (prototypes) and instances not well represented by the prototypes (criticisms). This dual representation of data offers an enriching perspective, capturing both the central tendencies and the outliers of the data.

To illustrate this concept, Figure 3.10 provides an example of prototypes and crit-

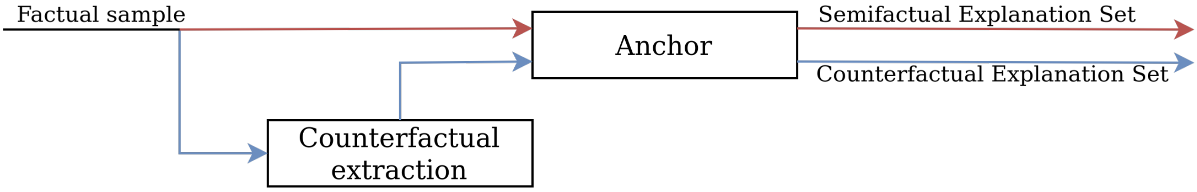
Factual sample
Anchor
Semifactual Explanation Set
Counterfactual Explanation Set
Counterfactual extraction

Figure 3.11: Anchor_ES *Explanation Set extraction workflow. In semifactual Explanation Sets (red path), the modified* Anchor *approach is directly applied. For counterfactual Explanation Sets (blue path), an intermediate stage involving the extraction of a counterfactual instance is required. This counterfactual instance is then used as input for the* Anchor *method to derive the counterfactual Explanation Set.*

icisms. The first row shows the prototypes, typical photographs of dogs with standard backgrounds. In contrast, the second row shows the criticisms, depicting dogs with particularities, such as black and white filters, accessories, or facing backward. These features make these instances less common than the prototypes in the dataset.

3.5. Generic method to extract Explanation Sets

This section introduces a model-agnostic method to extract Explanation Sets, called *Anchor_ES*. It is based on *Anchor*, recognized for its high-precision, model-agnostic explanations for individual predictions. Following the convention from Section 2.3.1, we use the term "*Anchor*" to refer to the extraction method and "anchor" to refer to the explanation. In essence, anchors define regions where predictions remain consistent with the factual sample. Within the Explanation Sets framework, anchors are a case of semifactual Explanation Sets: the grouping measure is the identity similarity and uses a rule-based representation. Note that the precision measure of *Anchor* is similar to the fidelity of Explanation Sets. *Anchor* does not specify restrictions for the feasible set (i.e., spans the whole feature space).

Figure 3.11 shows the *Anchor_ES* extraction workflow. For semifactual Explanation Sets, we directly apply the modified *Anchor* method, which will be explained in the following paragraphs. In counterfactual Explanation Sets, an additional intermediate step is required. We initiate the process by deriving a counterfactual $\mathbf{x}$ for the factual sample $\hat{\mathbf{x}}$. Then, we obtain a semifactual Explanation Set, denoted as S', for this newly generated $\mathbf{x}$. Since counterfactuals and semifactuals are complementary, the semifactual Explanation Set S' for $\mathbf{x}$ serves as a counterfactual Explanation Set for $\hat{\mathbf{x}}$.

First, we will outline the *Anchor* extraction method, highlighting potential modifications for the feature space restrictions (feasible set) and alternative similarity functions. Using the described *Anchor* method, we will explain the methodology for extracting counterfactual Explanation Sets later.

The *Anchor* method starts with a specific instance that requires an explanation and its prediction. First, it fits a discretizer with the training data. The data is discretized because searching for anchors in a discrete space is much more efficient than in a continuous space. Then, *Anchor* does an iterative beam search with beam size b. Starting with an empty rule (i.e., applies to all instances), the rules (candidates) are iteratively improved until its precision (fidelity) is greater than a parameter τ (usually set to 0.95).

At each iteration, the best-performing previous candidates are extended by adding one restriction per feature (e.g., $age < 3$ or $sex = F$). Subsequently, the best b candidates with the highest precision are selected using a multi-armed bandit method. This method reduces the number of predictions needed to estimate precision, generating perturbed samples for the estimation only when necessary. These predictions are made using a parameter called "prediction_fn", which conceptually serves as a proxy for the *ML* model. Upon the completion of this process, three possible scenarios arise:

1. If one or more rules exhibit precision greater than τ, the rule with the highest coverage is returned.
2. If all features have been considered but the best anchor does not meet the precision constraint, the method fails to identify an anchor.
3. When there are still features to consider and the precision is unmet, a new iteration starts.

Only two modifications are needed to adapt this method. The first and most important is the prediction function. Anchors contain mostly observations whose prediction is the same as the factual sample. As previously mentioned, for the evaluation of these observations, *Anchor* uses the prediction function, which is usually the model. However, we can provide alternative prediction functions that include restrictions. Thus, we define a prediction function that includes the feasible set and the grouping measure restrictions. The new prediction function is:

$$prediction\_fn(x) : \mathbb{1}[m(f(\hat{\mathbf{x}}), f(\mathbf{x})) = 1 \ and \ \hat{\mathbf{x}} \in S]$$

where m is a similarity-based grouping measure and S the feasible set. The prediction function equals 1 when the grouping measure is satisfied, and the observation resides within the feasible set; otherwise, it returns 0. It is important to highlight that, by definition, *Anchor* incorporates the factual sample in the explanation. Therefore, we include the requirement that the feasible set must contain the factual sample for this extraction method. Note that the factual sample meets both restrictions: the grouping measure (because it is a similarity) and the feasible set (as it is a prerequisite for the method).

The second modification is a minor optimization. In the initial sampling (perturbation), we filter out all observations that do not belong to the feasible set. In addition, if the remaining observations are fewer than a value, the difference is filled with uniformly sampled observations within the feasible set. This adjustment precedes the fitting of the *Anchor* discretizer, ensuring that a substantial proportion of the bins reside within the feasible set. Nonetheless, specific bins may lie outside the feasible set, necessitating the feasible set verification step within the prediction function. This modification restricts the search parameters for *Anchor* and minimizes the number of observations outside the feasible set processed by the prediction function, ultimately reducing the runtime.

We now delve into the generation of counterfactual Explanation Sets using the previously described method. The task of counterfactual extraction is posed as an optimization problem. The aim is to find an observation that meets the grouping measure (dissimilarity) and is close to the factual sample. We seek to minimize the restriction function in this method, which is usually the Manhattan distance. The objective function is defined as follows: If the observation is not in the feasible set or the grouping measure criterion is not met, the objective function yields a predetermined penalization value. Otherwise, it returns the restriction function for the observation.

To address this optimization, we employ the *Tree Parzen Estimators* (*TPE*) implementation from the Hyperopt library (Bergstra *et al.*, 2013). The search space is uniformly defined for both categorical and numerical variables. In this framework, categorical variables are specified by their respective categories, while numerical variables are defined by their range. Both are tailored to align the search space closely with the feasible set. The library also includes a warm-up dataset, wherein the perturbed dataset from *Anchor* and their corresponding objective function evaluations are supplied. The optimization is executed for 50 iterations using default parameters. Finally, the modified *Anchor* approach is applied to the identified counterfactual, yielding the final counterfactual Explanation Set.

Chapter 4

Random Forest Optimal Counterfactual Set extractor

This chapter introduces *Random Forest Optimal Counterfactual Set Extractor* (*RF-OCSE*), a method to extract counterfactual Explanation Sets from a *Random Forest* (*RF*) that contains the optimal (closest) counterfactual. The method is based on the partial conversion of a *RF* into a single *Decision Tree* (*DT*) by using a modified version of the *Classification and Regression Trees* (*CART*) algorithm. Within the Explanation Set framework, *RF-OCSE* can consider restrictions (feasible set) based on the Manhattan, Euclidean, Gower, and sGower (Fernández *et al.*, 2019) distances. The grouping measure is the identity similarity, and the representation of the counterfactual Explanation Sets is rule-based. This chapter addresses the following objective:

O4) To develop a method to extract these new explanations from a *RF* leveraging on its internal structure and axis-parallel decision surface.

Extracting counterfactuals from a *DT* is relatively simple. All leaves with a different class from the factual class are considered counterfactuals. Since rules define the decision surface of a *DT*, the optimal counterfactual can be obtained by searching for the rule that belongs to the counterfactual class and has the smallest distance. To calculate the distance between a sample and a rule, we determine the minimum distance between that sample and all other samples that satisfy the rule. Also, we can define a counterfactual Explanation Set using the rule that generates the counterfactual. If the counterfactual is optimal, the counterfactual Explanation Set containing this counterfactual is considered optimal.

This chapter is structured as follows. In Section 4.1, we introduce the notation needed to understand the proposal. We present a method to convert a *RF* into a *DT* to extract counterfactual Explanation Sets in Section 4.2. In Section 4.3, we describe how the fusion can be improved by pruning the resulting DT. Finally, Section 4.4 details how counterfactual Explanation Sets can be extracted without fully fusing a RF.

4.1. Notation

The following notation will be used throughout the rest of this chapter. Let a RF be defined as a set T of DTs. Let R be a set of rules of all DTs in a RF, that is, the paths from the root of a DT to the leaves, and r a given rule. The set R has at least one rule of each DT. A DT, t, is defined by a set of rules. Let $t(R)$ be the subset of rules in the set R that belongs to the DT t. Let L be the set of classes in the problem and l a given class. A condition c is defined as a feature $s(c)$, a comparison type $cmp(c)$ ($\leq$ or $>$) and a threshold value $v(c)$. Let $r(c)$ be the rule that contains the condition c. Let C be the set of all conditions. Let a rule r be defined as a set of conditions $C(r)$ and the probability $p_l(r)$ for each class $l \in L$. Let $C(R)$ be the subset of conditions of C in the rules R, and let C_s be the subset of conditions C whose feature is s. Finally, let f be a RF model and d a distance function.

4.2. Fusion of Random Forest tree predictors

The fusion of the RF tree predictors aims to represent a RF as a single DT without modifying the decision surface. The fusion procedure starts with a DT from the RF. Next, every leaf of this tree is substituted with another DT from the RF. As this happens, the leaves of the initial tree are propagated to the newly introduced leaves. In other words, when a DT replaces a leaf, it is merged with each leaf of that DT. This iterative process continues until all DTs from the RF are integrated. The predictions of the rules in the resultant DT are determined by computing the average probability across the combined leaves of that rule.

Figure 4.1 shows an example of this fusion method for a RF with two DTs. In the fused DT, it can be seen that some leaves are unreachable because two conditions of their rule are exclusive. The leaves 2, 3, 5, 6, 7, and 10 can not be reached because they have two exclusive conditions, $s_1 \leq 0$ and $s_1 > 0$. The simplified DT without the unreachable paths is depicted in Figure 4.2.

As the number of DTs in a RF increases, the number of unreachable nodes increases exponentially, increasing computational time and memory consumption. Moreover, the sequence in which DTs are chosen can significantly influence the dimension of the resulting DT, potentially rendering the problem intractable.

To minimize the dimensions of the resulting DT, a fusion method that leverages in a

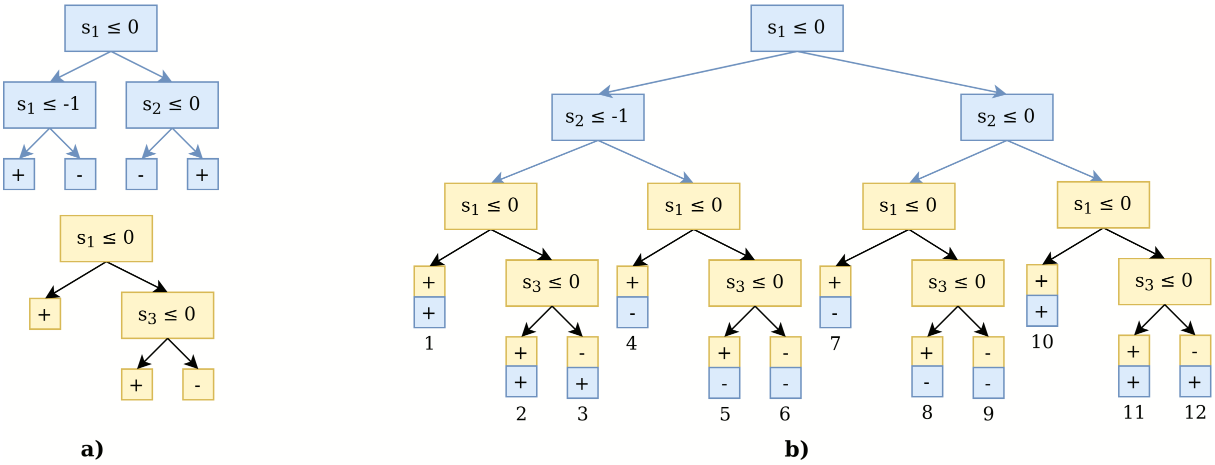

Figure 4.1: *Example of a simple method to fuse a* RF *with two* DT*s (a) into a single* DT *(b).*

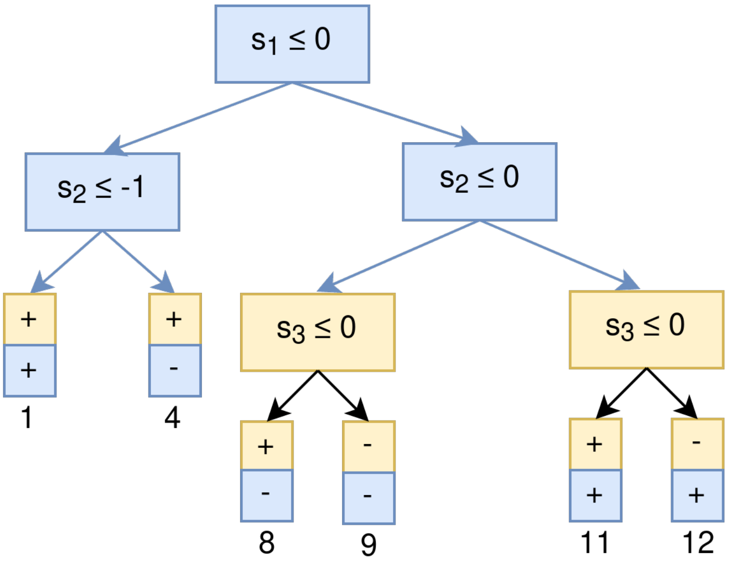

Figure 4.2: *Simplified* DT *from the example in the Figure 4.1. The number in the leaves indicates the origin of the leaves in the full* DT*.*

variant of the *CART* algorithm has been proposed. This method takes as input a set of *DT*s and outputs the corresponding *DT*.

Within this fusion framework, each observation symbolizes a rule extracted from the *DT*s. Each feature within an observation corresponds to a set of conditions associated with that feature. Such a set can be empty (no constraints on the feature), it could comprise a singular condition (e.g., $s_1 \leq 0$ or $s_1 > 0$), or even two conditions (e.g., $s_1 > 0$ and $s_1 \leq 1$). The most restrictive condition is kept when a rule has multiple conditions of a similar kind (like "less or equal than" and "greater than"). For instance, among the conditions $s_0 > 0$, $s_0 > 1$, and $s_0 > 2$, the condition $s_0 > 2$ is deemed the most restrictive.

The splits are exclusive in the original *CART* algorithm, and consequently, an obser-

Algorithm 1 partition_calculation

```
Require: $(R, s, v)$
  $R_l, R_r \leftarrow R$
  for $c \in C(R)_s$ do
    if $cmp(c)$ = '>' and $v(c) \leq v$ then
      $R_l \leftarrow R_l \cup \{r(c)\}$
    else if $cmp(c)$ ='≤' and $v(c) > v$ then
      $R_r \leftarrow R_r - \{r(c)\}$
    end if
  end for
  $R_l \leftarrow$ add_condition($R_l$, $s \leq v$)
  $R_r \leftarrow$ add_condition($R_r$, $s > v$)
  return $R_l, R_r$
```

vation can only belong to one partition. However, rules might not have a condition for a feature, and certain conditions could be satisfied on both partition sides. For instance, the condition $s_1 \leq 2$ is satisfied on both sides of the split $s_1 \leq 1$. Therefore, when using rules as observations, the splits are not exclusive.

The partitions are determined using Algorithm 1. Taking the rules R from the current partition, a feature s, and a threshold value v as inputs, the algorithm calculates the resulting partition R_l and R_r (left and right sides, respectively). The algorithm initializes by equating the resultant partitions R_l and R_r to the complete partition R. Subsequently, the algorithm traverses all conditions within the partition for the specified feature $C(R)_s$. In the left partition, $(s \leq v)$, all rules with a condition c such that $v(c) > v$ are filtered. Conversely, on the right partition, $(s > v)$, all rules with a condition c such that $v(c) \leq v$ are filtered. As highlighted earlier, any “less or equal than” condition with a value surpassing the partition threshold is valid for both sides. The same applies to “greater than” conditions with values beneath the threshold. Finally, the add_condition method adds to every rule in R_l, the condition $s \leq v$, and the condition $s > v$ to each rule in R_r, ensuring that the rules accurately define the partition boundaries.

The *CART* algorithm computes the Gini impurity based on the weighted average of the Gini impurity of each partition, which is based on the class probabilities of the partition. In our modified *CART* algorithm, the class probability of the partition needs to be adapted to align with the aggregation method of the *RF*. Initially, we determine the average probability associated with each *DT* within a partition:

$$P_{l,t}(R) = \frac{\sum_{r \in t(R)} p_l(r)}{\#t(R)} \tag{4.1}$$

Here, $\#t(R)$ denotes the number of rules in set R associated with the *DT* t. Given that at least one rule from every *DT* exists within a partition, this probability is always

Algorithm 2 select_split

```
Require: $(R)$
    $best\_s, best\_v$
    $best\_score \leftarrow \infty$
    for $c \in C(R)$ do
        $R_l, R_r \leftarrow partition\_calculation(R, s(c), v(c))$
        $score \leftarrow gini\_impurity(R_l, R_r, R)$
        if $score < best\_score$ then
            $best\_score \leftarrow score$
            $best\_s \leftarrow s(c)$
            $best\_v \leftarrow v(c)$
        end if
    end for
    return $best\_s, best\_v$
```

defined. Then, the overall probability estimation for the partition is derived as the mean of the probabilities from all DTs:

$$P_l(R) = \frac{\sum_{t \in T} P_{l,t}(R)}{\#T} \tag{4.2}$$

In this context, $\#T$ represents the number of DTs. Using this overall probability estimation for the partition, the Gini impurity can be calculated like in a single DT.

The method for selecting splits, as outlined in Algorithm 2, employs an exhaustive search strategy. For every condition present in the rules of the partition, the algorithm divides the partition into R_l and R_r using Algorithm 1 and computes the Gini impurity. The outcome of this algorithm is the split that yields the lowest Gini impurity.

The fusion method, detailed in Algorithm 3, mirrors the $CART$ approach, adopting a recursive partitioning mechanism. The termination criterion for recursion is only met when a single rule from each DT is left, i.e., $\#R = \#T$. This scenario is always achievable since the rules derived from a DT span the entire feature space and are inherently non-overlapping. Hence, any chosen split from the rules in the partition (R) is always satisfied by at least one rule of each DT. At every step, if the termination criterion is not met, the algorithm selects a split using Algorithm 2. Then, the algorithm is invoked recursively on the partitions (R_l and R_r) determined by Algorithm 1. The fusion method returns a binary tree structure, representing the same decision surface as the input DTs.

4.3. Partial Random Forest to Decision Tree fusion

The primary limitation of the proposed fusion technique is the size of the resulting DT. As the number of DTs in a RF and their depth grows, the number of nodes in the

Algorithm 3 rf_to_dt

```
Require: $(R)$
  if $\#R = \#T$ then
    $node \leftarrow \{l = P_l(R) : l \in L\}$
  else
    $s, v \leftarrow select\_split(R)$
    $R_l, R_r \leftarrow partition\_calculation(R, s, v)$
    $node_l \leftarrow rf\_to\_dt(R_l)$
    $node_r \leftarrow rf\_to\_dt(R_r)$
    $node \leftarrow (s, v, node_l, node_r)$
  end if
  return node
```

converted DT increases exponentially. In the worst-case scenario, where the DTs have no shared features, the corresponding DT for a RF with N DTs of depth m would have a depth of $N \cdot m$ and $2^{N \cdot m+1} - 1$ nodes.

The fusion approach described in Algorithm 3 replicates the decision boundary and probabilities of a RF. However, counterfactual extraction mechanisms only use the leaf classes. To calculate the leaf classes, the fusion technique only requires a partial conversion of the RF, significantly reducing the node count in the derived DT. This reduction can be achieved by changing the stopping criterion and halting once the class is determined. In such cases, adding further splits will not alter the class. This is evident when all rules in the partition are of the same class or when most DTs have only a rule, with the other DTs lacking the influence to alter the final decision.

The stopping criterion can employ a worst-case scenario for partitions with unset DTs (a DT with multiple rules in the partition). The rule that defines this scenario is calculated as follows:

$$r^*_{t,l,z}(R) = \underset{r \in t(R)}{\arg\max}\,(p_z(r) - p_l(r)) \tag{4.3}$$

For a given DT, t, the worst-case rule denotes the maximum probability disparity between a class z and the expected class l of the partition R.

The definition can be extended for a RF to represent the mean of the worst-case rule probabilities across all DTs. Let

$$W_{l,z}(R) = \frac{\sum_{t \in T} p_l(r^*_{t,l,z}(R))}{\#T} \tag{4.4}$$

be the mean probability of the worst-case rules for the expected class l against class z within the partition R. The metric $W_{l,z}(R)$ is a lower bound of the probability $P_l(R')$ against class z, where R' is a subset of R. Similarly to R, the subset R' contains at least

Algorithm 4 early_stop

```
Require: $(R)$
    $l \leftarrow \arg\max_{l \in L} P_l(R)$
    for $z \in L : z \neq l$ do
        if $W_{l,z}(R) \leq B_{z,l}(R)$ then
            return FALSE
        end if
    end for
    return TRUE
```

one rule from each DT. Let

$$B_{z,l}(R) = \frac{\sum_{t \in T} p_z(r^*_{t,l,z}(R))}{\#T} \tag{4.5}$$

be the mean probability of the best-case rules for another class z against the class l within the partition R. $B_{z,l}(R)$ acts as an upper bound for the probability $P_z(R')$ against class l.

The early stop technique, described in Algorithm 4, checks if the partition class is predetermined, leveraging the worst and best case scenarios estimates previously discussed. This method only checks the most likely class l in the partition. Consequently, only the class with the highest likelihood of being the partition class undergoes evaluation, reducing the execution time. For every class z where $z \neq l$, the method, under the worst-case scenario, verifies that $W_{l,z}(R) > B_{z,l}(R)$. If all tests are satisfied, the partition class is deduced as l. If not, the partition class can not be estimated using the worst-case strategy.

4.4. Partial counterfactual set extraction

While the partial fusion technique does reduce the size of the resultant DT, it still becomes intractable as the count and depth of DTs increase. Notice that each leaf with a class different than the factual class qualifies as a counterfactual, establishing a direct link between the size of the DT and the volume of counterfactuals. Typically, users expect a few counterfactual explanations and iterate through them if they do not meet their expectations. Thus, these counterfactuals are sorted to give users the most relevant explanations according to a distance.

The counterfactual extraction technique extracts counterfactuals from the fused DT. However, extracting counterfactuals during the RF merging process is feasible. Combining the fusion and counterfactual extraction techniques can produce a counterfactual whenever a partition class is determined and different from the factual class. This eliminates the need for an explicit memory representation of the fused DT, which could be memory-intensive. However, the order of the generated counterfactuals remains arbitrary and does not adhere to the designated distance. Consequently, to sort them, the extraction method must compute all counterfactuals, leading to a full *RF* conversion and the inherent size challenges of the fusion.

To avoid the computation of every potential counterfactual, the fusion method can be adapted to prune paths where the distance to the factual sample surpasses the distance to the nearest identified counterfactual (similarly to a branch and bound method). As a result, the fusion becomes more restrictive as closer counterfactuals are identified, mitigating the size problem. A drawback of this approach is the need to partially merge the *RF* for each counterfactual extraction, in contrast to a unique complete fusion. However, only the former is feasible when working with large *RF*s.

For the estimation of the initial counterfactual, we consider two strategies:

- **Minimum Observable (MO)**: Selects the closest instance in the training dataset corresponding to a counterfactual class. It is the base strategy.
- **Hot Start (HS)**: Performs the *MO* strategy in a dataset consisting of the union of the training set, the closest counterfactuals found for other instances, and an artificial dataset. This artificial dataset comprises "perturb_size" instances artificially generated by perturbing instances from the dataset. These instances are randomly selected with replacement from the dataset. The perturbation involves adding noise from a distribution $\mathcal{N}(0, \sigma)$ to "n_features" randomly selected features.

The performance of the proposed approach can be augmented by centering the fusion on the factual sample. Rules whose distance to the factual sample exceeds a specified threshold can be discarded. This distance is determined by measuring the proximity of the nearest sample from the rule to the factual sample. Like path pruning, the threshold is the distance to the nearest identified counterfactual. This strategy facilitates the construction of a *DT* that locally replicates a *RF*, ensuring that rules defined within that locality remain intact.

The extraction procedure, described in Algorithm 5, extracts a rule set that satisfies the optimal counterfactual from a *RF*. The algorithm takes as input a rule set $\boldsymbol{R}$, a *RF* model f, a distance d that can be either the Manhattan, Euclidean, Gower, or sGower distances, the factual sample $\hat{\mathbf{x}}$, the updated factual sample reflecting the current state $\mathbf{x}$, and the maximum allowable distance for the extraction of the counterfactual set, denoted as max_d.

The *RF-OCSE* method starts by pruning rules, keeping only those in $\boldsymbol{R}$ that are within a distance of max_d from the factual sample $\hat{\mathbf{x}}$, yielding $\boldsymbol{R}_{prune}$. Subsequently, the method employs the *early_stop* algorithm to check if the class has been determined. If so, the

Algorithm 5 *RF-OCSE*

```
Require: $(R, f, d, \hat{\mathbf{x}}, \mathbf{x}, max\_d)$
    $R', found \leftarrow \{\}, FALSE$
    $R_{prune} \leftarrow prune\_rules(R, max\_d, \hat{\mathbf{x}})$
    if $early\_stop(R_{prune})$ and $f(\hat{\mathbf{x}}) \neq f(\mathbf{x})$ and $d(\hat{\mathbf{x}}, \mathbf{x}) < max\_d$ then
        $R', found \leftarrow R_{prune}, TRUE$
        $max\_d \leftarrow d(\hat{\mathbf{x}}, \mathbf{x})$
    else
        $s, v \leftarrow select\_split(R_{prune})$
        $R_{left}, R_{right} \leftarrow partition\_calculation(R_{prune}, s, v)$
        if $\mathbf{x}_s \leq v$ then
            $R_{meet}, R_{not\_meet} \leftarrow R_{left}, R_{right}$
        else
            $R_{meet}, R_{not\_meet} \leftarrow R_{right}, R_{left}$
        end if
        $R', max\_d, found \leftarrow RF\text{-}OCSE(R_{meet}, f, d, \hat{\mathbf{x}}, \mathbf{x}, max\_d)$
        $\mathbf{x}_{not\_meet} \leftarrow update(x, s, v)$
        if $d(\hat{\mathbf{x}}, \mathbf{x}_{not\_meet}) \leq max\_d$ then
            $R'_{not\_meet}, max\_d, found_{not\_meet} \leftarrow RF\text{-}OCSE(R_{not\_meet}, f, d, \hat{\mathbf{x}},$
                                        $\mathbf{x}_{not\_meet}, max\_d)$
            if $found_{not\_meet}$ then
                $R', found \leftarrow R'_{not\_meet}, TRUE$
            end if
        end if
    end if
    return $R', max\_d, found$
```

method verifies if it is a counterfactual and whether its distance is under the threshold max_d. If these criteria are met, the rule set R' that defines the counterfactual is returned.

If the class can not be determined, the method selects a split c, separating R_{prune} into R_{meet} (rules that meet) and R_{not_meet} (those that do not). Given its proximity to the factual sample, a recursive call is first made with R_{meet}. The method then updates the distance max_d and keeps the returned rule set in R'. The method modifies c not to meet the split, resulting in c_{not_meet}, considering the feature type. The method then checks the distance between c_{not_meet} and the factual sample against max_d. If it is lower, a recursive call is made with R_{not_meet}. If a valid counterfactual emerges from this, R' is updated. The method concludes by returning R', max_d, and a flag indicating if a counterfactual was found.

The optimal counterfactual Explanation Set is derived by combining rules R' present when the optimal counterfactual was found. This combination requires selecting the most restrictive conditions from the rules for every feature. Notably, while the early stop method can determine a class even with multiple active rules in a tree, an observation

Algorithm 6 *greedy-rule-selection*

```
Require: $(R, p_c, y')$
    $R' \leftarrow \{\}$
    $R_s \leftarrow R \textit{ sorted by descending } p_{y'(r)}$
    $i \leftarrow 0$
    $acc_{prob} \leftarrow 0$
    while $i < \#R_s$ and $acc_{prob} < p_c$ do
        $acc_{prob} \leftarrow acc_{prob} + p_{y'}(R_s[i]) \,/\, \#R_s$
        $R' \leftarrow R' \cup \{R_s[i]\}$
        $i \leftarrow i + 1$
    end while
    return $R'$
```

can only meet a singular rule in each tree. Hence, rules not aligning with the optimal counterfactual are omitted from the rule combination.

In order to provide broader counterfactual Explanation Sets (higher coverage), the number of rules considered in the rule combination process can be reduced. As mentioned in Section 3.1, a higher coverage results in easier-to-understand explanations. The goal is to consider the least number of rules that guarantee the outcome matches the counterfactual class. A *greedy-selection* strategy, detailed in Algorithm 6, is considered. This algorithm takes the returned rule set from the *RF-OCSE* extraction, which the optimal counterfactual $\mathbf{x}'$ satisfies, alongside the desired probability threshold p_c and the counterfactual class y', where $y' = f(\mathbf{x}')$.

The method *greedy-rule-selection* starts by arranging the rules r in R in a descending sequence based on their probability for the label y'. Sequentially, it traverses the sorted rules, incorporating them into the resultant rule set, R', until the cumulative probability surpasses the threshold p_c. The output of the method is R'. The probability threshold p_c is typically designated as 0.50, the minimum probability that ensures the label prediction. Nonetheless, other thresholds can be explored. This is particularly relevant in scenarios like multi-class or imbalanced classification. Moreover, if p_c is set below 0.50, it can be employed to generate counterfactual Explanation Sets whose fidelity is lower than 1.

Chapter 5

Experiments

This chapter evaluates the proposed explanation methodology, Explanation Sets, and explanation extraction methods, *Anchor_ES* and *Random Forest Optimal Counterfactual Set Extractor* (*RF-OCSE*). The evaluation comprises two main sections. Section 5.1 details how the Explanation Set methodology can enhance existing counterfactual and semifactual explanations through two use cases. These explanations are extracted using *Anchor_ES*. Then, Section 5.2 compares *Anchor_ES* and *RF-OCSE* against state-of-the-art alternatives in counterfactual Explanation Set extraction, highlighting the benefits of the proposed methodology. This chapter addresses the following objective:

O5) To validate the proposed explanation methodology and compare the extraction methods to alternatives in the literature.

Explanation Sets are validated with several metrics calculated using 10-fold cross-validation. The explanations and their associated quality metrics are generated in the test partitions, simulating explanations on unseen instances. For clarity, we will refer to counterfactual Explanation Sets with one element as counterfactuals in this chapter. The metrics listed below are common to all experiments, though specific metrics will be detailed.

- **Coverage**: The coverage is the percentage of observations from the training feasible set (i.e., the intersection between the training set and feasible set) included in the Explanation Set. A higher coverage is desirable because it implies that the explanation is more generic and aligned with a higher data manifold closeness.
- **Fidelity**: The fidelity is estimated as the proportion of samples in the Explanation Set that meet the grouping measure. Higher fidelities are also preferred because otherwise, the explanation would not be faithful. The fidelity is estimated using uniformly sampled data.

- **Distance**: Individual counterfactuals are evaluated in terms of distance to the factual sample. A smaller distance is preferred since it indicates that the observations are more similar and, consequently, easier to understand.

5.1. Explanation Sets experiments

Besides the base metrics specified previously (fidelity and coverage), the complexity of the explanations is evaluated through their number of conditions. This metric is defined as the number of features with restrictions in the explanation. A lower number of conditions is desirable because the explanation will be easier to understand and, generally, have a higher coverage. Finally, individual counterfactuals are evaluated in terms of number of changes. This measure is the number of features that differ between the factual sample and another observation. Similarly to the number of conditions, it measures the complexity of the explanation. Counterfactuals with few feature changes (high sparsity) are preferred because they are easier to understand.

5.1.1. Regression case study

In this case study, we consider the Concrete Compressive Strength dataset (Yeh, 2007) from the UCI repository (Dua & Graff, 2017). This dataset consists of 1030 instances. The goal is to estimate the concrete compressive strength (MPa), a highly nonlinear function of its age and ingredients. The dataset includes the age (days) and seven ingredients (kg in a m^3 mixture). The output variable, the concrete compressive strength, ranges from 2.30 to 82.60 MPa. We will refer to the compressive strength of the factual sample as $\hat{h}$ and the compressive strength of any other observation as h'.

The primary objective of this case study is to demonstrate the extension of semifactuals and counterfactuals to other tasks, specifically regression, and to show how the explanations vary depending on the selected grouping measure. The representations considered in this experiment for counterfactual and semifactual Explanation Sets (following the taxonomy in Section 3.4) are rule-based (restrictive and approximate), a combination of rule-based explanations (restrictive and approximate), and finite enumeration (exact and non-restrictive).

Two feasible sets definitions are considered. The restriction function of the base feasible set is the sGower distance (Fernández *et al.*, 2019) parametrized with the Manhattan distance for numeric variables and simple-matching for categorical variables. The second feasible set includes a combination of the sGower distance and a data manifold closeness restriction. This restriction uses the one-class Support Vector Machine (SVM) implementation from *Scikit-Learn* (Pedregosa *et al.*, 2011) with default parameters. Only observations not classified as outliers belong to the feasible set. Specifically, the restriction function evaluates non-outliers as 0 and a value greater than the restriction value for outliers. The restrictions are combined using the sum method (the max method would produce the same results in this case) outlined in Section 3.3.

The restriction function is minimized to extract individual counterfactuals (i.e., the closest non-outlier counterfactual). This process is analogous to iteratively reducing the restriction value each time a counterfactual is found until no further counterfactuals can be identified.

First, we evaluate semifactual Explanation Sets explanations. These explanations are extracted using a similarity, $sr_k(\hat{h}, h')$, that considers two compressive strengths, $\hat{h}$ and h', to be equal when their absolute difference is less than k (see Eq. 3.9).

The explanations are extracted using $k = 5$ and $k = 10$. It is worth noting that in problems with a real-valued output (e.g., regression or probabilities), it does not make sense to group only elements whose prediction is identical (i.e., identity similarity). Besides, there might be no difference between two close predictions from the user's perspective, or the *Machine Learning* (*ML*) model might not be sensitive enough to discriminate at such precision. For instance, in this use case, where the average gap between the sorted target output is 0.07, there is probably no noticeable difference between 50.01 and 50.005. The models obtained an average mean squared error over all folds of 18.44, and consequently, it is not sensitive enough to discriminate at such precision or even a few units.

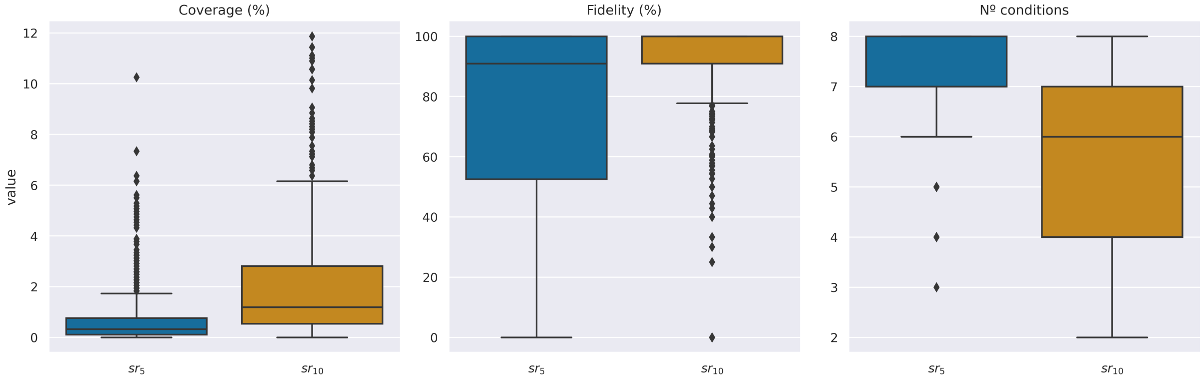

Figure 5.1: *Semifactual Explanation Sets coverage, fidelity, and number of conditions for each similarity. Higher values in coverage and fidelity are preferred, and lower values in the number of conditions.*

Figure 5.1 shows a box plot for each metric and similarity. It can be seen that the semifactual-based explanations using $k = 10$ are better in all metrics than those obtained using $k = 5$. These results are not unexpected because the set of all semifactuals using sr_5 is a subset of sr_{10}. Thus, the results using sr_{10} should be at least as good as sr_5. The percentage of valid explanations (meeting the fidelity requirement) is low, 74.17% and 44.76%, for sr_{10} and sr_5, respectively. The low success rate can be attributed to the very restrictive nature of the grouping measure and the limitations of Anchor in imbalanced problems that will be mentioned in Section 5.1.2.

The average coverage is low, 2.02% and 0.67% for sr_{10} and sr_5, respectively. However, the average coverage in real-valued problems will be far smaller than that of a binary classification problem. Consider the maximum theoretical coverage in the dataset, which is 32.34% and 15.96% for sr_{10} and sr_5, respectively. Conversely, in a balanced classification problem using the identity similarity, the maximum theoretical coverage is 50.00%. This difference might be even bigger if a lower radius is required or the label distribution is more spread.

The low maximum theoretical coverage also has implications on the high variance of the fidelity and the number of conditions. Given the factual sample, $\hat{\mathbf{x}}$, the grouping measure transforms the *ML* model in a binary classifier, where an observation $\mathbf{x}$ is grouped with $\hat{\mathbf{x}}$ if its prediction is 1 or not grouped if 0. Thus, a low maximum theoretical coverage implies more 0s than 1s. Consequently, the problem is imbalanced, which occurs when the number of instances across the labels is unequal. The degree of imbalance can be measured using the imbalance ratio, which is the number of instances of the majority class divided by the number of instances of the minority class. A high imbalance ratio, where the minority class is the grouping class (1), affects the quality of the explanation extracted with the current Explanation Set extraction method based on Anchor. This problem could be partially addressed using an imbalance-aware sampling procedure, generating new observations for the minority class using synthetic methods, or increasing the bin count in the Anchor discretizer.

Regarding the number of conditions, having low coverage indicates that it is likely that the number of conditions is high or that the conditions are very restrictive. This dependence arises from high coverages being only achievable when the rules are broad and cover several instances from the dataset.

Note that higher values across all metrics do not necessarily imply that these explanations are better from the perspective of the user because it depends on which is an acceptable k for the similarity. The choice of the k parameter depends on the domain and the accuracy of the model.

The representation for the semifactual Explanation Sets is rule-based. As an example, we present an explanation for the observation with values: cement=540.00, blast furnace slag = 0.00, fly ash = 0.00, water = 162.00, superplasticizer = 2.50, coarse aggregate = 1040.00, fine aggregate = 676.00, and age = 28.00. The rule-based semifactual Explanation Set using the similarity sr_5 is the following: cement > 350.00, water ≤ 164.90, age > 7.00, fine aggregate ≤ 734.15 and 0.00 < superplasticizer ≤ 6.50. Using the similarity sr_{10}, the explanation is: cement > 349.00, water ≤ 165.60, age > 14.00, fine aggregate ≤ 733.50, 0.00 < superplasticizer ≤ 6.30, and fly ash ≤ 0.00.

Then, we evaluate counterfactual explanations without manifold restrictions. These explanations are represented using finite enumeration and extracted with two dissimilarities. First, we consider the similarities sr_5 and sr_{10} converted into dissimilarities, sr'_5 and sr'_{10}, similarly to Eq. 3.4. Then, we consider a dissimilarity, $gt_{vo}(\hat{h}, h')$, where a element h' is grouped with $\hat{h}$ if h' is greater than $\hat{h}$ plus a value o (see Eq. 3.7).

The dissimilarity gt_{vo} is considered with $o = 0$ and $o = 5$. Figure 5.2 depicts the distance and the difference in the prediction between the factual sample and the counterfactuals extracted for each dissimilarity. Thus, we can visually determine if the extracted counterfactuals are valid and if there is a correlation between the actual prediction, the prediction difference, and the distance. The method extracted counterfactuals in all cases except 10 instances using the dissimilarity $gt_{v,5}$ and one in $gt_{v,0}$. The prediction of the factual samples corresponding to the invalid counterfactuals is contained in the range $[76.84, 80.52]$, close to the maximum value in the dataset 82.60. The method did not generate valid counterfactuals because predicting values higher than the maximum value in the training set might be hard or even impossible (e.g., *Decision Tree* (*DT*) or *Random Forest* (*RF*)).

Consequently, counterfactual explanations might not always be possible for a given observation and dissimilarity measure. If a counterfactual cannot be found, then the *ML* model is constant (because, in this case, the feasible set is the feature space). Thus, their respective semifactual Explanation Sets (converting the dissimilarity into a similarity) are not helpful. Rather, the most significant explanation is the absence of counterfactuals. In all other instances where counterfactuals were generated successfully, their corresponding semifactual Explanation Sets are meaningful explanations.

There is a slight downward trend in the counterfactuals extracted using the dissimilarities sr'_5 and sr'_{10}. Thus, the counterfactuals extracted from observations with high values tend to have a lower value, and those with low values have a high value. This is explained by the fact that it is easier to find observations whose prediction is closer to the mean of the labels because its distribution is bell-shaped. A similar phenomenon can be seen in the counterfactuals extracted using the dissimilarities $gt_{v,0}$ and $gt_{v,5}$. However, since the prediction of the counterfactuals should be higher than that of the factual sample, the observations with higher predictions cannot have counterfactuals with lower values. Therefore, the value difference is as close to the valid frontier as possible. Regarding the counterfactual distance to the factual sample, there does not seem to be a relation between the actual value, counterfactual value, and their distance.

Next, using the previous dissimilarities, we compare the counterfactuals extracted with and without manifold closeness restrictions. For clarity, the counterfactuals extracted without manifold closeness restrictions will be referred to as base counterfactuals. Figure 5.3 shows the comparison between the distances of the extracted counterfactuals for each dissimilarity. It can be seen that all observations lie above the diagonal, which is expected because base counterfactuals are always closer to the factual sample. The 52.59% of

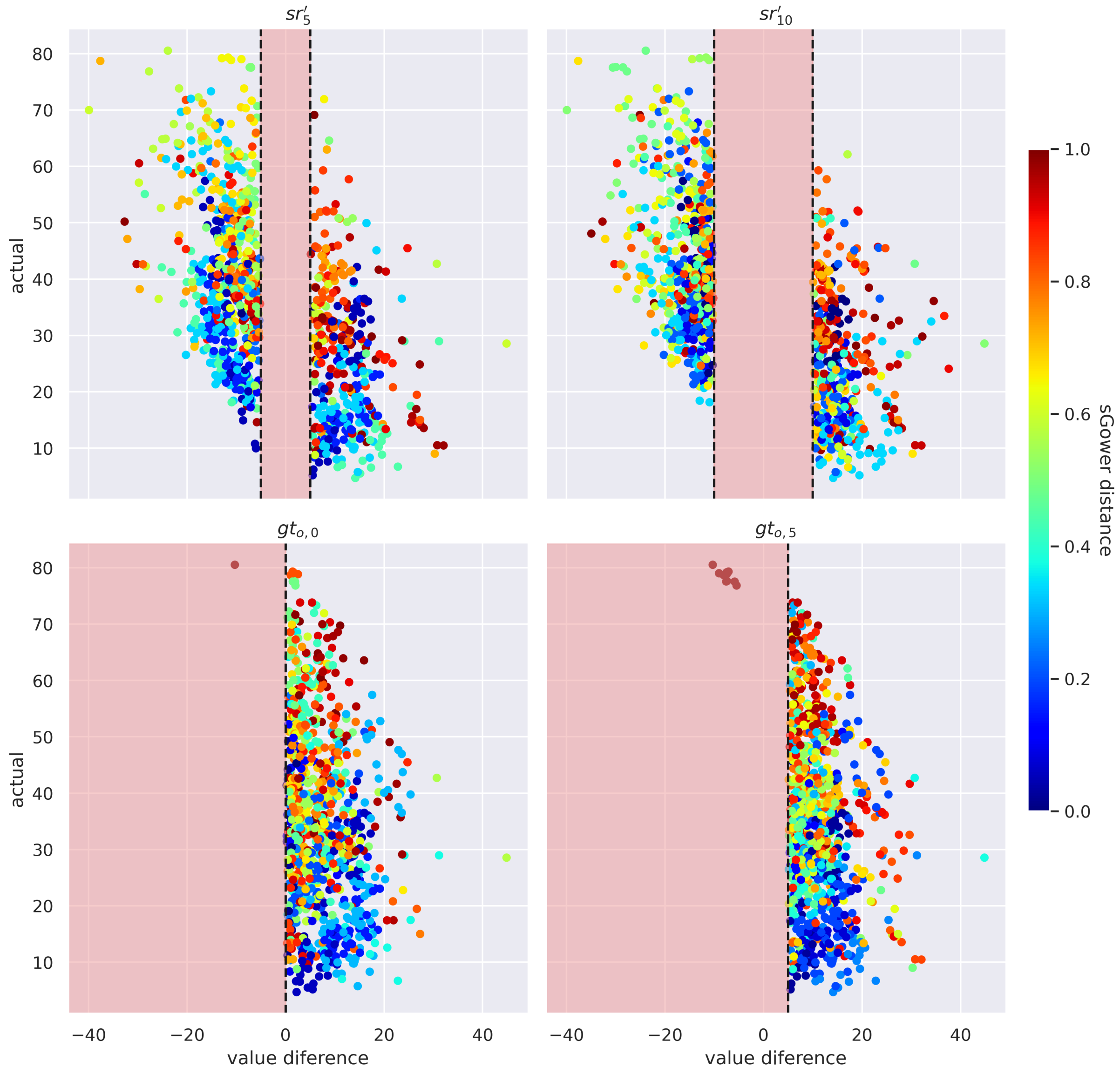

Figure 5.2: *Counterfactuals extracted with several dissimilarities without manifold restrictions. The value difference is between the factual sample prediction and the counterfactual prediction. Actual is the factual sample prediction. The sweet pink regions denote the regions of the space where the grouping measure is met. For visualization purposes, the distance between the factual sample and the counterfactual is transformed using a quantile-based transformation to make it uniform in the interval* $[0, 1]$.

the base counterfactuals meet the manifold closeness restrictions and mostly lie in the diagonal.

On the other hand, when the base counterfactuals do not meet the manifold restrictions, the distance for the counterfactual extracted with manifold restrictions increases. This arrangement implies that the extraction method using manifold restrictions can find counterfactuals comparable to the base counterfactuals when possible. The main finding in this comparison is that the difference in distance between the base and restricted coun-

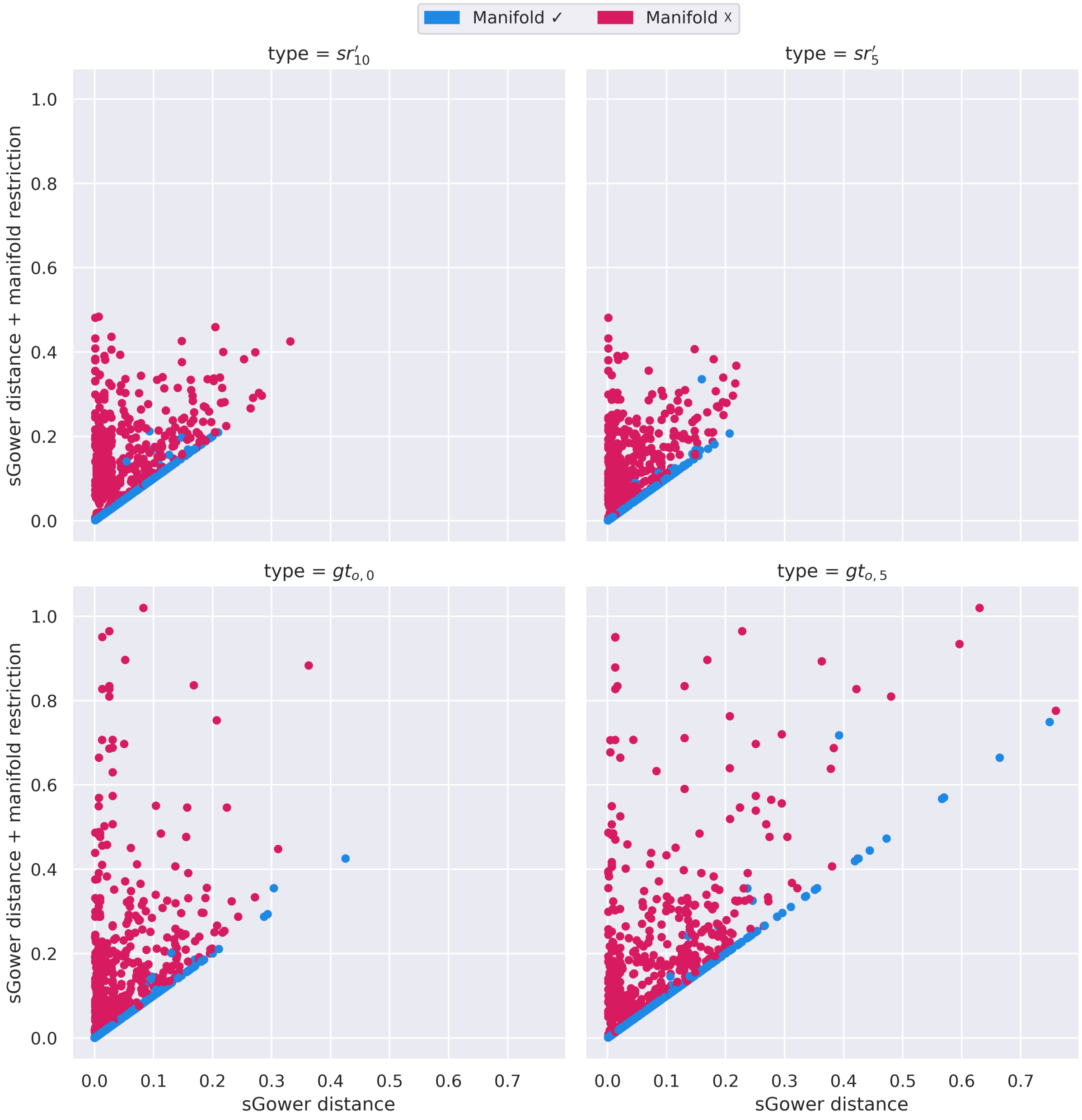

Figure 5.3: *Comparison of the distance to the factual sample of the counterfactuals extracted with and without manifold closeness restrictions for each dissimilarity. For a given point, the x-axis and y-axis represent the sGower distance from the counterfactual to the factual sample extracted with and without manifold restrictions, respectively. The color indicates if the counterfactual extracted without restrictions meets the manifold restrictions (blue) or not (ruby).*

terfactuals is minimal in most cases. Consequently, a realistic counterfactual can usually be selected with a small distance penalization.

There is a notable difference between the distances in a small set of cases, especially using the dissimilarities $gt_{v,0}$ and $gt_{v,5}$. However, this is not a limitation of the extraction method, and it might provide valuable information for understanding the *ML* method with the help of domain experts. For instance, it could imply that there are mixtures not represented in the dataset that yield a better compressive strength. Conversely, it could also indicate that the model is not well-defined in those regions. Nonetheless, both scenarios require domain experts to assess their validity and draw conclusions.

Finally, we show an explanation based on semifactual and counterfactual Explanation Sets for a random observation from the dataset. The observation has the following values: cement=332.50, blast furnace slag = 142.50, fly ash = 0.00, water = 228.00, superplasticizer = 0.00, coarse aggregate = 932.00, fine aggregate 594.00, and age 270.00. The prediction for this observation is 41.87. The grouping measure is $gt_{v,0}$. A new binary target is calculated using the grouping measure over all the samples from the training set to create this explanation. Then, a DT classifier with a maximum depth of 3 was trained on this data. Figure 5.4 shows the resulting DT. We can obtain 8 Explanation Sets from this tree, one for each leaf. Thus, the representation is a combination of several rule-based counterfactual and semifactual Explanation Sets. The choice of the maximum depth is a trade-off between Explanation Set specificity and fidelity. The number of leaves is positively correlated with the depth of the Decision Tree. Thus, a large depth produces a large number of Explanation Sets that are too specific (low coverage) and have a high fidelity. In contrast, a low depth produces broader Explanation Sets with lower fidelity but higher fidelity. The choice of a maximum depth completely depends on the requirements of the domain, and in this example, it was chosen low for visualization purposes.

In the resulting DT, the leftmost leaf represents a pseudo counterfactual Explanation Set because 41.87 is not greater than itself, and it is defined as *age* ≤ 42.00, *cement* ≤ 266.10, *water* ≤ 156.20. This pseudo counterfactual Explanation Set has a coverage of 2.46% and a fidelity of 76.47% (lower than the required 90.00%). The second leaf, also classified as > 41.87 has a fidelity of 98.77% and, consequently, represents a valid counterfactual Explanation Set. These two leaves can be joined by removing their last condition, resulting in a valid counterfactual Explanation Set with a bigger coverage. On the other hand, the last orange leaf represents a semifactual Explanation Set defined as *age* > 42.00, *superplasticizer* > 6.30, and coarse agg. ≤ 1087.54. Its fidelity is 97.22%, and the coverage is 6.99%.

5.1.2. Classification case study

The proposed framework is evaluated in a classification task in the second case study. For this purpose, the Adult dataset (Dua & Graff, 2017) has been selected. The goal of the dataset is to estimate if a person earns less or more than \$50*k*. The data is based on census data and dates to 1994. The dataset contains 29170 instances and 12 features. The features are numerical: age, capital gain, capital loss, and hours per week, and categorical: work class, education, marital status, occupation, relationship, race, sex, and country. The dataset is slightly imbalanced. Approximately 80% of the observations

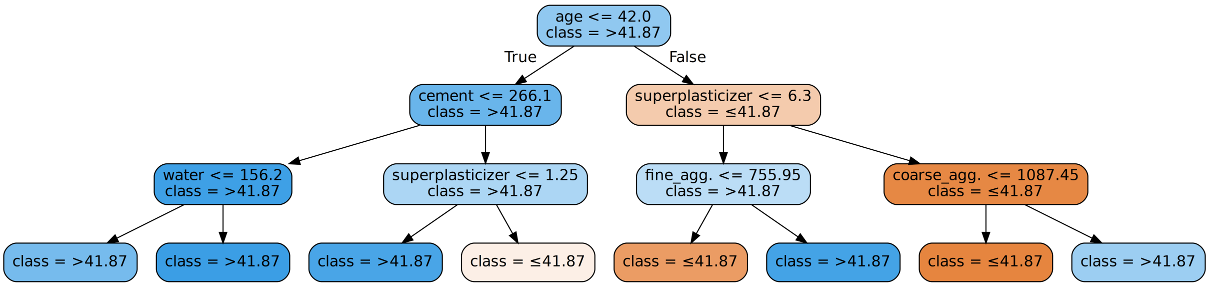

Figure 5.4: *Semifactual and counterfactual Explanation Sets extracted for a random observation from the dataset. The prediction for this observation is* 41.87*. The conditions from the root to a leaf denote an Explanation Set. The leaves with value* ≤ 41.87 *are semifactual Explanation Sets, and the leaves with value* > 41.87 *are counterfactual Explanation Sets. Only the leaves 2, 3, 5, and 6 are valid Explanation Sets (meet the fidelity requirement).*

belong to the $\leq \$50k$ group, and the remaining 20% belong to the $> \$50k$ group. The dataset is preprocessed similarly to *Model-Agnostic Counterfactual Explanation* (*MACE*) experiments (Karimi *et al.*, 2020).

The focus of this case study is the effects of using different feasible sets. Specifically, we compare explanations obtained without restrictions to those extracted with various actionability restrictions. Additionally, we examine through an example the effect on the distance of implementing a diversity penalization. The grouping measure employed is the identity similarity for semifactuals and its complementary (see Eq. 3.4) for counterfactuals. The representations (see taxonomy in Section 3.4) are rule-based (restrictive and approximate) and finite enumeration (exact and non-restrictive).

First, we evaluate the pattern of the changes in semifactual and counterfactual Explanation Sets and the individual counterfactuals used to generate the counterfactual Explanation Sets. These explanations are extracted using two feasible sets:

- **Base feasible set**: The restriction function is the sGower distance (Fernández *et al.*, 2019) parametrized with the Manhattan distance for numeric variables and simple-matching for categorical variables.

- **Restricted feasible set**: Besides the base sGower distance, it includes actionability restrictions over categorical features (see Eq. 3.12). These restrictions are combined using the sum method (see Section 3.3). The actionability restrictions are multiplied by the restriction value. The features not considered actionable are marital status, race, relationship status, and sex. In individual counterfactuals, the age is also fixed.

The restriction value is initially set to a high value to ensure that no observation is filtered out based on the sGower distance (e.g., the number of features). Then, on the individual counterfactual extraction, we minimize the restriction function directly, which, as previously mentioned, is equivalent to iteratively reducing the restriction value until no counterfactual is found.

As an illustrative example, consider the following observation: age=37, workclass=private, education=masters, marital status=married, occupation=white collar, relationship=wife, race=white, sex=female, capital gain=0.00, capital loss=0.00, hours per week=40.00, and country=United States. The explanations for this observation using the base feasible set are:

- Individual counterfactual (finite enumeration): the modified values are age=28 and education=associates.
- Semifactual Explanation Set (rule-based): education=masters, occupation=white collar, marital status=married.
- Counterfactual Explanation Set (rule-based): age $\leq$ 28.00 and hours per week $\leq$ 40.00.

Figure 5.5 shows a summary of the most common restriction (semifactual) and change (counterfactuals) patterns. Thus, we can visually determine the most common features considered in the explanations. As an example, we explain the most common patterns for counterfactual Explanation Sets (Figure 5.5 A) and semifactual Explanation Sets (Figure 5.5 B). The most common pattern in the counterfactual Explanation Sets appears in the 18% of the explanations of the $>$ \$50k class. These counterfactual Explanation Sets only have changes in the age and hours per week features. On the other hand, the most common pattern in the semifactual Explanation Sets explanations indicates that in the 40% of cases, only the age and marital status have restrictions.

In the individual counterfactual explanations, there is little difference between the number of feature changes for the two classes. Unlike Anchor explanations that are rule-based, individual counterfactuals are represented by an observation. Therefore, counterfactuals are not affected by too-specific regions (i.e., small regions of the feature space belonging to a given class) that would result in low coverage Anchor explanations. Most counterfactuals involve changes over age, which might suggest that this feature is very relevant in the classification. The capital gain and hours per week are also relevant in the base explanations, and their presence gets magnified in the restricted explanations.

Besides age, the features: race, sex, relationship, and marital status are also fixed in the restricted explanations. In contrast with the feature age, these features only involve changes in a few patterns that are not common. This does not imply that these features are less relevant in the classification, and different restrictions might generate other patterns. While changes over these features might not be helpful if the goal is to try to change the outcome, they are useful in other tasks like detecting possible biases and assessing their importance.

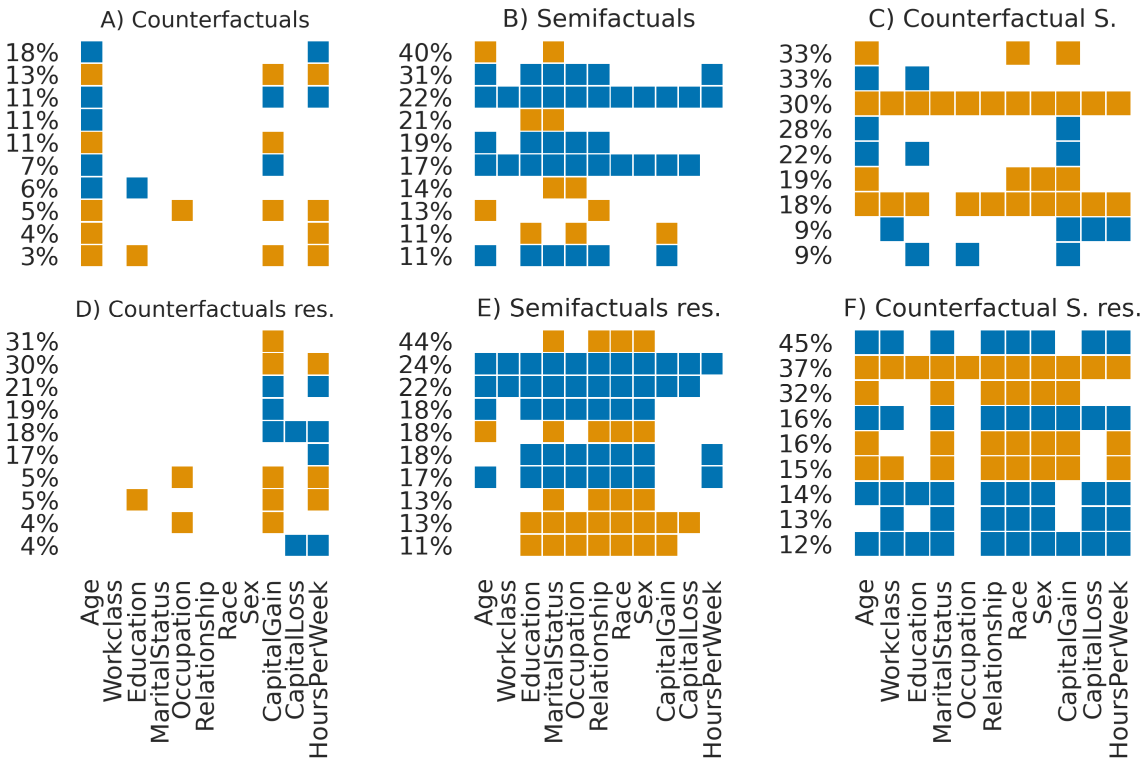

Figure 5.5: *Counterfactual and semifactual Explanation Sets, and individual counterfactual change patterns. The patterns (rows) indicate the percentage of explanations sharing the same structure. The presence of blue and orange squares indicates a change in that feature in counterfactual-based explanations and a restriction over that value in semifactual-based explanations. Orange rows indicate that the pattern belongs to the* $\leq \$50k$ *class, and the blue color is related to the* $> \$50k$ *class.*

In the semifactual Explanation Sets (see Figure 5.5 B) and E)), there is a high difference in the number of conditions between the patterns of the group $\leq \$50k$ and the $> \$50k$ group. The reason behind this difference is that the classes are slightly imbalanced. This imbalance primarily affects the number of conditions and the coverage, as it promotes more specific explanations that result in more conditions and lower coverage. In addition, the actionability restrictions also decrease the average fidelity and coverage and increase the number of conditions. Unlike the semifactual Explanation Sets, the number of conditions is lower in the $> \$50k$ group in counterfactual Explanation Sets. However, this does not contradict the previous findings since the counterfactuals for the $> \$50k$ group belong to the $\leq \$50k$ class and vice versa. Therefore, it supports the previous findings that Anchor obtains better explanations for the majority class in imbalanced problems.

Table 5.1 compares the quality metrics for both counterfactual and semifactual Explanation Sets. In the semifactual-based explanations, the average number of conditions increases by 35.36% when adding the restrictions, and the coverage increases by 233.14%. The coverage is calculated over the training feasible set, and consequently, it will increase if the restrictions result in a more homogeneous training feasible set. However, the standard deviation of the coverage also increased by 299.08%, which suggests that there are several semifactual Explanation Sets with extreme coverage values (either very high or 0% coverage). As previously seen in Figure 5.5, there is a high difference in quality between the two classes. The fidelity remains similar, and valid semifactual Explanation Sets were produced in most cases (i.e., $fidelity > 0.90$).

Regarding counterfactual Explanation Sets, the number of conditions increased by 27.87% when adding the restrictions, and the fidelity decreased by 23.20%. On the other hand, the coverage increased by 1.88%. The only setting where most of the counterfactual Explanation Sets were valid was for the class > $50$k$ without restrictions. In the others, less than half of the cases were valid explanations. This reveals a big difference in quality between semifactual and counterfactual Explanation Sets. This difference is primarily the effect of two factors:

- Counterfactual extraction method: This method might generate counterfactuals in wiggly regions of the feature space (i.e., underrepresented for the counterfactual class) because it is optimizing the restriction function, not for counterfactual Explanation Sets quality. Consequently, the counterfactual Explanation Sets generated there using Anchor will have low coverage or fidelity.
- Impact of restrictions: In the same way the restrictions help to achieve high coverage in the semifactual Explanation Sets because of the homogeneity, it penalized the counterfactual Explanation Sets.

Explanation type	Label	N. conditions	Coverage (%)	Fidelity (%)
S.F. Base	≤ \$50k	2.19 (0.64)	15.25 (8.70)	98.30 (1.69)
	> \$50k	7.50 (2.34)	0.40 (0.70)	93.33 (10.47)
S.F. Restricted	≤ \$50k	3.89 (1.47)	50.86 (37.73)	98.09 (3.37)
	> \$50k	6.64 (3.14)	1.15 (1.97)	92.85 (13.48)
C.S. Base	≤ \$50k	7.14 (2.36)	1.03 (1.64)	83.00 (12.57)
	> \$50k	3.05 (1.34)	22.33 (13.83)	95.73 (3.89)
C.S. Restricted	≤ \$50k	8.33 (2.33)	1.72 (2.72)	59.55 (32.35)
	> \$50k	6.95 (2.53)	20.2 (22.51)	83.45 (13.68)

Table 5.1: *Explanation set quality metrics calculated for the semifactual and counterfactual Explanation Sets explanations with the base and restricted feasible set. The mean and standard deviation (in parenthesis) are calculated for each explanation type, restriction setting, and label.*

Next, we evaluate how the proposed methodology can enforce diversity over the counterfactuals extracted from the base and restricted feasible sets. The penalization considered is the inverse of the base distance plus 1 to the previous counterfactuals using Eq.

3.13. The average distances were recorded as 0.12, 0.20, and 0.13 for the factual, diverse, and restricted diverse counterfactuals, respectively. The calculated average diversity was 0.42 for base diverse counterfactuals and 0.30 for restricted diverse counterfactuals. From these results, it is evident that the method successfully enforces diversity in both diverse counterfactual settings. While the restricted diverse counterfactuals are, on average, closer to the factual sample than the base diverse ones, they have slightly lower diversity. This phenomenon might be attributed to the higher cost associated with changing the class using non-restricted features, which is supported by their average restriction function, which is approximately 1.81 in both cases.

The representation for diverse counterfactual Explanation Sets is finite enumeration, in particular, a set with two elements: the counterfactual and the diverse counterfactual. Notice that more than one diverse counterfactual can enrich the explanation.

5.2. Counterfactual Explanation Sets in Random Forest

This section focuses on the extraction of counterfactual Explanation Sets in *RF*. Specifically, it compares the proposed extraction methods, *Anchor_ES* and *RF-OCSE*, against several state-of-the-art alternatives. The evaluation is performed in 10 real datasets using counterfactual and counterfactual Explanation Sets metrics. Another goal of this experiment is to assess the performance difference between full (i.e., use model internals) and black-box access. The fidelity is set to $\alpha = 0.90$. Any "counterfactual Explanation Set" with fidelity lower than α is considered a pseudo counterfactual Explanation Set. The *RF* was implemented in *Scikit-Learn* (Pedregosa *et al.*, 2011) with default parameters. The code for *Anchor_ES* and *RF-OCSE* is available online[2].

The chapter is structured as follows. Section 5.2.1 presents the datasets, metrics, and methods used in the experiment. Sections 5.2.2 and 5.2.3 describe the counterfactual and counterfactual Explanation Sets evaluations, respectively.

5.2.1. Experiments setup

Baselines. The performance of *Anchor_ES* and *RF-OCSE* is evaluated using several metrics against state-of-the-art methods whose Python implementation is available online. These methods are listed in Table 5.2. In these methods, only *Anchor_ES*, *Forest-based Tree* (*FBT*), *LOcal Rule-based Explanations* (*LORE*), and *RF-OCSE* produce counterfactual Explanation Sets. Specifically, they generate rule-based explanations. Regarding the validity of the explanations, *MACE*, *Minimum Observable* (*MO*), and *RF-OCSE* can always find counterfactuals (when they exist). In the *FBT* method, the explanations are extracted using the same method as *LORE*. The parameters for *Hot Start* (*HS*) in

[2] `https://github.com/rrunix/libfastcrf`

Method	Valid	Set explanation	Model access	Parameters
RF-OCSE	✓	✓	Full	HS(n_features=3, perturb_size=2)
Anchor_ES		✓	Black-box	
MACE[2] (Karimi *et al.*, 2020)	✓		Full	tolerance 10^{-5}
LORE[3]		✓	Black-box	Genetic neighborhood
Feature-Tweaking (*FT*)[2] (Tolomei *et al.*, 2017)			Full	
FBT[4] (Sagi & Rokach, 2020)	✓	✓	Full	
MO[2]	✓		Black-box	

Table 5.2: *Methods considered in the counterfactuals and counterfactual Explanation Sets evaluation. A ✓in the valid column indicates if the method guarantees that a counterfactual is always found (if it exists). In the set explanation column, a ✓indicates that the method produces counterfactual Explanation Sets. Model access indicates if the method uses the model internals (full) or only makes predictions (black-box). Finally, any relevant parameter is listed in the parameters column.*

Dataset	N. Features	Feature types	N. Instances
abalone	8	real, categorical	4177
adult	11	real, integer, binary, categorical	30718
banknote	4	real	1372
compas	5	integer, binary	5278
credit	14	real, integer, binary	29623
mammographic masses	5	integer	830
occupancy	5	real	2665
pima	8	real, integer	768
postoperative	8	integer, binary, categorical	86
seismic	15	integer, binary, categorical	2584

Table 5.3: *Description of the datasets used in the experiments.*

RF-OCSE have been empirically set on toy examples. Note that *RF-OCSE* with *HS* and *MO* obtains the same explanations, so the distinction between them is only made when it is relevant.

Datasets. Ten tabular datasets, listed in Table 5.3, are considered. These datasets contain integer, categorical, ordinal, and real features. The datasets *adult*, *compas*, and *credit* are preprocessed using the same approach as in *MACE* experiments (Karimi *et al.*, 2020). The categorical and binary variables are represented using numerical encoding, and the continuous variables are standardized.

Metrics. Besides the base metrics specified previously (fidelity, coverage, and distance), we use the extraction time, the percentage of populated counterfactual Explanation Sets (i.e., *coverage* > 0), and the stability (robustness) of the extraction method as defined in (Alvarez-Melis & Jaakkola, 2018) (Eq. 2). The tolerance in the stability is $\epsilon = 0.50$. Lower values in the stability metric (Local Lipschitz estimates) indicate that

[2] `https://github.com/amirhk/mace`

[3] `https://github.com/riccotti/lore`

[3] `https://github.com/sagyome/forest_based_tree`

Dataset	**Method**										
	Anchor_ES		*FBT*		*FT*		*LORE*		*MACE*	*MO*	*RF-OCSE*
abalone	84.76	100%	78.77	9%	90.76	95%	67.55	20%	7.02	83.42	**0.00**
adult	87.60	100%	30.44	36%	64.30	100%	37.50	19%	0.51	75.60	**0.00**
banknote	60.94	100%	23.58	31%	NA	0%	20.65	30%	0.22	61.77	**0.00**
compas	49.79	100%	0.04	100%	54.24	85%	8.66	84%	**0.00**	0.58	**0.00**
credit	93.04	100%	90.14	12%	93.68	100%	60.79	21%	21.87	89.29	**0.00**
mammographic_mases	79.07	100%	8.12	82%	59.52	100%	8.05	88%	0.04	43.12	**0.00**
occupancy	65.78	100%	28.74	42%	71.22	30%	31.41	34%	0.66	65.29	**0.00**
pima	91.67	100%	66.41	15%	81.13	100%	47.17	28%	1.74	86.01	**0.00**
postoperative	48.69	100%	5.18	67%	39.58	73%	0.76	70%	**0.00**	40.20	**0.00**
seismic	86.73	100%	91.64	6%	94.31	30%	65.92	5%	3.68	77.56	**0.00**

Table 5.4: *Relative counterfactual improvement of* RF-OCSE *over the alternatives. The percentage of valid counterfactuals is in parentheses in those methods that do not always generate valid counterfactuals by design. NA implies that the relative counterfactual improvement could not be calculated because there were no valid counterfactuals. The best relative counterfactual improvement for each dataset is in bold.*

the method is more robust. The distance is reported in terms of improvement of *RF-OCSE* to the other approaches. Given the counterfactual generated by *RF-OCSE*, $\mathbf{c}$, the relative counterfactual improvement over other approaches is defined as follows (Karimi *et al.*, 2020):

$$rci(\mathbf{c}, \mathbf{z}, \hat{\mathbf{x}}) = 100 \cdot (1 - d(\mathbf{c}, \hat{\mathbf{x}})/d(\mathbf{z}, \hat{\mathbf{x}})) \tag{5.1}$$

where $\mathbf{z}$ is the counterfactual from the other approach, $\hat{\mathbf{x}}$ is the factual sample, and d is the Gower distance. The metrics are only calculated over the valid counterfactuals.

5.2.2. Counterfactual evaluation

The average relative counterfactual improvement of *RF-OCSE* over the alternatives for each method is reported in Table 5.4. *RF-OCSE* obtains slightly better results than *MACE*. This is attributed to the choice of tolerance in *MACE* since it can approximate the optimal counterfactual within an arbitrary tolerance. This tolerance parameter helps reduce extraction time by requiring fewer iterations. On the other hand, *RF-OCSE* can not make such a trade-off because it always extracts the optimal counterfactual.

RF-OCSE counterfactual distance to the factual sample is always equal or better than *MO*, as the counterfactual obtained with *MO* is used as an approximation for the initial counterfactual in *RF-OCSE*. The high difference in counterfactual distance obtained by *MO* in the *compas* dataset with respect to the other datasets is because the number of possible individuals is considerably less than in the other datasets. The *compas* dataset consists of three binary variables (race, gender, and charge degree), an ordinal variable with three levels based on the individual age, and the prior count, which is an integer variable ranging from 0 to 37. However, roughly 90% of the observations are in the range $0 - 10$. In this 90% of the dataset, there are 264 possible observations that translate to

231 real observations in the dataset.

Regarding the *FT* approach, the average improvement obtained by *RF-OCSE* is 64.87%, producing valid counterfactuals on 83.91% of the cases. *FT* method did not produce a valid counterfactual in the banknote dataset because the counterfactuals could not be derived from a single rule of the *DT*s in the *RF*. The average improvement obtained over the *LORE* approach by *RF-OCSE* is 34.84 %, and its high variance might imply that *LORE* highly depends on the data distribution. Further, the method only produced valid counterfactuals in 31.11% of the cases. Regarding *Anchor_ES*, it produced a valid counterfactual in all cases, and *RF-OCSE* obtained an average improvement of 74.80% over it. Finally, the average improvement of *RF-OCSE* over *FBT* is 40.30%, and its variance is the highest, while the percentage of valid counterfactuals is 34.45%. This result indicates that having full access to the *ML* model can significantly reduce the counterfactual distance and even provide optimality and validity guarantees.

The average counterfactual extraction times in seconds for each dataset and method are reported in Table 5.4. Extraction times are an essential aspect of counterfactual extraction methods because large extraction times are not acceptable in some applications that are meant to be interactive. The time measures could be divided into three groups based on their behavior. Methods in the first group have consistent extraction times regardless of the dataset or the size of the *RF*. This is the case of *LORE* that took, on average, from 15 to 17 seconds to extract an explanation in the experiments. In the second group, the time is influenced by the size of the dataset, as seen in *MO*, which took less than a second in most cases. The third group includes *Anchor_ES*, *FBT*, *FT*, *MACE*, and *RF-OCSE* (with both *HS* and *MO* as initialization strategies) and their time depends on the size (complexity) of the *RF*.

RF-OCSE with *HS* obtained the best extraction time in all cases but one. In that case, *FBT* obtained a better result, but only 36% of the extracted counterfactuals were valid. The second best result was obtained by *RF-OCSE* with *MO*. The difference between them was significant except in three cases that obtained the same extraction time. Thus, we can confirm that the maximum distance parameter in the *RF-OCSE* method plays a considerable role in the extraction time. Further, the time taken in the initial sampling in *HS* is negligible compared with the time reduction in the extraction. Finally, similarly to the conclusions drawn with the counterfactual distance, having full access to the *ML* model gives a great advantage in extraction times.

Table 5.6 displays the average stability of the extraction methods. *FBT* emerges as the most stable method, possibly due to the simplicity of the decision surface of the approximated *DT*, which is supported by the high percentage of invalid counterfactuals (as shown in Table 5.4). *RF-OCSE* and *MACE* rank second in stability, backing the findings made in (Blanchart, 2021) that optimal counterfactual extractors possess high stability. Conversely, *Anchor_ES* and *LORE* yielded significantly poorer results, likely

Dataset	**Method**							
	Anchor_ES	*FBT*	*FT*	*LORE*	*MACE*	*MO*	*RF-OCSE(MO)*	*RF-OCSE(HS)*
abalone	29.76	0.90	0.90	16.56	83.09	0.68	0.17	**0.12**
adult	161.13	**1.20**	7.16	16.64	23.95	1.33	1.96	1.96
banknote	5.23	0.66	0.01	15.54	1.83	0.20	0.02	**0.01**
compas	89.50	0.19	1.08	16.03	11.27	0.79	0.05	**0.00**
credit	161.63	0.92	9.30	17.29	41.86	1.31	0.49	**0.12**
mammographic_mases	1.07	1.19	0.66	15.73	7.76	0.14	0.02	**0.00**
occupancy	8.97	0.97	0.02	15.55	2.26	0.38	0.02	**0.01**
pima	19.16	1.19	0.75	16.35	19.80	0.13	**0.05**	**0.05**
postoperative	3.46	0.30	0.05	16.34	6.44	0.01	0.01	**0.00**
seismic	129.41	0.27	0.08	16.90	13.68	0.07	**0.26**	**0.26**

Table 5.5: *Average extraction time for each dataset and method in seconds. The best extraction time for each dataset is in bold.*

Method's stability						
Anchor_ES	*FBT*	*FT*	*LORE*	*MACE*	*MO*	*RF-OCSE*
11.15 (8.59)	**1.17** (0.16)	4.75 (3.48)	19.88 (41.90)	1.31 (0.25)	1.73 (0.29)	1.29 (0.25)

Table 5.6: *Average stability of the extraction method. The standard deviation is in parenthesis right to the mean. Lower stability metric values (Local Lipschitz estimates) are desirable. The best stability is in bold.*

because of the high variability in their sampling procedure, which affected the subsequent counterfactual estimation.

A counterfactual example extracted by each method for a sample in the *adult* dataset is shown in Table 5.7. The goal of the *adult* dataset is to predict if the income of an individual is less or equal than \$50k using demographic information such as age, work hours per week, capital gain, and marital status. In this example, the individual earns less than \$50k, and the counterfactuals suggest possible changes to increase the income to more than \$50k. The generated counterfactuals sometimes contain impractical changes, such as age changes. However, these changes are impractical because of the domain information. For example, a system to evaluate if an individual can get a driver license might suggest waiting until an individual has the minimum driving age.

5.2.3. Counterfactual Explanation Sets evaluation

In the counterfactual Explanation Sets experiments, reported in Table 5.8, the coverage, the percentage of counterfactual Explanation Sets populated, the fidelity, and the percentage of explanations that meet the fidelity requirement are listed for each dataset and method.

The counterfactual Explanation Sets extracted by *LORE* provide the broadest coverage. However, this wide coverage comes at the expense of having, on average, low fidelity. *LORE* only managed to produce valid explanations on 17.60% of the cases. Thus, while providing good coverage, these explanations do not correctly identify rules on the dataset

Feature	Factual sample	Methods						
		ANCHOR-ES	*FBT*	*FT*	*LORE*	*MACE*	*MO*	*RF-OCSE*
Age	25.00	39.0					34.00	
CapitalGain	0.49		0.56	1.42	0.86	0.96	-0.15	0.96
CapitalLoss	-0.22	-0.15						
EducationLevel	Doctorate							
EducationNumber	14.00							
HoursPerWeek	37.00	60.0					40	
MaritalStatus	Never-Married							
NativeCountry	United-States							
Occupation	Prof-specialty							
Relationship	Not-in-family							
WorkClass	Private							

Table 5.7: *Counterfactual example extracted from the adult dataset for each method. Only the changes over the factual sample are shown. In this example, the counterfactuals extracted by* FBT *and* LORE *belong to the factual class and are invalid.*

Dataset	**Method**							
	Anchor_ES		*FBT*		*LORE*		*RF-OCSE*	
	fidelity	*coverage*	*fidelity*	*coverage*	*fidelity*	*coverage*	*fidelity*	*coverage*
abalone	52.05 (7.85%)	**16.43** (99.95%)	66.72 (12.26%)	6.2 (91.55%)	52.30 (5.46%)	10.84 (87.02%)	**100.00**	0.00 (0.79%)
adult	62.90 (6.42%)	7.54 (70.80%)	70.97 (28.86%)	2.26 (71.87%)	30.35 (6.82%)	**11.42** (94.02%)	**100.00**	0.02 (17.10%)
banknote	96.60 (90.31%)	10.02 (100.00%)	72.03 (13.70%)	10.36 (27.77%)	40.87 (20.19%)	**26.19** (97.67%)	**100.00**	0.03 (6.41%)
compas	89.36 (59.40%)	23.28 (100.00%)	100.00 (98.69%)	0.85 (98.69%)	71.37 (17.20%)	**24.14** (94.51%)	**100.00**	0.86 (98.94%)
credit	54.64 (17.62%)	4.16 (97.95%)	61.44 (10.52%)	1.21 (56.29%)	59.01 (3.48%)	**8.03** (78.50%)	**100.00**	0.00 (0.81%)
mammographic_mases	81.24 (31.45%)	**36.15** (100.00%)	88.34 (38.19%)	0.64 (50.24%)	78.36 (45.54%)	26.66 (93.86%)	**100.00**	0.18 (60.84%)
occupancy	87.29 (61.46%)	29.63 (99.55%)	80.42 (17.15%)	3.30 (32.50%)	40.25 (32.91%)	**40.01** (97.26%)	**100.00**	0.44 (9.42%)
pima	91.14 (67.58%)	5.14 (99.87%)	62.17 (13.67%)	3.74 (60.42%)	51.28 (6.77%)	**12.07** (93.75%)	**100.00**	0.00 (0.13%)
postoperative	78.87 (56.98%)	6.10 (100.00%)	100.00 (15.12%)	0.24 (16.28%)	63.46 (15.12%)	**10.29** (73.26%)	**100.00**	0.41 (26.74%)
seismic	42.76 (19.58%)	**7.15** (99.88%)	87.58 (18.15%)	0.25 (37.50%)	76.60 (22.69%)	3.52 (68.25%)	**100.00**	0.00 (2.67%)

Table 5.8: *Evaluation of counterfactual Explanation Sets extracted by* Anchor_ES, FBT, LORE, *and* RF-OCSE. *The results of fidelity and coverage are the average over the test samples. The percentage of explanations meeting the fidelity restriction is next to the average fidelity. The populated percentage is in parentheses, right to the coverage. The best result for each dataset and metric is in bold.*

that generate valid counterfactuals. A broader coverage should only be preferred when it contains mostly valid counterfactuals. Otherwise, they do not add relevant information.

A broad coverage is only possible when a set of feature conditions applies to a significant portion of the individuals in the counterfactual class. However, this is not always possible in complex datasets and *ML* models where the effect of the features depends on the region of the feature space.

Regarding *FBT*, its coverage is lower than that of *LORE*, but it provides a better fidelity. Nevertheless, only 26.31% of the cases met the fidelity requirement. The high fidelity contrasts with the high number of invalid counterfactuals in Table 5.4. This could be explained by the method producing a good approximation of the *RF* that is not well-defined near the surface decision, which is where the counterfactuals are.

Anchor_ES obtained a coverage similar to *LORE*, but it produced significantly more valid explanations. Specifically, the explanations satisfy the fidelity requirement in 41.86%

of the cases. In addition, almost all counterfactual Explanation Sets are populated. Consequently, it is a good alternative when full access to the *ML* model is not possible.

On the other hand, for *RF-OCSE*, the counterfactual Explanation Sets have low coverage, but they have the maximum fidelity, as they do not have non-counterfactual by design. Despite having a low coverage, the counterfactuals extracted from the counterfactual set have a high similarity with the samples in the dataset that satisfy the counterfactual set using the proximity measure proposed in (Breiman, 2002). Thus, the proposed changes are realistic (close to the data manifold). This is a consequence of constructing the counterfactual set from the remaining rules in the conversion, as most counterfactuals from the counterfactual set fall in the same leaves in the *RF* (they might differ in some leaves, but the class is already determined). Besides, the counterfactual set is guaranteed to contain the optimal counterfactual by design.

As a result of providing very specific counterfactual Explanation Sets, they are satisfied by at least one sample of the dataset in 21.14% of the cases. Counterfactual Explanation Sets that do not contain samples from the training set provide an additional possibility for explainability. This property allows us to detect and explain relevant regions not represented in the training set. In the case of highly populated datasets, these situations should be rare. This is the case of the *compas* dataset, where most feature combinations are covered.

The significant difference in coverage and percentage of counterfactual Explanation Sets populated between *RF-OCSE* and the extracted from *LORE* is because it uses local rules (they are built using local information), whereas *RF-OCSE* use global rules. *Anchor_ES* and *FBT* also use global rules, but we skip them from the comparison because of the low fidelity. Local rules only take into account splits relevant to the neighborhood. In contrast, global rules, in addition to those splits relevant in the neighborhood, also assert all the splits that were relevant for the classification. Thus, global rules provide a context for the changes that allow better generalization and comparison with other individuals. However, this context does not imply that a feature outside the bounds of the rule will result in an invalid counterfactual. It only provides the context in which the prediction was made.

An example of a counterfactual Explanation Set extracted by *Anchor_ES*, *FBT*, *LORE*, *RF-OCSE* is shown in Table 5.9. In the *Anchor_ES*, *FBT*, and *LORE* methods, the explanations are pseudo counterfactual Explanation Sets because they did not meet the fidelity requirement. *RF-OCSE* explanation has restrictions in most features, but this does not make it difficult to understand the explanation as most conditions do not imply changes. Besides, as previously mentioned, the observations within *RF-OCSE* counterfactual Explanation Sets are similar using the proximity measure proposed in (Breiman, 2002). In contrast, *Anchor_ES* has many changes but with higher coverage. However, it does not satisfy the fidelity requirement.

Feature	**Method**			
	Anchor_ES	*FBT*	*LORE*	*RF-OCSE*
Age	**37 < age ≤ 48**	≤ 29.50		≤ 26.00
CapitalGain		**0.55 < x ≤ 0.82**	**> 0.86**	**> 0.96**
CapitalLoss	≤ −0.22	≤ 4.38735		
EducationLevel	Masters, Doctorate, Assoc., ...	Doctorate, Prof-school, Assoc., ...		Masters, Doctorate, Assoc., ...
EducationNumber	**≤ 13**	> 10.50		> 13.50
HoursPerWeek	**> 45**	> 27.50		≤ 38.00
MaritalStatus				All except married and divorced
NativeCountry				
Occupation		Other-service, Priv-house-serv, Prof-specialty, ...		
Relationship	**Husband**	Husband, Not-in-family		All except husband
WorkClass	**Gov.**	Gov., Private		Gov., Private, Self-emp-inc

Table 5.9: *Example of counterfactual set extracted by* RF-OCSE *and pseudo counterfactual set from* LORE *and* FBT *for the Adult dataset. The conditions that are not satisfied by the factual sample are in bold. The factual sample is the same as in the example in Table 5.7.*

The coverage in *RF-OCSE* can be improved by simplifying the ruleset that generates the counterfactual Explanation Set. This simplification can be achieved using Algorithm 6. The threshold is set to 0.50, which ensures that the greedy-rule-selection retains the rules necessary to achieve a fidelity of 100.00% (i.e., it can only contain counterfactuals). However, if the probability threshold is less than 0.50, the method produces counterfactual Explanation Sets whose fidelity is lower than 100.00% (i.e., it can contain non-counterfactuals). Also, having a broader coverage increases the percentage of populated counterfactual Explanation Sets.

In Table 5.10, the counterfactual Explanation Sets extracted through the simplification of the rule selection are evaluated. The rule selection is made with different probability thresholds in Algorithm 6. Additionally, a dynamic approach that decreases the probability threshold until the coverage exceeds 0 is considered.

The rule selection relaxation vastly increases in the best case (dynamic approach)

Dataset	**Method**					
	*RF-OCSE*_dynamic		*RF-OCSE*_0.4		*RF-OCSE*_0.2	
	fidelity (%)	coverage	fidelity (%)	coverage	fidelity (%)	coverage
abalone	88.42	**0.63** (100.00%)	**99.54**	0.10 (14.92%)	93.98	0.28 (62.75%)
adult	96.83	**0.86** (100.00%)	**99.05**	0.11 (58.19%)	97.87	0.77 (84.42%)
banknote	99.55	7.16 (100.00%)	**99.99**	1.81 (81.71%)	99.71	**7.67** (98.76%)
compas	99.13	**0.87** (100.00%)	**99.85**	0.86 (99.62%)	99.50	**0.87** (99.85%)
credit	86.98	**0.11** (100.00%)	**96.89**	0.00 (9.44%)	95.20	0.02 (46.44%)
m. mases	94.67	**0.94** (100.00%)	**99.89**	0.26 (74.58%)	95.42	0.67 (89.88%)
occupancy	98.36	**7.25** (100.00%)	**99.61**	2.09 (67.69%)	98.63	6.73 (92.35%)
pima	88.10	**1.32** (100.00%)	**98.57**	0.02 (9.11%)	93.79	0.27 (56.90%)
post-op.	87.56	**2.45** (100.00%)	**100.00**	0.69 (44.19%)	87.52	1.82 (86.05%)
seismic	77.77	**0.12** (100.00%)	**96.10**	0.00 (10.91%)	89.42	0.02 (39.51%)

Table 5.10: *Evaluation of the counterfactual Explanation Sets extracted by the rule selection simplification in* RF-OCSE*. The number after* RF-OCSE *indicates the probability threshold used in Algorithm 6, or if the approach is dynamic. The results of fidelity and coverage are the average over the test samples. The percentage of populated Counterfactual Explanation Sets is in parentheses, right to the coverage. The best result for each dataset and metric is in bold.*

the coverage from 0.19 to 2.71 while having a high fidelity. Also, the percentage of populated counterfactual Explanation Sets is 100.00%. This increment in coverage is notable from the threshold of 0.40, which suggests that many counterfactual Explanation Sets have a very restrictive condition that significantly impacts their classification as counterfactuals. The rule selection relaxation behaves similarly in all datasets except for the seismic dataset. In this dataset, the fidelity in the dynamic approach is 77.77%, which contrasts with the much higher fidelity values using the probability thresholds 0.20 and 0.40. This can be attributed to some counterfactual Explanation Sets not being realistic, as the changes are not met by any sample in the dataset. However, this is not a problem of the extraction method, but the decision surface of the *ML* model itself, as counterfactuals by definition, live in the vicinity of the decision surface where the class flip happens. This fact can be used to diagnose *ML* models by inspecting the decision surface through counterfactual Explanation Sets.

Chapter 6

Conclusions

This thesis has focused on explanation techniques for *Machine Learning* (*ML*) models. Explanation techniques aim to make the *ML* models and/or their predictions interpretable by humans. Given the increasing integration of Artificial Intelligence into our society, these techniques are more crucial than ever. They target a wide audience, from *ML* experts and domain users developing these tools to non-technical individuals who may not even realize they are using them. Recognizing the diverse backgrounds and objectives of these users, we aim to empower them with tools to tailor the explanations to their preferences and goals. With this in mind, we set at the beginning of this thesis the following objectives:

O1) To provide a new explanation methodology unifying counterfactuals and semifactuals based on similarity measures, emphasizing their complementarity and a standard methodology to define the feasible sets.

O2) To provide a taxonomy of current set-based representations in the literature for counterfactuals and semifactuals.

O3) To develop an agnostic method to extract these new explanations based on *Anchor*, a well-known agnostic explanation method.

O4) To develop a method to extract these new explanations from a *Random Forest* (*RF*) leveraging on its internal structure and axis-parallel decision surface.

O5) To validate the proposed explanation methodology and compare the extraction methods to alternatives in the literature.

The objectives O1 and O2 have been achieved through the Explanation Sets framework, which encompasses counterfactuals and semifactuals under a single framework. It provides users the tools for expressing restrictions over observations (feasible set) and considering different notions of similarity for the outcomes. Furthermore, the framework extends these explanation techniques to tasks where a similarity in the output can be defined. Explanation Sets also advocate for using sets of observations for explanations instead of a single observation, a concept supported by the literature (Guidotti *et al.*, 2018, 2019; Ribeiro *et al.*, 2018).

The extraction of Explanation Sets has been addressed from two perspectives: *Anchor_ES* (Objective O3), a black-box agnostic method to extract counterfactual and semifactual Explanation Sets, and *Random Forest Optimal Counterfactual Set Extractor* (*RF-OCSE*) (Objective O4), a full-access method to extract counterfactual Explanation Sets from a *RF*. The new explanation methodology was successfully validated in two case uses, and the proposed extraction methods were compared with alternatives in the literature in counterfactual Explanation Set extraction (Objective O5).

From an academic perspective, we have tried to address the following research questions from Section 1.2:

Q1) Can a *RF* be converted into a *Decision Tree* (*DT*)? Is it a valid mechanism to explain a *RF*?

Q2) From an explainability point of view, are sets of observations better than a single observation?

Q3) How do different notions of similarity affect the extracted counterfactual and semifactual explanations?

Q4) Are full-access explanation techniques better than black-box techniques?

RF is among the most used *ML* algorithms. They require little data preprocessing, support categorical features, and perform well even with little to no hyperparameter tuning. In contrast with their building block, *DT*, they are not considered interpretable. The question of whether they are equivalent naturally arises. The conversion of a *RF* into a *DT* is detailed in Chapter 4. However, because of the combinatory nature of the process, it is not useful in real-world cases, less to provide an explanation, ruling out the interpretability of decision trees. This interpretability is tied to complexity, and only simple *DT*s are interpretable out-of-the-box. Nevertheless, this conversion opened opportunities for other explanation methods, such as counterfactuals. Specifically, the partial conversion of a *RF* into a *DT* enabled to extract the closest (optimal) counterfactual. Also, the usage of the rule within the partially converted *DT* that this counterfactual satisfies provides a set of counterfactuals.

Using sets of counterfactuals or semifactuals rather than a single observation is motivated in Chapter 3, providing several representations already being used in the state of the art as an example. The results from Chapter 5 also provide examples of why this set representation is better. For instance, rule-based Explanation Sets with low coverage should be used with caution because they are defined in low-density regions of the model that have not been tested. If we were considering only an observation, we would not have this information.

Counterfactuals and semifactuals based on similarity measures are introduced in Chapter 3, providing examples of their usefulness depending on the goal of the explanation. They were validated in a real example in Chapter 5, which provided useful insights. The usage of restrictive similarity measures leads to an imbalanced problem where the current explanation extraction methods are not well-suited. We showed that the quality of the explanations is much better for the majority class. Also, the restrictiveness of the similarity should take into account the precision of the model.

Methods with full access obtained better results across all metrics and took less time in the extraction process. The conclusion from the experiments in Chapter 5 is that they should be preferred when possible. Further, since only a few types of *ML* models are used in practice, this should not be much of a problem. Full-access methods such as *RF-OCSE* and *MACE* can also guarantee optimality and the former fidelity. While agnostic methods such as *Anchor* and *Anchor_ES* can provide statistical guarantees on fidelity, the computing time was significantly higher in the experiments, and they did not find satisfactory explanations in several cases.

The main contributions of this research, as well as the open questions and improvement opportunities, are listed below. It also presents a list of publications in scientific journals and international conferences that have resulted from this research.

6.1. Main contributions

The main contributions of this thesis focus on unifying and extending counterfactual and semifactual explanation methods and methods to extract them. Two major contributions have been made to these areas, leading to two scientific articles (Fernández *et al.*, 2020) and (Fernández *et al.*, 2022). These contributions have been presented in Chapters 4 and 3, respectively.

The main conclusions of each of the two contributions are summarized below:

- The introduction of the proposed explanation framework, Explanation Sets, and *Anchor_ES*, an agnostic approach to extract Explanation Sets.
- The development of *RF-OCSE*, an approach to extract counterfactual Explanation Sets from *RF* with optimality guarantees.

Explanation Sets

In Fernández *et al.* (2022), a new explanation framework called Explanation Sets that unifies counterfactual and semifactual explanations was presented. Explanations Sets are an example-based Explanaible *ML* technique to explain *ML* predictions. The key idea is simple yet powerful: explain *ML* predictions using observations from a subregion of the feature space (neighborhood) and whose prediction compared with factual prediction satisfies a criterion based on the user preferences (grouping measure). Besides, a method to extract counterfactuals and semifactual Explanation Sets, called *Anchor_ES*, was introduced.

This thesis further elaborates on the Explanation Sets framework, simplifying the definitions and restrictions, but the concept and results remain. Specifically, the restrictions on the feature space originally imposed through a neighborhood (distance) are now specified through a feasible set. This feasible set is defined using a smaller than inequality constraint. The main advantage of this definition is that symmetry is no longer required, which significantly simplifies some restrictions relative to the factual sample (e.g., actionability). Further, fidelity, which was previously a desirable property, is now a requirement in the Explanation Set definition. An Explanation Set is valid only if it satisfies a user-defined fidelity.

The proposed explainability framework was evaluated on two use cases concerning classification and regression tasks using the *Anchor_ES* extraction method. In the regression task, several dissimilarities and similarities were considered. In this case study, we show how converting the regression problem into a binary classification problem using the grouping measure leads to an imbalanced classification problem. This fact is not a limitation of the proposal, but most counterfactual and semifactual extraction methods perform worse in imbalanced problems. Further, a large difference in explanation set quality between the minority and majority classes should be expected because most observations of the feasible set belong to the majority class. This fact is also evidenced in the classification case study, where the classes are slightly imbalanced. In particular, when the restrictions result in a more homogeneous feasible set, this quality difference is magnified.

Another interesting finding is that depending on the *ML* model and task, it might not be possible to always extract a counterfactual explanation. This is because the output of some models, such as *DT* or *RF*, is bounded, and they cannot predict values lower or higher than those in the training set. Another reason is that the feasible set might be highly restrictive. However, in such cases, the absence of counterfactuals is the best explanation. It suggests that there does not exist a hypothetical scenario in which that dissimilarity under the given feasible set is fulfilled.

Finally, we show that extracting counterfactual Explanation Sets by first extracting a counterfactual and then extracting a semifactual set might not be the best approach. This is because the counterfactual extraction method only optimizes for counterfactual quality, not counterfactual set quality. While it is possible to promote counterfactual Explanation Sets quality in the counterfactual extraction method, the approach will be similar to extracting counterfactual Explanation Sets directly. The counterfactual Explanation Set extraction experiments further corroborated this point. In several instances, *Anchor_ES* struggled to provide satisfactory explanations and showed considerable instability. Nevertheless, the method did manage to achieve a compromise between fidelity and coverage.

Random Forest Optimal Counterfactual Set Extractor (RF-OCSE)

In Fernández *et al.* (2020), we presented counterfactual sets and a method to extract them from a *RF* called *RF-OCSE*. The counterfactual sets defined in this work laid the foundation for Explanation Sets, and within this framework, they are counterfactual Explanation Sets with fidelity $\alpha = 1$ using a rule-based representation. *RF-OCSE* is a counterfactual Explanation Set method based on the partial conversion of a *RF* into a *DT*. The partial fusion enables the conversion of only the locality of the counterfactual by filtering those rules whose distance is greater than the closest known counterfactual. The optimal counterfactual set is generated by combining the remaining rules in the conversion when the optimal counterfactual is found. This thesis proposes a new method to estimate the initial counterfactual called *Hot Start* (*HS*), where the closest counterfactual among the previously extracted counterfactuals (other extractions) and the dataset augmented with synthetic observations is considered.

The generated counterfactual sets were evaluated in terms of coverage, the percentage of populated counterfactual sets, and fidelity. *RF-OCSE* is the only method supporting set explanations that always produced valid explanations and used only a fraction of the time taken by the alternatives. This time reduction is more evident when using the *HS* method, significantly reducing the time in most cases. Thus, providing a good lower bound in the counterfactual search plays a huge role in the time taken. The coverage is lower than the alternatives, but the covered observations are similar under the proximity measure, which makes the comparison easier and more meaningful. The ability to find counterfactual sets that do not contain samples from the training set is not a limitation, but it informs us that explanations on these regions should be taken with caution.

The counterfactual Explanation Sets from *RF-OCSE* can be relaxed to provide a broader coverage while keeping a high fidelity. This relaxation is achieved by decreasing the threshold in the rule selection algorithm. These relaxed counterfactual Explanation Sets were evaluated in the same settings as counterfactual sets, achieving a better coverage while keeping a high fidelity.

6.2. Open questions and improvement opportunities

We have proposed a new methodology that extends and improves two of the most common explanation techniques and methods to extract these explanations. Yet, many challenges remain unresolved, both general and specific, in the different contributions. The issues outlined below represent an opportunity not only for innovation in the academic field but also to facilitate a seamless and safe integration of Artificial Intelligence within our lives. A key point in this integration is the development of good-quality explainability libraries with in-depth documentation and examples, which is often neglected in research. However, it is what ultimately leads to the adoption of these technologies. The possible improvements and extensions for each contribution are described below.

Explanation Sets

Future work will focus on developing new methods to extract Explanation Sets from *ML* models. New extraction methods will focus on overcoming the limitations of the current approaches in imbalanced problems that might occur due to a restrictive grouping measure. Similarly to *MACE*, full-access methods that leverage the model internals will be explored for the most common *ML* methods. End-to-end extraction approaches for counterfactual Explanation Sets will likely improve the quality of the explanations because they could be optimized throughout the process.

Another future area of work will be to study how different feasible sets affect the resulting explanations. Specifically, a comparison between user-defined restrictions and model-induced restrictions, as well as their possible use cases. In this regard, the effect of high dimensionality and the uncertainty of the predictions in the explanation quality will also be studied. Additionally, we will study new metrics to better estimate the quality of Explanation Sets. In particular, a coverage measure that considers the observation to be explained and weighs down distant observations, possibly also considering data manifold closeness, will be considered.

Random Forest Optimal Counterfactual Set Extractor (RF-OCSE)

Future work will focus mainly on including new definitions of similarity and feasible sets and enhancing performance. The applicability of the extraction method could be improved by incorporating new similarity measures, enabling the extraction of both semi-factuals and counterfactuals. To achieve this, new definitions of the worst and best-case rules in the partial fusion should be considered. Moreover, we plan to explore new restriction functions. A significant challenge in this endeavor is quickly calculating the minimum (for optimization) and maximum (to verify feasible set membership) restriction for all the observations in a rule. Additionally, strategies to optimize Explanation Set quality in the search process, rather than the best observation, will be investigated.

Regarding the performance, the method may convert a *RF* to a local *DT* in high-density areas to achieve zero-cost Explanation Sets using the full conversion method in a region of the input space. Explanation Sets could be efficiently extracted from the resulting DT by selecting the rule that satisfies the semifactual or counterfactual. Finally, the extraction method could be extended to support Gradient Boosting models by modifying the probability estimation method to be a weighted sum of the individual DTs.

6.3. List of publications

Published scientific articles:

- Fernández, R. R., De Diego, I. M., Aceña, V., Fernández-Isabel, A., & Moguerza, J. M. (2020). Random forest explainability using counterfactual sets. Information Fusion, 63, 196-207.
- Fernández, R. R., de Diego, I. M., Moguerza, J. M., & Herrera, F. (2022). Explanation sets: A general framework for machine learning explainability. Information Sciences, 617, 464-481.

International conferences contributions:

- Fernández, R. R., de Diego, I. M., Aceña, V., Moguerza, J. M., & Fernández-Isabel, A. (2019). Relevance metric for counterfactuals selection in decision trees. In Intelligent Data Engineering and Automated Learning–IDEAL 2019: 20th International Conference, Manchester, UK, November 14–16, 2019, Proceedings, Part I 20 (pp. 85-93). Springer International Publishing.

Appendix A

Explanation Set existence in Random Forest outside the label domain

Proposition 1. *Given a dissimilarity m like Equation 3.7, a counterfactual Explanation Set can not be defined in a Random Forest regressor for a factual sample whose prediction is equal to the maximum value of the label in the training set.*

Proof. Let X be the training set and Y the associated labels. The prediction in a Random Forest regressor is the average of the individual Decision Trees, whose predictions are also an average, in this case, of the labels of the instances that fall in that leaf. Let $y_{max} = max(Y)$ be the maximum value of the set of labels Y. Let $Y' \subseteq Y$ be a subset of Y. It is straightforward to see that $y_{max} \geq mean(Y')$, and thus, y_{max} is greater or equal to the prediction of the individual Decision Trees. Consequently, the Random Forest regressor can not make a prediction higher than y_{max} because it is an average of the Decision Trees. Since an Explanation Set requires at least one element that meets the grouping measure by definition, a counterfactual Explanation Set with the dissimilarity m can not be generated for an observation whose prediction is y_{max}.

□

Bibliography

Aceña, V., de Diego, I. M., Fernández, R. R., & Moguerza, J. M. (2022). Minimally overfitted learners: A general framework for ensemble learning. *Knowledge-Based Systems*, *254*, 109669.

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

Adhikari, A., Tax, D. M., Satta, R., & Faeth, M. (2019). Leafage: Example-based and feature importance-based explanations for black-box ml models. In *2019 ieee international conference on fuzzy systems (fuzz-ieee)* (pp. 1–7).

Alatalo, J., Korpihalkola, J., Sipola, T., & Kokkonen, T. (2022). Chromatic and spatial analysis of one-pixel attacks against an image classifier. In *International conference on networked systems* (pp. 303–316).

Alvarez-Melis, D., & Jaakkola, T. S. (2018). On the robustness of interpretability methods. *arXiv preprint arXiv:1806.08049*.

Arrieta, A. B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., ... Herrera, F. (2020). Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information Fusion*, *58*, 82–115.

Artelt, A., & Hammer, B. (2020). 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* (pp. 353–365).

Arya, V., Bellamy, R. K., Chen, P.-Y., Dhurandhar, A., Hind, M., Hoffman, S. C., ... others (2019). *One explanation does not fit all: A toolkit and taxonomy of ai explainability techniques.* (arXiv preprint arXiv:1909.03012)

Barredo-Arrieta, A., & Del Ser, J. (2020). Plausible counterfactuals: Auditing deep learning classifiers with realistic adversarial examples. *arXiv preprint arXiv:2003.11323*.

Bastani, O., Kim, C., & Bastani, H. (2017). *Interpreting blackbox models via model extraction.*

Belle, V., & Papantonis, I. (2020). *Principles and practice of explainable machine learning.* (arXiv preprint arXiv:2009.11698)

Bergstra, J., Yamins, D., & Cox, D. (2013). Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In *International conference on machine learning* (pp. 115–123).

Bien, J., & Tibshirani, R. (2011). Prototype selection for interpretable classification. *projecteuclid*.

Blanchart, P. (2021). An exact counterfactual-example-based approach to tree-ensemble models interpretability. *arXiv preprint arXiv:2105.14820*.

Bodria, F., Giannotti, F., Guidotti, R., Naretto, F., Pedreschi, D., & Rinzivillo, S. (2023). Benchmarking and survey of explanation methods for black box models. *Data Mining and Knowledge Discovery*, 1–60.

Breiman, L. (2001). Random forests. *Machine learning*, *45*(1), 5–32.

Breiman, L. (2002). Manual on setting up, using, and understanding random forests v3. 1. 2002. *URL: http://oz. berkeley. edu/users/breiman/Using_ random_forests_ V3*, *1*.

Breiman, L. (2017). *Classification and regression trees.* Routledge.

Collins, E. (2018). Punishing risk. *Geo. LJ*, *107*, 57.

Cook, R. D. (1977). Detection of influential observation in linear regression. *Technometrics*, *19*(1), 15–18.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, *6*(2), 182–197.

Deng, H. (2014). Interpreting tree ensembles with intrees. *arXiv preprint arXiv:1408.5456*.

Deng, H. (2019). Interpreting tree ensembles with intrees. *International Journal of Data Science and Analytics*, *7*(4), 277–287.

Deng, H., & Runger, G. (2013). Gene selection with guided regularized random forest. *Pattern recognition*, *46*(12), 3483–3489.

Deng, H., Runger, G., Tuv, E., & Bannister, W. (2014). Cbc: An associative classifier with a small number of rules. *Decision Support Systems*, *59*, 163–170.

Dhurandhar, A., Chen, P.-Y., Luss, R., Tu, C.-C., Ting, P., Shanmugam, K., & Das, P. (2018). Explanations based on the missing: Towards contrastive explanations with pertinent negatives. In *Advances in neural information processing systems* (pp. 592–603).

Doshi-Velez, F., & Kim, B. (2017). *Towards a rigorous science of interpretable machine learning.* (arXiv preprint arXiv:1702.08608)

Dua, D., & Graff, C. (2017). *UCI machine learning repository.* Retrieved from `http://archive.ics.uci.edu/ml`

European Commission. (2023). *Regulatory framework proposal on artificial intelligence.* Retrieved 16/10/2023, from `https://digital-strategy.ec.europa.eu/en/policies/regulatory-framework-ai`

Fernández, R. R., de Diego, I. M., Aceña, V., Fernández-Isabel, A., & Moguerza, J. M. (2020). Random forest explainability using counterfactual sets. *Information Fusion*, *63*, 196–207.

Fernández, R. R., de Diego, I. M., Moguerza, J. M., & Herrera, F. (2022). Explanation sets: A general framework for machine learning explainability. *Information Sciences*, *617*, 464–481.

Fernández, R. R., Diego, I. M. d., Aceña, V., Moguerza, J. M., & Fernández-Isabel, A. (2019). Relevance metric for counterfactuals selection in decision trees. In *International conference on intelligent data engineering and automated learning* (pp. 85–93).

Gower, J. C. (1971). A general coefficient of similarity and some of its properties. *Biometrics*, 857–871.

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

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

Guidotti, R., Monreale, A., Matwin, S., & Pedreschi, D. (2020). Black box explanation by learning image exemplars in the latent feature space. In *Machine learning and knowledge discovery in databases: European conference, ecml pkdd 2019, würzburg, germany, september 16–20, 2019, proceedings, part i* (pp. 189–205).

Guidotti, R., Monreale, A., Ruggieri, S., Pedreschi, D., Turini, F., & Giannotti, F. (2018). Local rule-based explanations of black box decision systems. *arXiv preprint arXiv:1805.10820*.

Harman, R., & Lacko, V. (2010). On decompositional algorithms for uniform sampling from n-spheres and n-balls. *Journal of Multivariate Analysis*, *101*(10), 2297–2304.

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., … Oliphant, T. E. (2020, September). Array programming with NumPy. *Nature*, *585*(7825), 357–362. Retrieved from https://doi.org/10.1038/s41586-020-2649-2 doi: 10.1038/s41586-020-2649-2

Karimi, A.-H., Barthe, G., Balle, B., & Valera, I. (2020). Model-agnostic counterfactual explanations for consequential decisions. In *International conference on artificial intelligence and statistics* (pp. 895–905).

Kelleher, J. D., & Tierney, B. (2018). *Data science.* MIT Press.

Kim, B., Khanna, R., & Koyejo, O. O. (2016). Examples are not enough, learn to criticize! criticism for interpretability. In *Advances in neural information processing systems* (pp. 2280–2288).

Koh, P. W., & Liang, P. (2017). Understanding black-box predictions via influence functions. In *Proceedings of the 34th international conference on machine learning-volume 70* (pp. 1885–1894).

Krishnan, S., & Wu, E. (2017). Palm: Machine learning explanations for iterative debugging. In *Proceedings of the 2nd workshop on human-in-the-loop data analytics* (pp. 1–6).

Larson, J., Angwin, J., Kirchner, L., & Mattu, S. (2016, May). *How we analyzed the compas recidivism algorithm.* Retrieved from https://www.propublica.org/article/how-we-analyzed-the-compas-recidivism-algorithm

Laugel, T., Lesot, M.-J., Marsala, C., Renard, X., & Detyniecki, M. (2017). *Inverse classification for comparison-based interpretability in machine learning.* (arXiv preprint arXiv:1712.08443)

Laugel, T., Lesot, M.-J., Marsala, C., Renard, X., & Detyniecki, M. (2019). The dangers of post-hoc interpretability: Unjustified counterfactual explanations. *arXiv preprint arXiv:1907.09294*.

Lipton, P. (1990). Contrastive explanation. *Royal Institute of Philosophy Supplements*, *27*, 247–266.

Lundberg, S. M., Erion, G. G., & Lee, S.-I. (2018). Consistent individualized feature attribution for tree ensembles. *arXiv preprint arXiv:1802.03888*.

Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances in neural information processing systems*, *30*.

Malik, N. (2020). Does machine learning amplify pricing errors in housing market?: Economics of ml feedback loops. *Economics of ML Feedback Loops (September 18, 2020)*.

McCloy, R., & Byrne, R. M. (2002). Semifactual "even if" thinking. *Thinking & Reasoning*, *8*(1), 41–67.

Miller, T. (2018). Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*.

Mittelstadt, B., Russell, C., & Wachter, S. (2019). Explaining explanations in ai. In *Proceedings of the conference on fairness, accountability, and transparency* (pp. 279–288).

Mohammadi, K., Karimi, A.-H., Barthe, G., & Valera, I. (2021). Scaling guarantees for nearest counterfactual explanations. In *Proceedings of the 2021 aaai/acm conference on ai, ethics, and society* (pp. 177–187).

Molnar, C. (2018). *Interpretable machine learning.* Lulu.com.

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

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Pentland, A. (2013). The data-driven society. *Scientific American*, *309*(4), 78–83.

Plumb, G., Al-Shedivat, M., Xing, E., & Talwalkar, A. (2019). *Regularizing black-box models for improved interpretability (hill 2019 version).* (arXiv preprint arXiv:1906.01431)

Poyiadzi, R., Sokol, K., Santos-Rodriguez, R., De Bie, T., & Flach, P. (2020). Face: feasible and actionable counterfactual explanations. In *Proceedings of the aaai/acm conference on ai, ethics, and society* (pp. 344–350).

Quinlan, J. R. (1986). Induction of decision trees. *Machine learning*, *1*, 81–106.

Quinlan, J. R. (2014). *C4. 5: programs for machine learning.* Elsevier.

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 1135–1144).

Ribeiro, M. T., Singh, S., & Guestrin, C. (2018). Anchors: High-precision model-agnostic explanations. In *Aaai* (Vol. 18, pp. 1527–1535).

Roese, N. J., & Epstude, K. (2017). The functional theory of counterfactual thinking: New evidence, new challenges, new insights. In *Advances in experimental social psychology* (Vol. 56, pp. 1–79). Elsevier.

Ruggieri, S. (2004). Yadt: Yet another decision tree builder. In *16th ieee international conference on tools with artificial intelligence* (pp. 260–265).

Sagi, O., & Rokach, L. (2020). Explainable decision forest: Transforming a decision forest into an interpretable tree. *Information Fusion*.

Schwalbe, G., & Finzel, B. (2023). A comprehensive taxonomy for explainable artificial intelligence: a systematic survey of surveys on methods and concepts. *Data Mining and Knowledge Discovery*, 1–59.

Sherman, S. J., & McConnell, A. R. (1995). Dysfunctional implications of counterfactual thinking: When alternatives to reality fail us. *What might have been: The social psychology of counterfactual thinking*, 199–231.

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

Su, J., Vargas, D. V., & Sakurai, K. (2019). One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, *23*(5), 828–841.

Tolomei, G., Silvestri, F., Haines, A., & Lalmas, M. (2017). Interpretable predictions of tree-based ensembles via actionable feature tweaking. In *Proceedings of the 23rd acm sigkdd international conference on knowledge discovery and data mining* (pp. 465–474).

Ustun, B., Spangher, A., & Liu, Y. (2019). Actionable recourse in linear classification. In *Proceedings of the conference on fairness, accountability, and transparency* (pp. 10–19).

Van Giffen, B., Herhausen, D., & Fahse, T. (2022). Overcoming the pitfalls and perils of algorithms: A classification of machine learning biases and mitigation methods. *Journal of Business Research*, *144*, 93–106.

Verma, S., Dickerson, J., & Hines, K. (2020). *Counterfactual explanations for machine learning: A review.* (arXiv preprint arXiv:2010.10596)

Von Kügelgen, J., Karimi, A.-H., Bhatt, U., Valera, I., Weller, A., & Schölkopf, B. (2022). On the fairness of causal algorithmic recourse. In *Proceedings of the aaai conference on artificial intelligence* (Vol. 36, pp. 9584–9594).

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

Wachter, S., Mittelstadt, B., & Russell, C. (2018, March). *Counterfactual Explanations without Opening the Black Box: Automated Decisions and the GDPR.* (arXiv: 1711.00399)

White, A., & Garcez, A. d. (2019). *Measurable counterfactual local explanations for any classifier.* (arXiv preprint arXiv:1908.03020)

White, K., & Lehman, D. R. (2005). Looking on the bright side: Downward counterfactual thinking in response to negative life events. *Personality and Social Psychology Bulletin*, *31*(10), 1413–1424.

Wolff, A., Gooch, D., Montaner, J. J. C., Rashid, U., & Kortuem, G. (2016). Creating an understanding of data literacy for a data-driven society. *The Journal of Community*

Informatics, *12*(3).

Yeh, I.-C. (2007). *Concrete Compressive Strength.* UCI Machine Learning Repository. (DOI: https://doi.org/10.24432/C5PK67)

Yousefzadeh, R., & O'Leary, D. P. (2019). *Interpreting neural networks using flip points.* (arXiv preprint arXiv:1903.08789)

Zhou, Y., & Hooker, G. (2016). Interpreting models via single tree approximation. *arXiv preprint arXiv:1610.09036*.

Zhu, J., Liapis, A., Risi, S., Bidarra, R., & Youngblood, G. M. (2018). Explainable ai for designers: A human-centered perspective on mixed-initiative co-creation. In *2018 ieee conference on computational intelligence and games (cig)* (pp. 1–8).