Desirability Rating Based Counterfactual (DeRaC) Framework for Multi-Dimensional Classification Problems

Neelabh Kshetry and Mehmed Kantardzic *

Posted Date: 18 February 2026

doi: 10.20944/preprints202602.1316.v1

Keywords: counterfactual explanations; explainable AI (XAI); algorithmic recourse; machine learning; interpretability; causal inference

 doi:10.20944/preprints202602.1316.v1

Article

Desirability Rating Based Counterfactual (DeRaC) Framework for Multi-Dimensional Classification Problems

Neelabh Kshetry and Mehmed Kantardzic *

University of Louisville, 2301 South 3rd Street, Louisville, KY 40292, USA
* Correspondence: mmkant01@louisville.edu

Abstract

Counterfactual explanations are increasingly vital for understanding and trusting machine learning models. This study presents, Desirability Rating based Counterfactual (**DeRaC**), a generalized framework for generating valid counterfactual explanations applicable to multi-dimensional classification problems, including single and multi-output classification with binary and multi-label outputs. By expanding the definition of counterfactual validity through a novel "desirability rating," the approach addresses limitations in existing methods for complex output spaces. This work details a novel framework, introducing concepts like partially valid counterfactuals and a quantitative measure of output desirability, which can be used with objective functions to find counterfactuals that also satisfy the various existing properties such as similarity, proximity, validity, actionability, etc. Experiments demonstrate the feasibility of systematically generating counterfactuals using existing optimization techniques, achieving varying degrees of validity and similarity. The research emphasizes the context-dependent nature of counterfactuals and lays the foundation for more transparent and trustworthy machine learning systems.

Keywords: counterfactual explanations; explainable AI (XAI); algorithmic recourse; machine learning; interpretability; causal inference

1. Introduction

Machine Learning and Artificial Intelligence models are being increasingly deployed in decision-making and decision-assisting systems across critical domains like finance (credit, insurance, e-commerce, etc.) [1], law enforcement (facial recognition) [2], cybersecurity (threat detection) [3], and more. These models often utilize complex algorithms, sometimes operating as "black boxes" where their internal workings are opaque. Explaining these AI-ML systems is challenging yet crucial for building trust, identifying biases, and ensuring fairness. A prominent technique within eXplainable Artificial Intelligence (XAI) is post-hoc explanation, focusing on understanding model decisions rather than the models themselves. Counterfactual explanation [4] is a widely used post-hoc method.

Counterfactual instances are generated based on a sample instance receiving a specific prediction from a model. They represent instances with minimal differences from the original instance, designed to yield a predetermined, or "desired," output. In Figure 1, a classification model $\mathtt{b}_1$ predicts "electrical failure" based on features "predictor$_1$" and "predictor$_2$". Given that the region inside the model $\mathtt{b}_1$ represents positive classifications, instance $\mathtt{x}$ receives a positive prediction. If the desired output is negative classification, $\mathtt{x}''$ achieves this while $\mathtt{x}'$ does not, making $\mathtt{x}''$ a "valid" counterfactual and $\mathtt{x}'$ an "invalid" one [5]. A typical counterfactual algorithm takes the original instance ($\mathtt{x}$), the prediction model ($\mathtt{b}_1$), and the desired output as input, seeking a solution that balances proximity to the original instance with adherence to the desired prediction, while considering factors like proximity, actionability and similarity.

 doi:10.20944/preprints202602.1316.v1

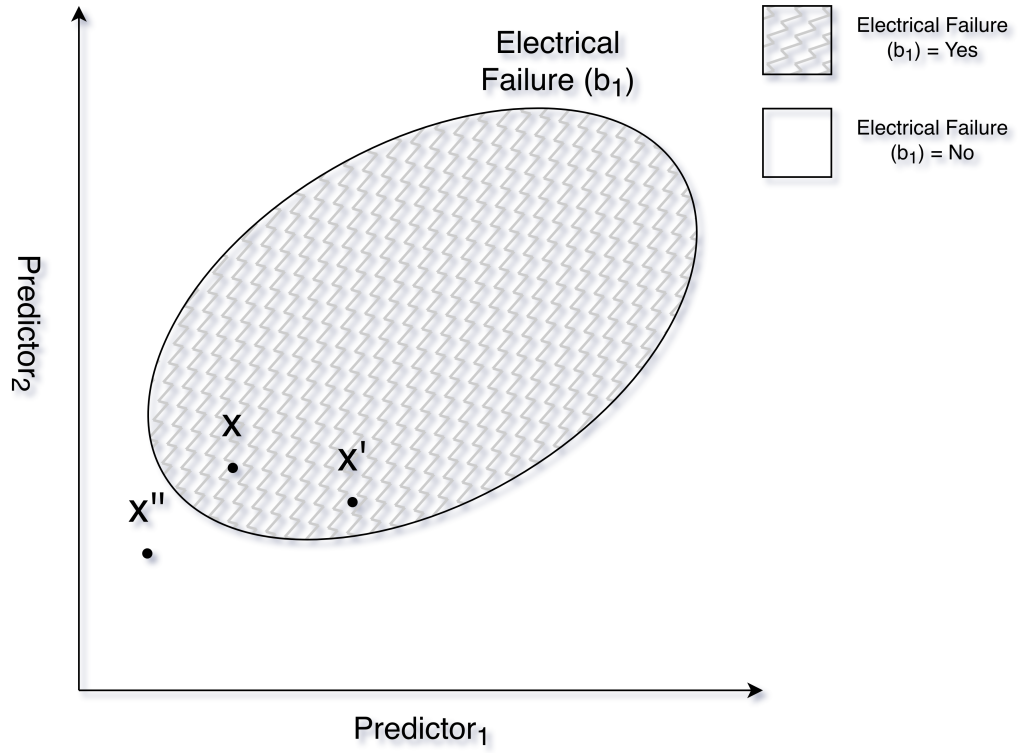

Figure 1. Example of valid and invalid Counterfactual Instance x' and x'' based on Original Instance x and Classification Model b_1

While counterfactual explanations offer a powerful means of understanding AI/ML model decisions, practical application is hampered by challenges in high-dimensional spaces. Existing approaches often rely on very loose definitions of "validity", requiring a counterfactual instance simply to "change" the original output. This is particularly ambiguous in multi-dimensional classification, where there exists multiple "different" outcome from the original outcome of the instance, and often not all of the "different" outcomes are equal. We propose a new framework to expand the definition and method of counterfactual generation that handles the nuances raised by the multi-dimensional classification problems (we will discuss this further in section 2. Our contributions, therefore, lie in:

1. formalizing the **DeRaC** framework for counterfactual generation for multi-dimensional classification problems,
2. establishing partially valid counterfactuals, and demonstrating its use,
3. optimizing multi-dimensional counterfactual generation based on critical properties.

2. Ambiguity of Counterfactual in Multi-Dimensional Classification Problems

While counterfactual explanations are well-established for single-output classification, applying them to more complex multi-dimensional problems, encompassing multi-label, multi-output, and combinations thereof, introduces significant ambiguity. In traditional binary classification, finding a "different" prediction is sufficient to define a "valid" counterfactual. However, this approach becomes inadequate when dealing with multiple classes or outputs.

2.1. Multi-Label Classification Problems

Consider a multi-label classification problem, such as the Iris dataset [6], where the target variable "class" has 3 classes "Iris Setosa", "Iris Versicolour", or "Iris Virginica". As shown in Figure 2, an instance ($\mathtt{x}$) with a predicted class of $\mathtt{C}_2$ has two possible "different" predictions. Treating all these as equally "valid" counterfactuals overlooks the nuance of their usefulness. The validity of a counterfactual hinges on achieving a "desirable" output, which is context-dependent and tied to the original

 doi:10.20944/preprints202602.1316.v1

instance and specific use case. It is also important to note that multi-label problems involve categorical classes that are mutually exclusive; an instance can only belong to one class at a time.

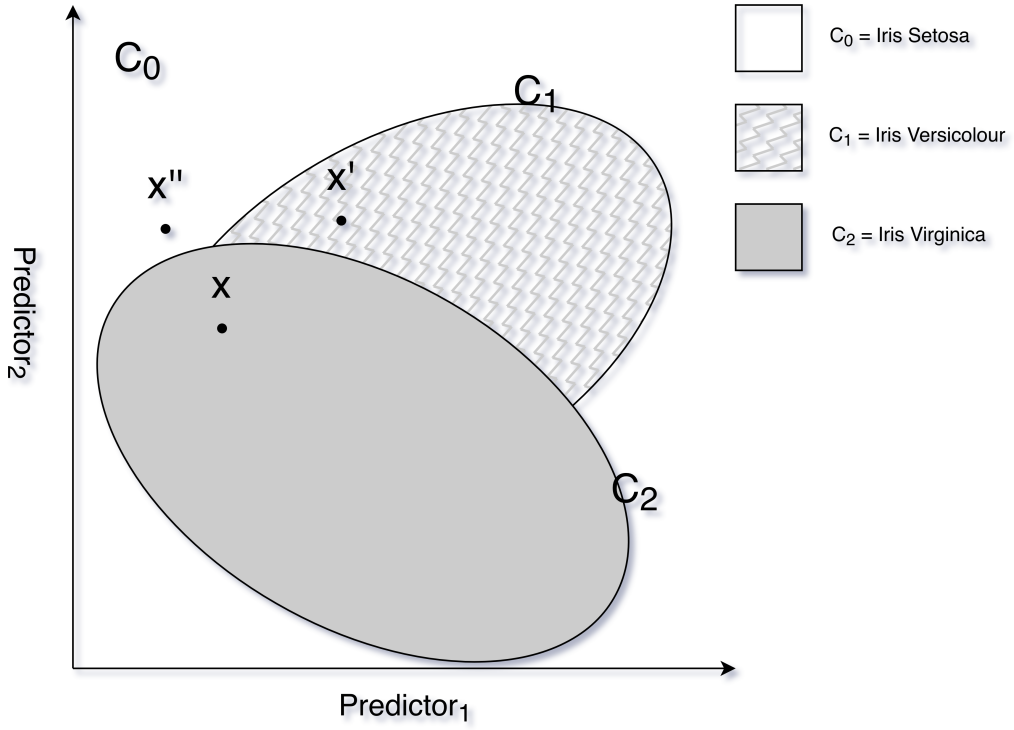

Figure 2. Classification for multi-class (3 classes) problem where the classes are c_0, c_1, and c_2

We can further divide multi-label problems into nominal and ordinal classifications. In nominal problems, classes have no inherent order (like the example above with Iris dataset [6]). However, in ordinal problems, classes have a defined order, impacting the assessment of desirability. For example, in the frequently used Wine Quality dataset [7], a prediction of 7 is more desirable than 4 if the goal is to achieve a higher value (such as 9).

2.2. Multi-Output and Multi-Dimensional Classification Problems

Beyond multi-label problems, we encounter multi-output classification, where multiple independent classifiers predict different outputs, as illustrated in Figure 3. Unlike multi-label problems, outputs in multi-output scenarios are not mutually exclusive. A combination of both multi-label and multi-output structures forms multi-dimensional classification problems. The core challenge remains consistent: simply finding a "different" prediction is insufficient. The validity of a counterfactual depends on achieving a combination of desirable outcomes across all outputs, which requires defining and quantifying what constitutes a "desirable" state for each dimension and the overall problem. Therefore, a more robust framework to evaluating counterfactual validity is necessary for effective explanation in these complex scenarios.

 doi:10.20944/preprints202602.1316.v1

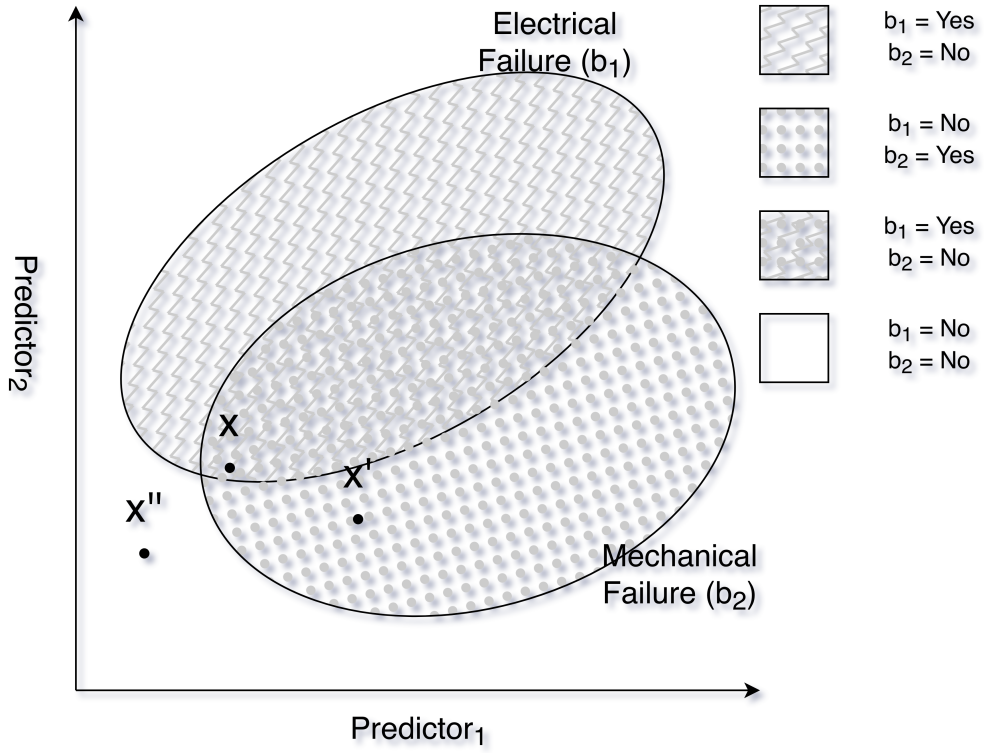

Figure 3. 2-output binary classification problem where outputs o_1, o_2 are modeled by b_1, b_2 each giving either `Yes` or `No` class

This gap leaves fundamental concepts like "desired-undesired" outputs and "valid-invalid" counterfactuals ill-defined in the context of multi-dimensional classification systems. To address this, this paper formalizes counterfactual explanations for any combination of multi-label and multi-output problems, introducing the notion of partially valid counterfactuals and demonstrating their utility for model explanation via counterfactual explanations. We further present an optimization framework for generating these counterfactuals, incorporating key properties such as validity, proximity, and similarity.

3. Related Work

Explainable Artificial Intelligence (XAI) has emerged as a critical field of research in recent years, driven by the increasing deployment of complex AI systems across diverse applications [8]. The core challenge of XAI lies in bridging the 'black box' nature of many AI models, particularly deep learning, by providing humans with understandable explanations of how these systems arrive at their decisions. According to Ali et al. [9], this landscape can be divided into data explainability (focused on explaining the training data through training and AI models), model explainability (focused on creating AI models that are naturally more explainable), and post-hoc explainability (focused on explaining AI model decisions after the fact). More recently, research has increasingly centered on counterfactual explanations, a post-hoc explanation method, which offer a specific approach to addressing this challenge by identifying the minimal changes to an input that would alter a model's prediction [5].

3.1. Multi-Dimensional Problems in Machine Learning

Multi-label and multi-output problems are increasingly prevalent in various fields, driving significant research into machine learning solutions. These problems involve predicting multiple outputs simultaneously, moving beyond traditional single-target prediction. As highlighted in several studies, this approach is crucial for tasks like predicting chemical parameters of river water quality from biological data [10], forecasting blood-drug concentrations in time series data [11], and estimating

 doi:10.20944/preprints202602.1316.v1

multiple biophysical parameters from remote sensing images [11]. The ability to model relationships between multiple dependent variables offers more comprehensive and accurate predictions than treating each output in isolation.

Machine learning techniques, particularly regression trees and their ensembles, are frequently employed to tackle these complex problems. Researchers have explored methods like multi-target regression trees, model trees with linear models in leaves, and adaptations of multi-label classification techniques for regression tasks [12]. These methods aim to capture the correlations between different outputs and improve predictive performance. Furthermore, problem transformation methods convert multi-target regression into multiple single-target problems, offering another avenue for solution [12]. The use of algorithms like support vector regression, ridge regression, and even simpler approaches like linear regression demonstrate the versatility of machine learning in this domain [12,13].

The effectiveness of these machine learning approaches in regression domain is validated through rigorous evaluation metrics such as correlation coefficients and root mean squared error (RMSE), often assessed using cross-validation techniques [12]. The small confidence intervals observed in some studies indicate the stability and reliability of these algorithms when applied to sufficiently large datasets. This demonstrates the growing importance of machine learning in addressing multi-output and multi-label challenges, enabling more informed decision-making and deeper insights in diverse application areas.

3.2. XAI and Introduction of Counterfactual Explanation

Wachter et al. [14] introduced the concept of counterfactual explanations as a novel method in the XAI space. The paper argues that the explanations of model decisions should focus on external factors that can be changed rather than focusing on internal workings of the algorithms. A counterfactual of an instance `x` classified by a model `b` as `y = b(x)` is an instance `x′`, where the model's decision is different; i.e., `b(x′) != y`, and the difference between `x` and `x′` is minimal [4,5]. A counterfactual explainer [5] is a function that takes a classifier `b`, known instances `X`, and an instance `x` and returns a set of *valid* counterfactual examples $C = \{x'_1, \ldots, x'_h\}$ such that $b(x'_i) \neq b(x)$ for all x'_i in C.

These definitions are primarily centered around changing a single prediction outcome (`b(x′) != b(x)`) for a specific instance, as is typical in classification, where the output is a single class label. The Guidotti survey [5] explicitly states its analysis and categorization focus on counterfactual explanation methods designed to explain black-box classification models because the large majority of the literature addresses this problem. Despite a number of surveys [4,15–19] that have explored benchmarking, identified deficits, and conducted analyses of existing counterfactual explanation research, a common thread emerges: almost all of these studies predominantly assess single-output binary classification problem. Guidotti [5] also defines the properties of desirable counterfactuals which is summarized in Table 1.

3.3. Current Techniques for Counterfactual Generation

While the foundational concepts of counterfactual explanations have been established [14], a diverse range of methods have been developed to generate these explanations in practice. Many approaches formulate the search for counterfactuals as an optimization problem, aiming to find the smallest perturbation to an input that flips the model's prediction. Some works such as [20–22] focus on different methods of using generative adversarial networks to generate counterfactuals. Other works focus on different ways to efficiently solve the objective function and calculate counterfactuals such as using fuzzy decision trees [23], solving for multiple constraints [24], using ensemble methods [25], using program syntheis [26], or solving other problems of optimizing [27–29].

 doi:10.20944/preprints202602.1316.v1

Table 1. Desirable Properties of Counterfactual Explanations [5]

Property	Description
Validity	A counterfactual x' is valid if it changes the classification outcome: $b(x') \neq b(x)$.
Minimality (Sparsity)	x' is minimal if it has the fewest attribute changes compared to other valid counterfactuals x''.
Similarity (Proximity)	x' should be close to x based on a distance function d: $d(x, x') < \epsilon$, where ϵ is a predefined threshold. Also referred to as proximity.
Plausibility	x' should have feature values consistent with a reference population X. Values should be realistic and not outliers within X. Also known as feasibility or reliability.
Discriminative Power	x' should clearly demonstrate the reasons for the change in prediction. A human observing x and x' should understand the differing classification.
Actionability	x' should only differ from x in terms of actionable features (features that can be changed). Immutable features should remain constant. Also known as recourse.
Causality	x' should respect known causal relationships between features, as defined by a Directed Acyclic Graph (DAG). Changes in features should maintain established causal links.
Diversity	A set of counterfactuals $C = \{x'_1, ..., x'_h\}$ should be diverse, maximizing the difference between the counterfactuals while maintaining minimality and similarity.

Other works have focused on developing tools to generate counterfactuals such as the decision explorer (DECE) by Cheng et al. [30], which is an interactive framework to allow users to explore model decisions and generate counterfactuals. The what-if tool (WIT) [31] and language interpretability tool (LIT) [32], both developed by research teams at Google Research offer functionalities directly related to counterfactual explanations. WIT allows users to automatically identify counterfactual examples, aiding in understanding model decisions . Similarly, LIT provides first-class support for counterfactual generation, enabling users to add new data points and immediately visualize their effect on model predictions, and facilitates side-by-side comparisons. These tools acknowledge the importance of counterfactuals as a method for interpreting model behavior and offer practical ways to explore these explanations.

There are also other tools that help to visualize and explain classifiers using counterfactuals such as Prospector [33], RuleMatrix [34], SystemD [35], ViCE [36], and WiXAI [37]. There also exists a significant amount of work done specifically on non-tabular data such as images, text, time-series datasets. Works such as [38], focus on visual counterfactuals identifying how an image could be changed to alter a vision system's prediction. Their method involves finding regions in a 'query' image and a 'distractor' image (predicted as the desired alternative class) and swapping them to shift the prediction. This is distinct from simply highlighting important features; it focuses on minimal edits to achieve a different outcome. Other research works such as [39–44]. For text, research focuses on automated counterfactual generation, acknowledging the challenge of maintaining natural language while making minimal changes such as [45]. These tools and techniques collectively highlight the growing interest in leveraging counterfactual explanations for model debugging, fairness analysis, and increased transparency in machine learning systems.

3.4. Applications and Extensions of Counterfactual Explanations

Beyond the core definition and generation of counterfactuals, research in this area is diversifying into several key directions. Some work focuses on improving the quality of counterfactuals themselves, emphasizing plausibility, actionability, and diversity. For instance, Arie et al. [46] propose making counterfactual-based analysis more optimized and time-efficient by using databases, while Piccialli et al. [47] propose utilizing counterfactual explanations to detect important decision boundaries and training a more compact easily explainable model like decision tree that closely resembles the original

 doi:10.20944/preprints202602.1316.v1

black-box model. Gao et al. [48], propose a self-training model that uses pseudolabels generated from a base classification model that uses those pseudolabels to retrain combined with factual data. Their method is iterative and uses counterfactual virtual adversarial training (CVAT) to ensure the models don't get stuck with incorrect guesses.

Furthermore, Temraz et al. [49] propose an interesting approach to solving issues with imbalanced datasets by using counterfactual augmentation (CFA). The CFA method generates counterfactuals based on the factual dataset in the minority class, thus producing synthetic data points rather than oversampling. Another interesting approach by Sohns et al. [50], propose method of visualizing decision boundary for counterfactual reasoning using local linear maps of the decision space and combining it with other model inspection techniques. Our previous work [37] also created a tool, WiXAI, that allows users to explore the decision boundaries via an interactive and iterative visual perturbation of samples, which can help move them towards causal understanding of the relationship between features at a sample-level.

3.5. Counterfactual explanation for Causal Understanding

Since the introduction of Counterfactual explanations by Wachter et al. [14], counterfactuals are increasingly recognized for their potential to enhance understanding and interpretability, particularly within causal inference. These explanations, which detail how an input would need to change to achieve a different outcome, are not merely about prediction but also offer insights into the underlying causal mechanisms [51]. Researchers emphasize the importance of ensuring these counterfactuals are feasible, reflecting real-world constraints and causal relationships rather than simply statistical likelihood [51–53]. Generating feasible counterfactuals necessitates moving beyond statistical approaches and explicitly incorporating causal models [51], acknowledging that changes to input features must adhere to natural laws and feature interactions [52].

Recent work focuses on formulating feasibility as a causal concept, demanding that counterfactual perturbations respect the causal structure between input features [51]. Approaches like CEILS aim to bridge the gap between XAI counterfactuals and causal counterfactuals, generating explanations that are both interpretable and actionable by providing causally feasible actions [54]. Evaluating these explanations requires metrics like causal-constraint validity, which assesses the proportion of counterfactuals aligning with domain knowledge and causal frameworks [52], highlighting the shift toward more robust and reliable counterfactual generation for causal understanding.

3.6. Counterfactual for Multi-Dimensional Classification Problems

Despite the evident use of machine learning techniques in solving multi-label and multi-output problems, the use of counterfactual explanations is quite limited in this problem domain. Most of the existing literature focuses primarily on single-output binary classification. As such, the definition of changing the output suffices in finding counterfactuals. Existing research that work with non-binary classification problems, such as the work by Carlevaro et al. [55] on multi-label classification problems follow the same idea of "different output" to define valid counterfactuals. Any classification that is different than the original prediction is treated as valid counterfactuals. We will discuss in section 2 why such a definition can be ambiguity. While methods like those presented in Caron et al. [56] extend to multi-task learning via deep kernels, the focus remains on estimating causal effects and learning policies and not necessarily on generating easily interpretable counterfactuals for each individual output.

Furthermore, the challenge is compounded when considering combinations of treatments, as highlighted by Parbhoo et al. [57]. Addressing the exponential growth of possible treatment combinations necessitates scalable modeling approaches, but doesn't inherently solve the problem of generating understandable counterfactuals for each outcome within a multi-output setting. Current work often simplifies the problem by focusing on single-label or single-output scenarios, leaving a gap in understanding how to effectively generate actionable counterfactual explanations when multiple outputs or multiple labels are simultaneously considered.

 doi:10.20944/preprints202602.1316.v1

4. Desirability Rating based Counterfactual Framework for Multi-Dimensional Classification Problem

Counterfactual explanations offer a powerful approach to understanding model decisions by identifying minimal changes to an input that would lead to a different prediction. These changes are generated relative to an original instance, the decision model, and a specified desired output. Formally, a counterfactual $\mathtt{x}\prime$ is a minimally perturbed instance of $\mathtt{x}$ such that $\mathtt{b(x)} \neq \mathtt{b(x}\prime\mathtt{)}$, where $\mathtt{b}$ represents the model [5]. A "valid" counterfactual, in this context, is one that is sufficiently close to the original instance while successfully altering the model's prediction. However, as discussed in section 2, a more robust framework is needed in cases of multi-dimensional classification problems rather than the simple "different" classification. This section details a framework for generating counterfactual explanations specifically designed for multi-dimensional classification problems, outlining how we define and evaluate the "desirability" of counterfactual instances, and how we categorize them as "valid" or "partially valid." It is important to note that this extended framework applies not only to multi-label and multi-output problems but also existing binary and single-output classification problems.

4.1. Key Definitions and Scopes

Before detailing our **DeRaC** framework, it's crucial to establish the fundamental concepts we'll be working with. These definitions provide a common understanding of the problem space and how we approach generating and evaluating counterfactuals.

1. Type of Problem: This refers to the nature of the classification task. We consider a broad spectrum, including:

- *Binary Classification:* A single output variable with two possible values.
- *Nominal Multi-Class Classification:* Multiple output classes without inherent order.
- *Ordinal Multi-Class Classification:* Multiple output classes with a defined order or ranking.
- *Multi-Label Classification:* Multiple output classes with or without inherent order (includes both nominal and ordinal multi-class classification).
- *Multi-Output Classification:* Multiple, independent output variables, each with its own classification task.

2. Instances: These are the individual data points fed into the model. Formally, an instance $\mathtt{x}$ is a vector of features representing a single observation. The quality and relevance of these instances are crucial for the generation of meaningful counterfactuals. An instance can also be referred to as a sample or data point.

3. Desired Output: This is the target output we aim to achieve through the counterfactual explanation. It can be a single class label (in binary or multi-class classification), a set of labels (in multi-label classification), or a vector of values (in multi-output classification). The definition of the desired output directly influences the search space for counterfactuals. The desired output is context-dependent (depending on the user as well as the use-case).

4. Desirability Rating: A metric used to quantify how close an instance's output is to the desired output. The calculation of this rating depends on the *Type of Problem* as detailed next in Section 4.2. This forms the foundational basis of our Desirability Rating based Counterfactual (**DeRaC**) Framework.

5. Counterfactual Goal: The objective of finding a counterfactual is not merely to change the prediction, but to do so with minimal changes to the original instance. Our goal is to identify the smallest perturbation to $\mathtt{x}$ that results in an output closer to the *Desired Output*, as measured by the *Desirability Rating* while respecting the properties of effective Counterfactual Explanations previously defined by others [5].

4.2. Desirability Rating of Outputs and Instances

The goal of the metric "desirability rating" is the measure how desirable the output of an instance is given a predefined "desired" output. For single output binary classification problems, it is binary

 doi:10.20944/preprints202602.1316.v1

(either 'desired' or 'undesired'). Similarly, for nominal multi-class classification problems, since there is no preference between the undesired outputs, it is also binary score of 'desired' or 'undesired' output. We can see this function of desirability of an output in case of binary or nominal multi-label classification in equation 1. Here, the value 1 represents an output class `y` as "desired" and the value 0 represents an output class as "undesired".

$$d(y) = \begin{cases} 1, & \text{if } y \in S_d \\ 0, & \text{if } y \notin S_d \end{cases} \tag{1}$$

where, S_d is the set of desired output classes.

However, for ordinal multi-class classification problems, we have to consider whether an output is desired or close to desired output. If we lay out the ordinal classes in order and measure the ratio of how close the current output is from the desired output we can get the essence of how desirable is a certain output class. We can measure this as seen in equation 2 (represented by `d(·)`). From the equation, we can see that it is measured as the ratio of absolute difference between the ordinal number of closest desired output class (y_d) and the current output class (`y`) to maximum possible ordinal difference between two classes. While, binary classification and nominal multi-class classification leads to a desirability value of either `0` or `1`, in the case of ordinal multi-class classification, the value is a number in the range `[0, 1]`. A value of 0 represents, an undesired output, and a value of 1 represents a desired output, while a value in between can be considered partially desirable.

$$d(y) = 1 - \frac{|order(y_d) - order(y)|}{number\ of\ classes - 1} \tag{2}$$

where, y_d is the closest desired class and $y_d \in S_d$.

As we noted above, the complexity for measuring desirability of an output depends on the type of output. It can either be a binary value or a ratio between 0 and 1. Similarly, for multi-output problems, we can combine the desirability of each individual output to get an overall desirability score for an instance. The most straightforward combination would be to put equal weight to all the outputs and measure the what percentage of them have desired output. It is also possible to assign more or less weight to individual outputs depending on the problem. As such, the overall desirability rating of an instance can be measured as shown in equation 3. In equation 3, `x` represents the instance, `d(·)` represents desirability score for individual output, `w` represents the weight assigned to each specific output, and `D` represents the overall desirability rating of an instance for multi-output problems. Similar to equation 2, the value can be in the range of `[0, 1]`, with 0 representing undesired outcome, 1 representing desired outcome as well as a value in between representing partially desired outcome.

$$D(x) = \frac{sum(w_1 \times d(y_1), ..., w_n \times d(y_n))}{n} \tag{3}$$

where, $y_1 = b_1(x), y_2 = b_2(x), ..., y_n = b_n(x)$ represents the `n` outputs for instance `x`, and $w_1, ..., w_n$ represents weights assigned to each output that add up to `n`.

4.3. Valid and Partially Valid Counterfactual

As mentioned previously, Wachter et al. [14], describes counterfactuals in terms of desired outcomes, showing which external facts could be altered to achieve at a desired outcome. Counterfactual is thus framed in terms of "desired" and "undesired" outputs. Counterfactuals are declared 'valid' if they reach the desired outcome, otherwise they are treated as 'invalid'. For our problem of multi-label and multi-output classification, we have not only desired and undesired outcomes but also a spectrum of partially desired outcomes. While for single-output binary classification problems, a counterfactual was considered 'valid' if it reached the desired outcome, for these complex problems, it is possible to reach a partially desired outcome. As such, we also have to introduce the concept of partially

 doi:10.20944/preprints202602.1316.v1

valid counterfactual. A partially valid counterfactual can be described as a counterfactual that has desirability rating greater than the original instance but isn't completely desired.

$$D(x') > D(x) \text{ and } D(x') < 1 \tag{4}$$

Mathematically, it is shown in equation 4, where x' represents the partially valid counterfactual x represents the original instance, and $D(\cdot)$ represents the function to calculate the desirability rating of an instance from equation 3. A counterfactual that has the same or lower desirability rating than the original instance can be considered invalid counterfactual as it doesn't "progress" in terms of desirability. Alternatively, if an instance is already 'desirable', one can consider a different set of outputs as desired and recalculate desirability and find valid or partially valid counterfactual for that use-case.

4.4. Desirability Rating in the Counterfactual Search

The **DeRaC** framework outlined in Section 4 fundamentally alters the concept of counterfactual "validity" as traditionally defined by Guidotti et al. and others [5]. While existing definitions focus on a binary classification of counterfactuals as simply "valid" (prediction changes) or "invalid", our approach introduces a spectrum of validity through the *Desirability Rating* and the categorization of "Partially Valid" counterfactuals. This moves beyond a simple yes/no assessment and allows for *comparable* counterfactuals (some are demonstrably "more valid" than others, reflecting the degree to which they approach the desired output.

This added complexity is not merely academic, it unlocks new possibilities for counterfactual explanation generation. Instead of solely seeking any counterfactual that achieves the desired outcome, we can now formulate objective functions that explicitly optimize for higher *Desirability Ratings*. This allows us to prioritize counterfactuals that not only change the prediction but do so in a way that is *closer* to the desired output, offering more nuanced and potentially more actionable insights. These insights are exponentially more important when dealing with multi-dimensional classification problems.

Specifically, the *Desirability Rating* can be incorporated directly into the objective function used by counterfactual search algorithms. For instance, in an optimization process, the objective could be to minimize a combination of:

- **Perturbation Distance:** A measure of how much the counterfactual x' differs from the original instance x (e.g., L2 distance, Manhattan distance). This ensures minimal changes, a key principle of effective counterfactuals.
- **Negative Desirability Rating:** The negative of the *Desirability Rating* for the counterfactual. Maximizing the desirability rating is equivalent to minimizing its negative. This drives the search towards counterfactuals with higher validity.
- **Portion of Features Changed:** The fraction of features in x that are different in x'. This encourages minimality: counterfactuals with fewer changes are generally more actionable and easier to understand.

The relative weighting of these terms within the objective function allows for control over the trade-off between minimal perturbation and high desirability. A higher weight on the negative desirability rating prioritizes finding counterfactuals that are as close as possible to the desired output, even if it requires slightly larger perturbations.

Furthermore, incorporating the concept of "Partially Valid" counterfactuals opens the door to generating a *variety* of explanations. By systematically adjusting the acceptance threshold for the *Desirability Rating*, we can generate a diverse set of counterfactuals, ranging from those that fully achieve the desired outcome (Valid) to those that represent incremental steps towards it (Partially Valid). This variety is crucial for understanding the model's behavior in complex scenarios and for providing users with a more comprehensive picture of potential changes.

 doi:10.20944/preprints202602.1316.v1

It's crucial to remember that other properties of effective counterfactuals, such as plausibility, similarity, proximity, actionability, etc. [5], remain essential considerations. Our **DeRaC** framework complements these properties by adding a more nuanced quantifiable dimension of validity, enabling more sophisticated and targeted counterfactual generation strategies.

5. Finding Counterfactual with the DeRaC Framework (Experiments)

This section details the experiments conducted to evaluate the feasibility and effectiveness of defining and generating valid counterfactual explanations for multi-dimensional classification problems using the new **DeRaC** framework of desirability rating discussed in section 4. We evaluated our approach on three diverse datasets, focusing on the characteristics of generated counterfactuals with respect to proximity (L2 distance), validity (desirability rating), and optimization success. We compared the performance across different strategies for choosing the desired output for counterfactual generation simulating various different context-based desired output scenarios. We will later also contrast the results for multi-dimensional classification problems with those obtained on single-output binary classification problems to demonstrate universal compatibility.

5.1. Datasets Used

To ensure a robust evaluation, experiments were conducted using the datasets outlined in Table 2. These datasets vary in size, dimensionality, and class distribution, providing a comprehensive testbed for the proposed approach.

Table 2. Datasets Used in the Experiments

Dataset Name	Description	Output Type/Problem	Number of Outputs	Key Features
Mushroom Dataset [58]	Classifies mushrooms as edible or poisonous based on 22 features.	Categorical, Multi-label Classification (Edible/Poisonous + Habitat)	2	22 Features
Student Performance Dataset [59]	Information about students' performance in secondary school (grades, demographics).	Binary Classification (Above/Below Average)	3 (First, Second, Final Period)	Grades in First, Second, & Final Periods, Demographic Features
Wine Dataset [7]	Chemical properties of wines and a quality rating.	Ordinal Multiclass Classification	1	Chemical Properties
Adult Income Dataset [60]	Demographic features of individuals.	Binary Classification (Income $>$ or $<$ \$50K/year)	1	Education, Age, Marital Status, etc.

Note: The Wine and Student Performance dataset is treated as two separate problems (Red Wine & White Wine and Math & Portuguese respectively).

5.2. Model Training and Multi-Dimensional Classification

To use as the standard model for the basis of counterfactual generation, we trained four different model architectures: a Multi-Layer Perceptron (MLP), a Random Forest, a Support Vector Machine (SVM), and a Gradient Boosting Machine; on each output dimension independently. We employed a 5-fold cross-validation procedure with grid search for hyperparameter tuning to identify the best-performing model for each output. Specifically, for each output dimension, we selected the model

 doi:10.20944/preprints202602.1316.v1

configuration that maximized performance based on Root Mean Square Error for regression outputs, and accuracy and F1-score for classification outputs. Following model selection, we created a unified multi-output prediction model by integrating the best-performing individual model for each output dimension. This ensemble approach allowed us to predict all outputs simultaneously, providing a consistent basis for counterfactual generation.

5.3. Desired Output Selection

For each sample in the test set, we selected a desired output vector using one of the following strategies to simulate a variety of different scenarios of selecting desired output:

1. **Random:** A random output vector was chosen, effectively selecting random values for all outputs.
2. **Highest Value:** The desired output vector consisted of the highest observed value for each output dimension across the entire test set.
3. **Lowest Value:** The desired output vector consisted of the lowest observed value for each output dimension across the entire test set.
4. **Different Random Value:** For each output, a random value different from the original sample's output was chosen. This ensures the counterfactual is demonstrably different.

Counterfactual Generation: Using the trained model and the selected desired output, we generated a counterfactual instance using Powell's Method [61]. We used the desirability rating defined in section 4 to measure the validity (complete or partial) of the counterfactual instance. We verified the validity of each generated counterfactual by feeding it into the trained model and confirming that the model's predicted output matched the claimed desirability rating. This process was repeated for randomly selected **180** samples for a total of **11** runs for each combination of dataset (6 total datasets as explained in Table 2) and desired output selection strategy, allowing us to assess the robustness of our approach. This can be seen in Figure 4.

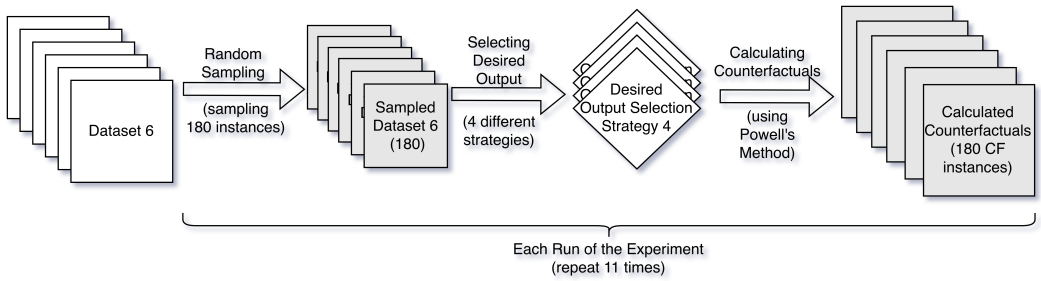

Figure 4. A schema of the experiment setup with 180 samples for each of the 6 datasets (from 4 data sources) repeated for 11 runs

5.4. Evaluation Metrics

We used the following metrics to evaluate the performance of our counterfactual explanation approach:

Average Distance: The average distance between the original input and the generated counterfactual across all samples and experimental runs. Lower distances indicate more similar counterfactuals.

Validity: The percentage of generated counterfactuals that passed the validity check. Higher validity indicates a more reliable generation process.

Optimization Success: The percentage of counterfactual searches that converged to a valid solution within a maximum number of iterations and within the specified distance threshold. This indicates the efficacy of the optimization algorithm.

5.5. Expected Results

We hypothesize that: (1) Defining valid and partially valid counterfactuals for multi-dimensional classification problems is systematically possible; (2) The desired output impacts the proximity, validity,

 doi:10.20944/preprints202602.1316.v1

and optimization success rate; and (3) The different strategies of selecting desired output will lead to different validity, proximity, and success rates (either partial or complete). We will combine these results to those obtained on single-output problem to demonstrate the feasibility of our approach for multi-dimensional classification scenarios.

6. Results

Even though we worked with four datasets, there were two versions each for both student performance dataset [59] and wine quality dataset [7]. This resulted in the number of datasets being six instead of four in our experiments. Among the methods tested, Random Forest and Multi-layer Perceptrons were the two that outperformed the rest in these experiments. The **scipy** library [62] was used to perform the optimization using Powell's Method [61]. On each random sample (instance), distance, validity of counterfactual, portion of features changed, desirability rating, etc. were measured.

In Figure 5, we can see the boxplot for the desirability rating of the original sample as well as the counterfactual sample for wine quality (white) dataset. In the figure, the light gray color represents the original desirability rating and the dark gray color represents it for counterfactuals, while the red line indicates the median. We can clearly see that the lower and upper limit of boxplot for original instances are at values `0.6` and `0.9`, while for counterfactual instances are at values `0.6` and `1.0`. Similarly, the median desirability rating for original instances is `0.7` and for counterfactual instances is `0.8`. This shows that our experiments systematically produce counterfactuals that have same or higher desirability rating than the original instances, thus shifting the median and range to higher values. This shows that for the multi-label classification problem of white wine quality, our **DeRaC** framework is successfully producing counterfactuals that are valid or partially valid, while avoiding counterfactuals that are less valid than original instances.

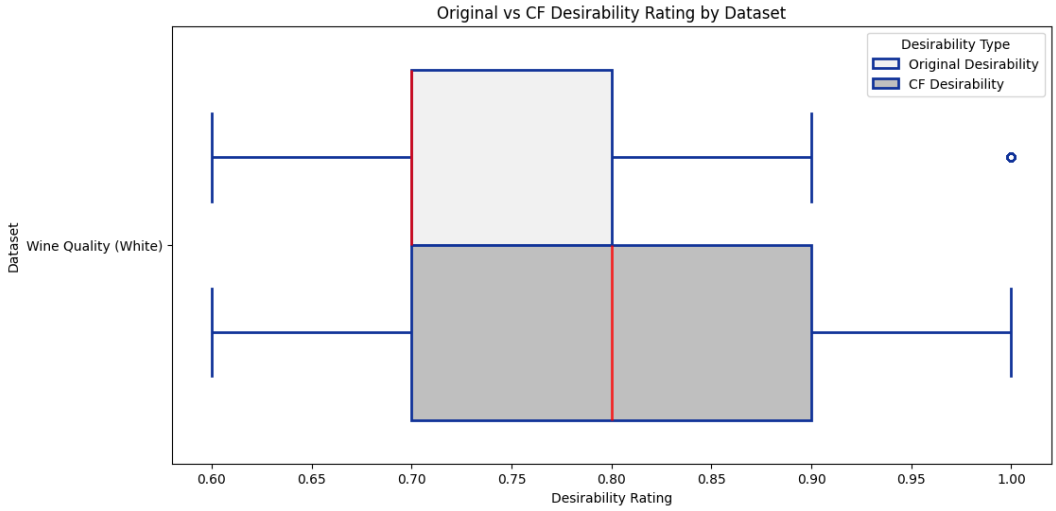

Figure 5. Boxplot of Original vs CF Desirability for white wine quality dataset

In Figure 6, we can see the boxplot for the desirability of the original sample and the counterfactual sample for each dataset. This is to show that across single-output binary classification, single output multi-label classification and multi-output classification problems, it is possible to calculate the desirability of instances. In Figure 6, we see that the desirability rating ranges between 0 and 1. Looking at the red line indicating the median, we see the difference between the median desirability rating of the counterfactual instance and original instance. In each dataset, the median desirability rating is higher for the counterfactual than the original instance, which shows that our method of finding counterfactual is able to generate some partially or completely valid counterfactual for all these problem types. We can verify this with Figure 7, which plots the average desirability rating per experiment run for the original instance against the counterfactual instance. In Figure 7, the blue line indicates the original and counterfactual desirability being the same. All the averages, indicated by

 doi:10.20944/preprints202602.1316.v1

colored dots, are above the line, which shows that systematically, the counterfactuals have a higher desirability rating making them at least partially valid on average.

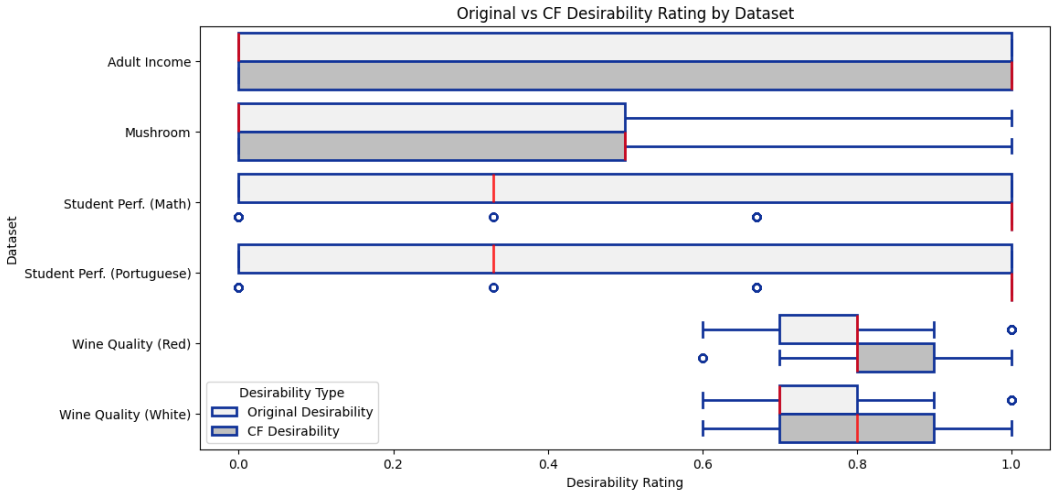

Figure 6. Boxplot of Original vs CF Desirability for each dataset showing systematic generation of valid counterfactuals

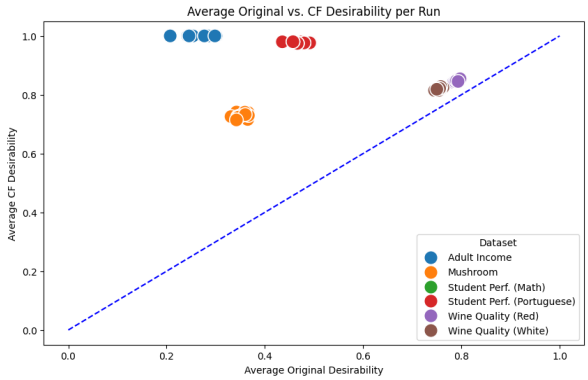

Figure 7. Average Original Desirability vs CF Desirability per run (NOTE: there are 11 slightly overlapping points representing 11 runs for each dataset)

It is important to note that the desirability rating is dependent on the desired output, which is not dependent on the dataset but rather the context (user or use-case). When seeking goal-achievement for the user, the desired output depends on the user's preferred outcome for each output feature. When looking into correctness of the prediction, the desired output may be the true values for each of the output features and in some instances like medical problems, the desired output might be whatever is the best outcome for the outputs. When seeking analysis of the model predictions itself, the desired output is dependent on the current prediction and what is to be analyzed. It is possible to try to test the robustness of the predictions by finding counterfactuals and the distance from the original or the number of features changed to compare how robust are the predictions. The longer distance to the counterfactual and more features changed, represents larger change needed to find the other prediction giving more robustness to the prediction.

We can also see in Figure 8, the average percentage of partially or completely valid counterfactuals found for each dataset based on the method of selecting the desired output. Here we see that the dataset perform differently in terms of the success rate of finding partially or completely valid counterfactual. This shows that the counterfactual generations depend on dataset and the model. However, we also see that within the same dataset, the different strategies of selecting the desired output also has some impact in the success rates. This reaffirms the notion of counterfactual being highly dependent on the individual cases as the desired outputs depend on those, as discussed in the paragraph above. To

 doi:10.20944/preprints202602.1316.v1

test for the robustness of model predictions, we look at the distances of counterfactual and portion of features changed, which can be seen in Figures 9 and 10.

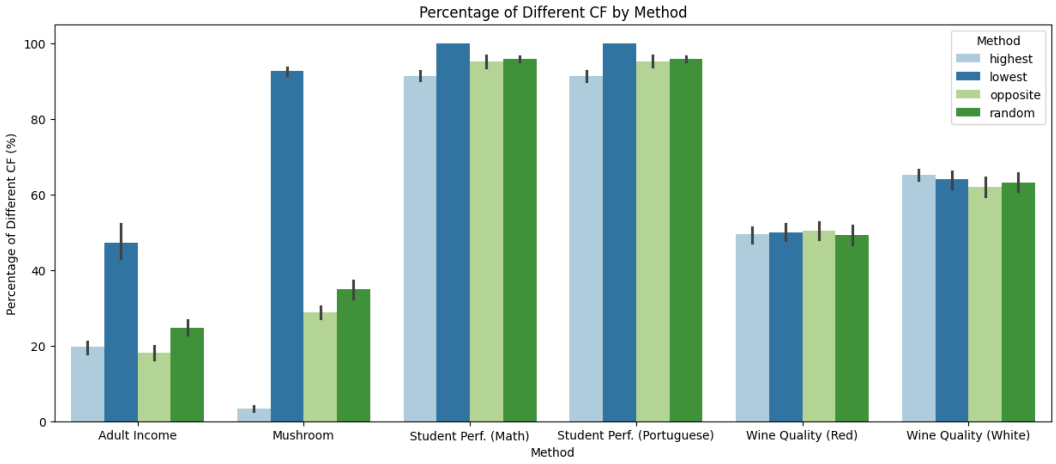

Figure 8. Percentage of partially or completely valid counterfactual for each dataset based on the various desired output selection strategies

In Figure 9, we see the average distance for each run of the experiment using the various strategies of selecting the desired output for each dataset. Here, the dark lines represents, for each desired output selection strategy, the mean distance, with its min-max range represented by the lighter shades of the same color. We can see the differences in the results when choosing the desired output via different strategies even within the same dataset. Similarly, in terms of distance from the original sample, there is no clear method of choosing desirable output that gives the least or highest distance across all datasets. Conversely, it means that counterfactuals depend on the context (the dataset, model, desired output). Similarly, we can see in Figure 10, the portion of features changed for each run using desired output computed from various different methods. Similar to Figure 9, the dark lines represents, for different desired output selection strategies, the mean portion of features changed, with its min-max range represented by the lighter shades of the same color. Consistent with mean distance, among the datasets, there is no clear desired output selection strategy that performs the best in terms of changing the fewest number of features possible. Similar to previous results, there are clear differences in the results between those strategies within each dataset.

 doi:10.20944/preprints202602.1316.v1

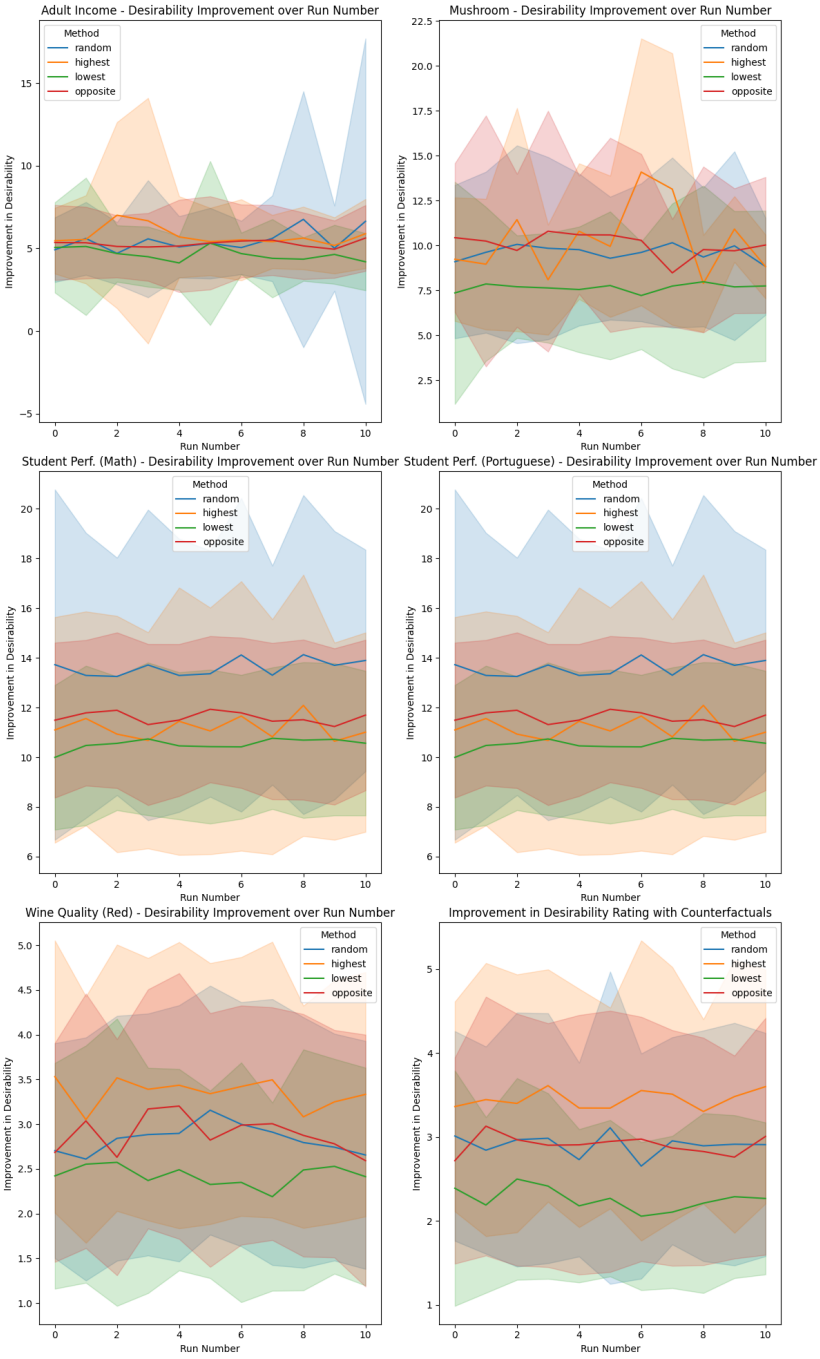

Figure 9. Average L2 distance of Counterfactuals from original instances for each run for each dataset

 doi:10.20944/preprints202602.1316.v1

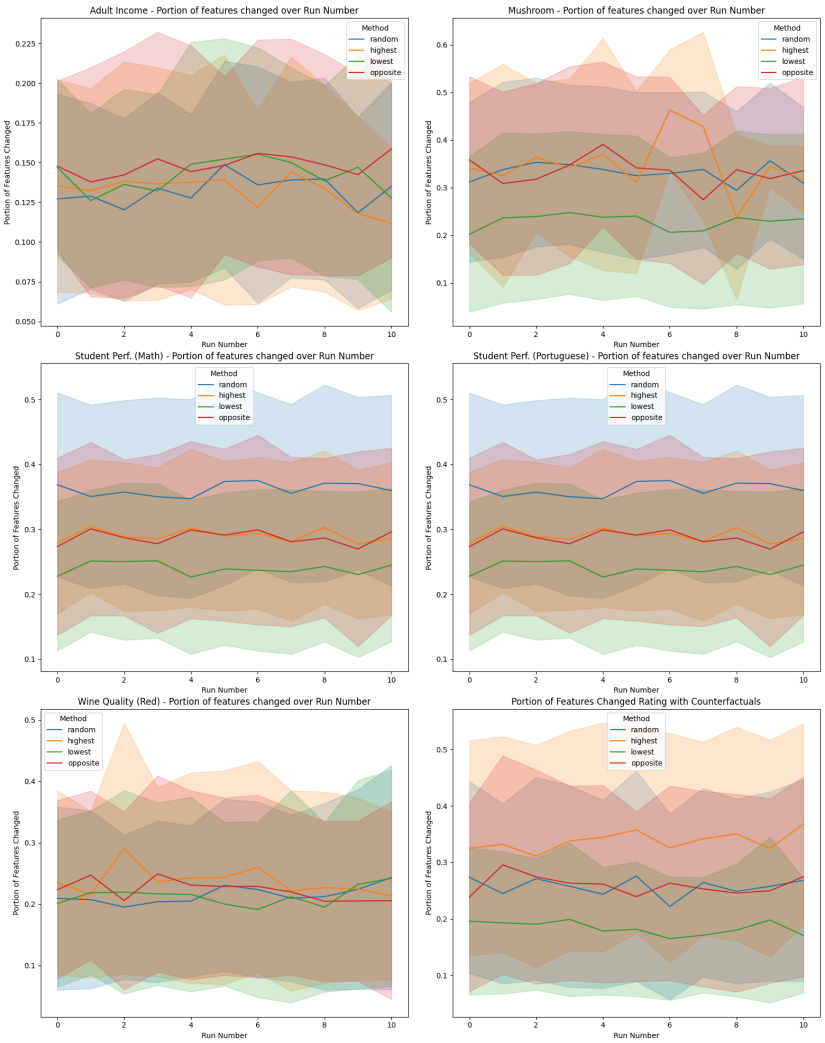

Figure 10. Average portion of features changed (to measure similarity) for each dataset for each run

7. Discussion of Results

This study demonstrates the feasibility of systematically generating valid counterfactual explanations for a variety of multi-dimensional classification problems, including single and multi-output classification tasks with both binary and multi-label outputs. The consistent ability to define and find counterfactuals across diverse datasets while being model-agnostic highlights the potential of this approach for eXplainable AI. The observed higher desirability ratings for counterfactual instances compared to original instances, as shown in Figures 6 and 7, highlights the ability to define and find valid counterfactuals using existing optimization methods. This is further supported by the percentage of partially or completely valid counterfactuals achieved, varying across datasets (Figure 8), and across different desired outputs demonstrating the method's capability of producing meaningful counterfactuals for explanation. We also measured *similarity* and *minimality*, as defined by Guidotti [5], assessed through distance and feature changes (Figures 9 and 10), which provides insight into how different the counterfactuals are from the original instances.

Our results emphasize the crucial role of defining the "desired output" in counterfactual generation, as this choice significantly impacts both validity, proximity, and similarity. Different strategies for determining this desired output led to varying results in terms of desirability rating, distance, and the number of features changed, even within the same dataset. This underlines the inherently case-dependent nature of counterfactual explanations, and highlights the ability to generate diverse set of counterfactuals based on the sample and desired output for that case. The lack of a universally "best" method for defining the desired output suggests that the optimal approach will be task-specific and

 doi:10.20944/preprints202602.1316.v1

potentially user-defined, especially in goal-achievement scenarios as discussed. As for model-analysis use, the desired output will be instance-specific and less user-defined.

While we explicitly evaluated validity and similarity, other important properties of counterfactual explanations were implicitly addressed or warrant further consideration. *Plausibility* is suggested by the validity scores, but a more rigorous evaluation would require analyzing counterfactuals against training dataset or even domain expert feedback on the generated counterfactuals. *Actionability* is also potentially present, as identifying the key features that, when changed, lead to a different outcome can inform interventions or decisions. However, we did not explicitly evaluate the feasibility or cost of enacting those changes in the real world. Our study did not assess the *discriminative power* of the counterfactuals, how well they distinguish the original instance from others with the same prediction. This is an important consideration as highly similar counterfactuals might not be very informative.

Finally, the observed differences in success rates across datasets suggest that the underlying data distribution and model complexity influence the effectiveness of counterfactual generation. Further investigation could explore how data characteristics affect the quality and interpretability of the resulting explanations. It is also worth noting that the concept of *diversity* is not explicitly addressed. Generating a single counterfactual might not fully capture the range of possible alternative outcomes. Future studies could explore methods for generating a diverse set of counterfactual explanations to provide a more comprehensive understanding of the model's behavior.

This work directly contributes to the growing field of eXplainable AI (XAI) by providing a robust and flexible framework, **DeRaC** for generating instance-level explanations. By systematically perturbing input features to achieve a desired outcome, we can probe model sensitivities and identify crucial feature interactions [37]. For example, analyzing which features consistently change across counterfactuals for a particular class reveals potential biases or unexpected dependencies within the model. Furthermore, the metrics we employ, validity, similarity, and minimality, offer quantifiable insights into the quality of these explanations, allowing for comparative analysis of different models or training procedures. This capability to assess explanation characteristics is essential for building trust and ensuring responsible AI development, enabling practitioners to not only understand what a model predicts, but also why, and how easily that prediction could change [14].

8. Conclusions and Future Works

The aim of this work is to formalize valid counterfactual for all classification problems including binary and multi-label classification for both single as well as multi-output problems. Rather than changing the existing definitions, the goal is to expand and codify them for problems that are not mostly discussed in the other works. There have been other works like [63] that mention multi-label problems, however they are using the existing notion of any different classification as valid counterfactual. As we discussed the problems associated with this approach in section 2, in case of multi-label and multi-output classification problems, simply "different" classification isn't always useful.

This problem was tackled by the introduction of the concept of desirability rating, which depends on pre-defined context-dependent desired outputs. These desired outputs are not dependent on the dataset or the model but rather on the instance or the current use-case (a concept that already exists in counterfactual explanation literature [5]). We then introduce the Desirability Rating based Counterfactual (**DeRaC**) framework, showing it is possible to mathematically measure how desirable the original and the counterfactual instances are and also to measure the differences between them. A valid counterfactual is defined as a counterfactual instance with desirability rating of 1 where as any improvement from the original instance in desirability makes it at least partially valid. It can be thought of as "on its way" to the desired output or "in-between" the original instance and the desired instance.

By expanding the definition of valid counterfactuals, we are still able to use concepts from earlier works like the properties of proximity, validity, similarity, actionability, etc. [4] and [5]. In the experi-

 doi:10.20944/preprints202602.1316.v1

ments, we use an existing optimization algorithm, Powell's Method [61], to optimize for desirability rating (an expansion to existing validity property), and are able to generate counterfactuals based on different desired outputs for all datasets. These were tested on single output binary classification, multi-label as well as multi-output classification problems. The experiment was conducted by sampling separated testing dataset with multiple runs for each method of selecting the desired output. These go to show the reproducibility as well as the compatibility with various different desired outputs. This is particularly important given the fact that multi-dimensional classification problems have more than one possible desired output unlike single-output binary classification problems.

It is also important to recognize that there is still further work to be done. Testing on more datasets with combination of multiple outputs and labels, should give more weight to the proposed formalization. In the next iterations, we can also test existing counterfactual generation methods like [30,31,36] that account for properties like proximity, validity, similarity, actionability, etc. on multi-label and multi-output problems. It is also possible to expand this formalization for regression problems and study them for efficiency and computational cost. Finally, user studies are needed to assess the human interpretability and trustworthiness of these counterfactual explanations, ultimately determining their value in real-world applications like decision support, model debugging, and fairness auditing. This research lays the groundwork for building more transparent, understandable, and trustworthy machine learning systems.

Author Contributions: Conceptualization, Neelabh Kshetry and Mehmed Kantardzic; Methodology, Neelabh Kshetry; Validation, Neelabh Kshetry; Writing – original draft, Neelabh Kshetry; Writing – review and editing, Neelabh Kshetry and Mehmed Kantardzic; Supervision, Mehmed Kantardzic

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Cao, L. Ai in finance: challenges, techniques, and opportunities. *ACM Computing Surveys (CSUR)* **2022**, *55*, 1–38.
2. Goswami, G.; Bhardwaj, R.; Singh, R.; Vatsa, M. MDLFace: Memorability augmented deep learning for video face recognition. In Proceedings of the IEEE international joint conference on biometrics. IEEE, 2014, pp. 1–7.
3. Sarker, I.H.; Furhad, M.H.; Nowrozy, R. Ai-driven cybersecurity: an overview, security intelligence modeling and research directions. *SN Computer Science* **2021**, *2*, 173.
4. Verma, S.; Boonsanong, V.; Hoang, M.; Hines, K.E.; Dickerson, J.P.; Shah, C. Counterfactual Explanations and Algorithmic Recourses for Machine Learning: A Review, 2022. arXiv:2010.10596 [cs], https://doi.org/10.48550/arXiv.2010.10596.
5. Guidotti, R. Counterfactual explanations and how to find them: literature review and benchmarking. *Data Mining and Knowledge Discovery* **2024**, *38*, 2770–2824. https://doi.org/10.1007/s10618-022-00831-6.
6. Fisher, R.A. Iris. UCI Machine Learning Repository, 1936. DOI: https://doi.org/10.24432/C56C76.
7. Cortez, Paulo, C.A.A.F.M.T.; Reis, J. Wine Quality. UCI Machine Learning Repository, 2009. DOI: https://doi.org/10.24432/C56S3T.
8. Marcinkevičs, R.; Vogt, J.E. Interpretable and explainable machine learning: A methods-centric overview with concrete examples. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* **2023**, *13*, e1493.
9. Ali, S.; Abuhmed, T.; El-Sappagh, S.; Muhammad, K.; Alonso-Moral, J.M.; Confalonieri, R.; Guidotti, R.; Del Ser, J.; Díaz-Rodríguez, N.; Herrera, F. Explainable Artificial Intelligence (XAI): What we know and what is left to attain Trustworthy Artificial Intelligence. *Information Fusion* **2023**, *99*, 101805. https://doi.org/10.1016/j.inffus.2023.101805.
10. Džeroski, S.; Demšar, D.; Grbović, J. Predicting Chemical Parameters of River Water Quality from Bioindicator Data. *Applied Intelligence* **2000**, *13*, 7–17. https://doi.org/10.1023/A:1008323212047.
11. Li, H.; Zhang, W.; Chen, Y.; Guo, Y.; Li, G.Z.; Zhu, X. A novel multi-target regression framework for time-series prediction of drug efficacy. *Scientific Reports* **2017**, *7*, 40652. Publisher: Nature Publishing Group, https://doi.org/10.1038/srep40652.

 doi:10.20944/preprints202602.1316.v1

12. Borchani, H.; Varando, G.; Bielza, C.; Larrañaga, P. A survey on multi-output regression. *WIREs Data Mining and Knowledge Discovery* **2015**, *5*, 216–233. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/widm.1157, https://doi.org/10.1002/widm.1157.
13. Xu, D.; Shi, Y.; Tsang, I.W.; Ong, Y.S.; Gong, C.; Shen, X. Survey on Multi-Output Learning. *IEEE Transactions on Neural Networks and Learning Systems* **2020**, *31*, 2409–2429. https://doi.org/10.1109/TNNLS.2019.2945133.
14. Wachter, S.; Mittelstadt, B.; Russell, C. Counterfactual Explanations without Opening the Black Box: Automated Decisions and the GDPR. *Harvard Journal of Law & Technology (Harvard JOLT)* **2017**, *31*, 841–888.
15. Bodria, F.; Giannotti, F.; Guidotti, R.; Naretto, F.; Pedreschi, D.; Rinzivillo, S. Benchmarking and survey of explanation methods for black box models. *Data Mining and Knowledge Discovery* **2023**, *37*, 1719–1778.
16. Stepin, I.; Alonso, J.M.; Catala, A.; Pereira-Fariña, M. A survey of contrastive and counterfactual explanation generation methods for explainable artificial intelligence. *Ieee Access* **2021**, *9*, 11974–12001.
17. Keane, M.T.; Kenny, E.M.; Delaney, E.; Smyth, B. If Only We Had Better Counterfactual Explanations: Five Key Deficits to Rectify in the Evaluation of Counterfactual XAI Techniques. In Proceedings of the Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, Montreal, Canada, 2021; pp. 4466–4474. https://doi.org/10.24963/ijcai.2021/609.
18. Laugel, T.; Jeyasothy, A.; Lesot, M.J.; Marsala, C.; Detyniecki, M. Achieving diversity in counterfactual explanations: a review and discussion. In Proceedings of the Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency, 2023, pp. 1859–1869.
19. Jiang, J.; Leofante, F.; Rago, A.; Toni, F. Robust counterfactual explanations in machine learning: A survey. *arXiv preprint arXiv:2402.01928* **2024**.
20. Mertes, S.; Huber, T.; Weitz, K.; Heimerl, A.; André, E. Ganterfactual—counterfactual explanations for medical non-experts using generative adversarial learning. *Frontiers in artificial intelligence* **2022**, *5*, 825565.
21. Del Ser, J.; Barredo-Arrieta, A.; Díaz-Rodríguez, N.; Herrera, F.; Saranti, A.; Holzinger, A. On generating trustworthy counterfactual explanations. *Information Sciences* **2024**, *655*, 119898.
22. Pawelczyk, M.; Agarwal, C.; Joshi, S.; Upadhyay, S.; Lakkaraju, H. Exploring counterfactual explanations through the lens of adversarial examples: A theoretical and empirical analysis. In Proceedings of the International Conference on Artificial Intelligence and Statistics. PMLR, 2022, pp. 4574–4594.
23. Maaroof, N.; Moreno, A.; Valls, A.; Jabreel, M.; Romero-Aroca, P. Multi-Class Fuzzy-LORE: A Method for Extracting Local and Counterfactual Explanations Using Fuzzy Decision Trees. *Electronics* **2023**, *12*, 2215.
24. Dastile, X.; Celik, T. Counterfactual explanations with multiple properties in credit scoring. *IEEE Access* **2024**.
25. Prado-Romero, M.A.; Prenkaj, B.; Stilo, G.; Celi, A.; Estevanell-Valladares, E.L.; Pérez, D.A.V. Ensemble Approaches for Graph Counterfactual Explanations. In Proceedings of the XAI. it@ AI* IA, 2022, pp. 88–97.
26. De Toni, G.; Lepri, B.; Passerini, A. Synthesizing explainable counterfactual policies for algorithmic recourse with program synthesis. *Machine Learning* **2023**, *112*, 1389–1409.
27. Kanamori, K.; Takagi, T.; Kobayashi, K.; Ike, Y.; Uemura, K.; Arimura, H. Ordered counterfactual explanation by mixed-integer linear optimization. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, 2021, Vol. 35, pp. 11564–11574.
28. Maiti, A.; Plecko, D.; Bareinboim, E. Counterfactual Identification Under Monotonicity Constraints. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, 2025, Vol. 39, pp. 26841–26850.
29. Zhou, G.; Yao, L.; Xu, X.; Wang, C.; Zhu, L. Learning to infer counterfactuals: meta-learning for estimating multiple imbalanced treatment effects. *arXiv preprint arXiv:2208.06748* **2022**.
30. Cheng, F.; Ming, Y.; Qu, H. DECE: Decision Explorer with Counterfactual Explanations for Machine Learning Models. *IEEE Transactions on Visualization and Computer Graphics* **2021**, *27*, 1438–1447. https://doi.org/10.1109/TVCG.2020.3030342.
31. Wexler, J.; Pushkarna, M.; Bolukbasi, T.; Wattenberg, M.; Viegas, F.; Wilson, J. The What-If Tool: Interactive Probing of Machine Learning Models. *IEEE Transactions on Visualization and Computer Graphics* **2019**, pp. 1–1. arXiv:1907.04135 [cs], https://doi.org/10.1109/TVCG.2019.2934619.
32. Tenney, I.; Wexler, J.; Bastings, J.; Bolukbasi, T.; Coenen, A.; Gehrmann, S.; Jiang, E.; Pushkarna, M.; Radebaugh, C.; Reif, E.; et al. The language interpretability tool: Extensible, interactive visualizations and analysis for NLP models. *arXiv preprint arXiv:2008.05122* **2020**.
33. Krause, J.; Perer, A.; Ng, K. Interacting with Predictions: Visual Inspection of Black-box Machine Learning Models. In Proceedings of the Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, San Jose California USA, 2016; pp. 5686–5697. https://doi.org/10.1145/2858036.2858529.

 doi:10.20944/preprints202602.1316.v1

34. Ming, Y.; Qu, H.; Bertini, E. RuleMatrix: Visualizing and Understanding Classifiers with Rules. *IEEE Transactions on Visualization and Computer Graphics* **2019**, *25*, 342–352. https://doi.org/10.1109/TVCG.2018.2864812.
35. Gathani, S.; Hulsebos, M.; Gale, J.; Haas, P.J.; Demiralp, Ç. Augmenting decision making via interactive what-if analysis. *arXiv preprint arXiv:2109.06160* **2021**.
36. Gomez, O.; Holter, S.; Yuan, J.; Bertini, E. ViCE: visual counterfactual explanations for machine learning models. In Proceedings of the Proceedings of the 25th International Conference on Intelligent User Interfaces, Cagliari Italy, 2020; pp. 531–535. https://doi.org/10.1145/3377325.3377536.
37. Kshetry, N.; Kantardzic, M. What-if XAI framework (WiXAI): from counterfactuals towards causal understanding. *Journal of Computer and Communications* **2024**, *12*, 169–198.
38. Goyal, Y.; Wu, Z.; Ernst, J.; Batra, D.; Parikh, D.; Lee, S. Counterfactual visual explanations. In Proceedings of the International Conference on Machine Learning. PMLR, 2019, pp. 2376–2384.
39. Kenny, E.M.; Delaney, E.D.; Greene, D.; Keane, M.T. Post-hoc explanation options for xai in deep learning: The insight centre for data analytics perspective. In Proceedings of the International Conference on Pattern Recognition. Springer, 2021, pp. 20–34.
40. Akula, A.R.; Wang, K.; Liu, C.; Saba-Sadiya, S.; Lu, H.; Todorovic, S.; Chai, J.; Zhu, S.C. CX-ToM: Counterfactual explanations with theory-of-mind for enhancing human trust in image recognition models. *Iscience* **2022**, *25*.
41. Kenny, E.M.; Keane, M.T. On generating plausible counterfactual and semi-factual explanations for deep learning. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, 2021, Vol. 35, pp. 11575–11585.
42. Thiagarajan, J.J.; Thopalli, K.; Rajan, D.; Turaga, P. Training calibration-based counterfactual explainers for deep learning models in medical image analysis. *Scientific reports* **2022**, *12*, 597.
43. Alipour, K.; Lahiri, A.; Adeli, E.; Salimi, B.; Pazzani, M. Explaining image classifiers using contrastive counterfactuals in generative latent spaces. *arXiv preprint arXiv:2206.05257* **2022**.
44. Zhao, W.; Oyama, S.; Kurihara, M. Generating natural counterfactual visual explanations. In Proceedings of the Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence, 2021, pp. 5204–5205.
45. Robeer, M.; Bex, F.; Feelders, A. Generating realistic natural language counterfactuals. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP 2021). Association for Computational Linguistics, 2021, pp. 3611–3625.
46. Arie, A.B.; Deutch, D.; Frost, N.; Horesh, Y.; Meyuhas, I. Optimizing Counterfactual-based Analysis of Machine Learning Models Through Databases. In Proceedings of the 27th International Conference on Extending Database Technology, EDBT 2024. OpenProceedings. org, 2024, pp. 597–609.
47. Piccialli, V.; Morales, D.R.; Salvatore, C. Supervised feature compression based on counterfactual analysis. *European Journal of Operational Research* **2024**, *317*, 273–285.
48. Gao, R.; Biggs, M.; Sun, W.; Han, L. Enhancing counterfactual classification via self-training. *arXiv preprint arXiv:2112.04461* **2021**.
49. Temraz, M.; Keane, M.T. Solving the class imbalance problem using a counterfactual method for data augmentation. *Machine Learning with Applications* **2022**, *9*, 100375.
50. Sohns, J.T.; Garth, C.; Leitte, H. Decision boundary visualization for counterfactual reasoning. In Proceedings of the Computer Graphics Forum. Wiley Online Library, 2023, Vol. 42, pp. 7–20.
51. Mahajan, D.; Tan, C.; Sharma, A. Preserving causal constraints in counterfactual explanations for machine learning classifiers. *arXiv preprint arXiv:1912.03277* **2019**.
52. Duong, T.D.; Li, Q.; Xu, G. Causality-based counterfactual explanation for classification models. *Knowledge-Based Systems* **2024**, *300*, 112200.
53. Xia, K.; Pan, Y.; Bareinboim, E. Neural causal models for counterfactual identification and estimation. *arXiv preprint arXiv:2210.00035* **2022**.
54. Crupi, R.; González, B.S.M.; Castelnovo, A.; Regoli, D. Leveraging Causal Relations to Provide Counterfactual Explanations and Feasible Recommendations to End Users. In Proceedings of the ICAART (2), 2022, pp. 24–32.
55. Carlevaro, A.; Lenatti, M.; Paglialonga, A.; Mongelli, M. Multi-Class Counterfactual Explanations using Support Vector Data Description.
56. Caron, A.; Baio, G.; Manolopoulou, I. Counterfactual Learning with Multioutput Deep Kernels. *arXiv preprint arXiv:2211.11119* **2022**.

 doi:10.20944/preprints202602.1316.v1

57. Parbhoo, S.; Bauer, S.; Schwab, P. Ncore: Neural counterfactual representation learning for combinations of treatments. *arXiv preprint arXiv:2103.11175* **2021**.
58. Mushroom. UCI Machine Learning Repository, 1981. DOI: https://doi.org/10.24432/C5959T.
59. Cortez, P. Student Performance. UCI Machine Learning Repository, 2008. DOI: https://doi.org/10.24432/C5TG7T.
60. Becker, B.; Kohavi, R. Adult. UCI Machine Learning Repository, 1996. DOI: https://doi.org/10.24432/C5XW20.
61. Powell, M.J.D. An efficient method for finding the minimum of a function of several variables without calculating derivatives. *The Computer Journal* **1964**, *7*, 155–162, [https://academic.oup.com/comjnl/article-pdf/7/2/155/959784/070155.pdf]. https://doi.org/10.1093/comjnl/7.2.155.
62. Virtanen, P.; Gommers, R.; Oliphant, T.E.; Haberland, M.; Reddy, T.; Cournapeau, D.; Burovski, E.; Peterson, P.; Weckesser, W.; Bright, J.; et al. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods* **2020**, *17*, 261–272. https://doi.org/10.1038/s41592-019-0686-2.
63. Carlevaro, A.; Lenatti, M.; Paglialonga, A.; Mongelli, M. Multiclass Counterfactual Explanations Using Support Vector Data Description. *IEEE Transactions on Artificial Intelligence* **2023**, *5*, 3046–3056.