Counterfactual Shapley Additive Explanations

Emanuele Albini
J.P. Morgan AI Research
London, UK
emanuele.albini@jpmorgan.com

Jason Long
J.P. Morgan AI Research
London, UK
jason.x.long@jpmorgan.com

Danial Dervovic
J.P. Morgan AI Research
London, UK
danial.dervovic@jpmorgan.com

Daniele Magazzeni
J.P. Morgan AI Research
London, UK
daniele.magazzeni@jpmorgan.com

ABSTRACT

Feature attributions are a common paradigm for model explanations due to their simplicity in assigning a single numeric score for each input feature to a model. In the actionable recourse setting, wherein the goal of the explanations is to improve outcomes for model consumers, it is often unclear how feature attributions should be correctly used. With this work, we aim to strengthen and clarify the link between actionable recourse and feature attributions. Concretely, we propose a variant of SHAP, *Counterfactual SHAP* (CF-SHAP), that incorporates counterfactual information to produce a *background dataset* for use within the marginal (a.k.a. interventional) Shapley value framework. We motivate the need within the actionable recourse setting for careful consideration of background datasets when using Shapley values for feature attributions with numerous synthetic examples. Moreover, we demonstrate the efficacy of CF-SHAP by proposing and justifying a quantitative score for feature attributions, *counterfactual-ability*, showing that as measured by this metric, CF-SHAP is superior to existing methods when evaluated on public datasets using tree ensembles.

CCS CONCEPTS

• **Computing methodologies** → **Machine learning**; **Artificial intelligence**; *Classification and regression trees*; *Supervised learning by classification*; • **Human-centered computing** → **Human computer interaction (HCI)**.

KEYWORDS

XAI, SHAP, actionable recourse, counterfactual explanations, feature attributions, feature importance, Shapley values, explainability

ACM Reference Format:
Emanuele Albini, Jason Long, Danial Dervovic, and Daniele Magazzeni. 2022. Counterfactual Shapley Additive Explanations. In *2022 ACM Conference on Fairness, Accountability, and Transparency (FAccT '22), June 21–24, 2022, Seoul, Republic of Korea.* ACM, New York, NY, USA, 17 pages. https://doi.org/10.1145/3531146.3533168

FAccT '22, June 21–24, 2022, Seoul, Republic of Korea

ACM ISBN 978-1-4503-9352-2/22/06.
https://doi.org/10.1145/3531146.3533168

1 INTRODUCTION

Government regulators are placing increasing emphasis on the fairness and discrimination issues in decision making processes using machine learning algorithms in high-stakes context as finance and healthcare. For example, the European Commission [14] put particular emphasis on the right to explain AI systems decisions while the U.S. credit regulations [63] are even more specific as they prescribe that automatic decisions must be explained in terms of key factors that contributed to an adverse decision. At the same time, in the academic literature several techniques have been proposed to address this issue (see [2, 6, 12, 20] for an overview).

In the context of *local explainability* many approaches on which researchers have focused in the last years are based on the notion of *feature attribution*, i.e., distributing the output of the model for a specific input to its features (e.g., [38, 40, 51]). In this paper in particular we will focus on SHAP, one of the most popular techniques to generate local explanations based on the notion of Shapley value [54] from game theory. Shapley value-based frameworks for Explainable AI (XAI) consider each feature as a player in a m-person game to fairly distribute the contribution of each feature to the output of the model. To do so they compare the output of the (same) model when a feature is present with that of when the same feature is missing. There are two main limitations with this approach that have been raised in the literature:

(a) It is not clear how to define the output of the model when a feature is missing. The most common approach is to estimate it as an expectation over a background distribution of the input features [39].
(b) There is no explicit guidance provided on how a user might alter one's behavior in a desirable way [33].

Another popular area of research has developed around *counterfactual explanations*, also known as *algorithmic recourse*, i.e., given a specific input one must find the "closest possible world (input)" [67] that gives rise to a different outcome. In practice, this means that these approaches aim to find (one or more) points that are (1) close to the one we want to explain; and (2) "plausible" (where plausibility can be defined in different ways in the literature, see [32] for more insights). Counterfactual explanations have two main limitations:

(a) Most of the approaches in the literature are limited at finding a single counterfactual point. While this may give the user a clear understanding of what they could do in order to

reverse an adverse outcome, it does not allow them to choose changes that are more suited for them.

(b) While there has been some attempt at generating diverse sets of counterfactuals (e.g., [43, 53]), there is no consensus on how to limit the cognitive load for the user caused by the sheer amount of information that is provided, or – in other words – on how to provide a more amenable explanation (in terms of size), as advocated also from a social science perspective [41].

In this paper we present how these two general approaches for explainability can be combined in order to provide a *counterfactual feature attribution* grounded on the game-theoretic approach afforded by Shapley values that we call *Counterfactual SHAP (CF-SHAP)*. We are motivated by the desire to retain the simple form of explanation provided by feature attributions, while introducing the actionability properties of counterfactual explanations.

In particular, our contributions are as follows.

- We enumerate the assumptions that are necessary to interpret Shapley values in a counterfactual sense and discuss what it means for a feature attribution method to demonstrate counterfactual behaviour.
- We introduce a general framework to measure the *counterfactual-ability* of a feature attribution as a way to quantitatively evaluate its ability to suggest to the user how to act upon the input in order to overcome an adverse prediction.
- In order to achieve higher counterfactual-ability, we propose to (1) use (a uniform distribution over) a set of counterfactuals as the background distribution for the computation of Shapley values and (2) to enrich the explanation with guidance on the direction in which to change the features, yielding the CF-SHAP algorithm.
- We benchmark CF-SHAP against baseline feature attribution techniques. CF-SHAP, using 100-nearest neighbours as a simple counterfactual generation technique, is shown to have the best counterfactual-ability on 3 publicly available datasets.

We note that in this paper we concentrate on tree-based models for the following reasons: (1) in the context of classification and regression for tabular data, tree-based ensemble models as XGBoost, CatBoost, LightGBM and Random Forest are deemed as the state-of-the-art in terms of performance [56] and therefore are widely adopted in many industries including finance [61]; (2) interventional Shapley values can be computed exactly and efficiently for tree-based models using the algorithm proposed in [37].

2 BACKGROUND

In the remainder of this paper we consider a trained binary classification *model* $f : \mathcal{X} \to \mathcal{Y}$ where $\mathcal{X} = \mathbb{R}^m$ and $\mathcal{Y} = \mathbb{R}$. We define the *decision function* $F : \mathcal{X} \to \{0, 1\}$ with (binary) *decision threshold* $t \in \mathbb{R}$ as[1]

$$F(\boldsymbol{x}) = \begin{cases} 1 & \text{if } f(\boldsymbol{x}) > t \\ 0 & \text{otherwise} \end{cases}.$$

[1]We use lower-case bold symbols to indicate vectors (e.g., $\boldsymbol{x}$) and non-bold symbols to indicate scalars (e.g., x_i).

We refer to $f(\boldsymbol{x})$ as the model *output* and to $F(\boldsymbol{x})$ as the model *prediction* or *outcome*.

Note that, without loss of generality, if not otherwise specified we use $t = 0$ as decision threshold. Moreover, without loss of generality, we assume that an input $\boldsymbol{x} \in \mathcal{X}$ such that $F(\boldsymbol{x}) = 1$ is an *adverse outcome* for the user, e.g., the rejection of a loan application. We also note that the results in this paper can be trivially generalized to multi-class models.

2.1 Shapley values

The Shapley values method is a technique used in classic game theory to fairly attribute the payoff to the players in an m-player cooperative game. Given a set of players $\mathcal{F} = \{1, \dots, m\}$ and the *characteristic function* $v : 2^{\mathcal{F}} \to \mathbb{R}$ of a game Γ Shapley values fairly attribute the payoff returned by the characteristics function to each player.

In the context of machine learning models the players are the features of the model and several ways have been proposed to simulate feature absence in the characteristic function (e.g., retraining the model without such feature [60]). In this paper we use the approximation of the characteristic function proposed in [38] and [37] (SHAP) that simulates the absence of a feature using the marginal expectation over a background distribution $\mathcal{D}$.

Formally, the Shapley value of player i is defined as:

$$\phi_i = \frac{1}{m} \sum_{S \subseteq \mathcal{F} \setminus \{i\}} \binom{m-1}{|S|}^{-1} [v(S \cup \{i\}) - v(S)]$$

$$\text{where } v(S) = \mathbb{E}_{\boldsymbol{x}' \sim \mathcal{D}} \left[f(\boldsymbol{x}_S, \boldsymbol{x}'_{\mathcal{F} \setminus S}) \right]$$

where $\mathbb{E}_{\boldsymbol{x}' \sim \mathcal{D}}$ denotes the expected value under the distribution $\mathcal{D}$ and with an abuse of notation $f(\boldsymbol{x}_S, \boldsymbol{x}'_{\mathcal{F} \setminus S})$ indicates the output of the model with feature values $\boldsymbol{x}$ for features in S and values $\boldsymbol{x}'$ for feature values not in S.

We will henceforth refer to the space of Shapley values $\mathbb{R}^m$ as Φ and to the Shapley values vector of $\boldsymbol{x}$ as $\boldsymbol{\phi}$.

2.2 Counterfactual Explanations

In its basic form, a (local) counterfactual explanation (CF) for an input $\boldsymbol{x}$ is a point $\boldsymbol{x}'$ such that (1) $\boldsymbol{x}'$ gives rise to a different prediction, i.e., $F(\boldsymbol{x}) \neq F(\boldsymbol{x}')$, (2) $\boldsymbol{x}$ and $\boldsymbol{x}'$ are close (under some distance metric) and (3) $\boldsymbol{x}'$ is a "plausible" input. This last constraint has been interpreted in several ways in the literature, it may involve considerations about sparsity (e.g., [57]), proximity to the data manifold (e.g., [45]), proximity to other counterfactuals (e.g., [44]), causality (e.g., [31]), actionability (e.g., [47, 64]) or a combination thereof (e.g., [13]). A plethora of techniques for the generation of counterfactuals exist in the literature using search algorithms (e.g., [3, 4, 58, 67]), optimization (e.g., [28]) and genetic algorithms (e.g., [55]) among other methods (we refer the reader to [29, 32, 59, 65] for recent surveys).

In the scope of this paper we need to consider only counterfactual explanation methods that are (1) able to generate a set of (multiple) counterfactuals and (2) do not require the model to be differentiable since we focus on tree-based models. We note that few counterfactual explanation techniques satisfying both of these requirements exist in the literature

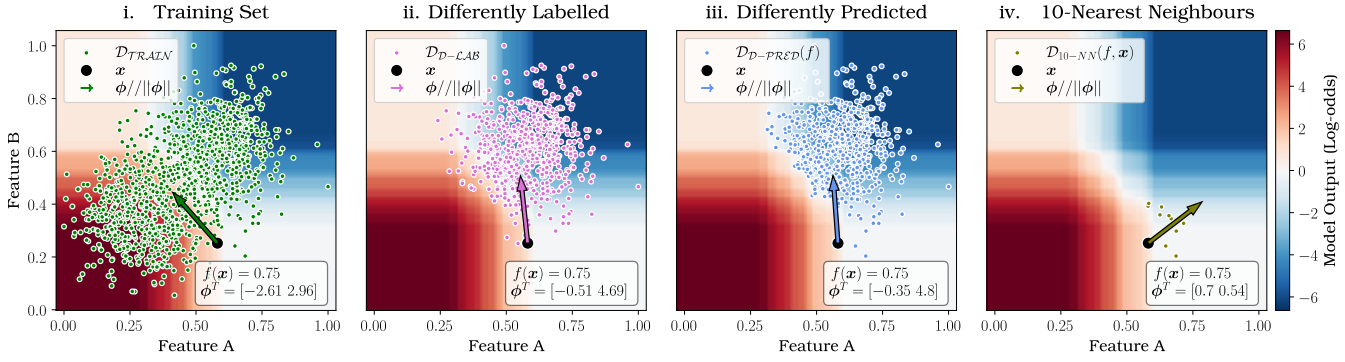

Figure 1: Effect of different choices of background dataset on the Shapley values (ϕ) of the same input (x) with the same model. Red regions correspond to areas of the feature space where the decision is adverse, i.e. $F(\boldsymbol{x}) = 1$, with blue regions representing the opposite, i.e. those $\boldsymbol{x} \in \mathcal{X}$ for which $F(\boldsymbol{x}) = 0$. Coloured arrows and scatter points represent the directions of the Shapley values vector and the background datasets used for their computation, respectively. We note how input-invariant distributions (i, ii, iii) do not give rise to SHAP values providing actionable guidance to overcome the adverse outcome; when using instead a set of counterfactuals (iv) a more actionable explanation is obtained (see Section 3.1 for more details).

3 COUNTERFACTUAL SHAP

In general, Shapley values do not have an obvious interpretation in counterfactual terms, this means that they do not provide suggestions on how a user can change their features in order to change the prediction [5, 33]. We argue that this is due to 2 main reasons: (1) the "arbitrary" choice of background distribution for the computation of Shapley values and (2) the lack of guidance on the best direction of the change for each of the features. We now discuss in details this two aspects.

3.1 Choice of the background distribution

Shapley values describe the contributions of the players (features) to the game payoff (model output). In the context of machine learning model explainability an important assumption is made: the simulation of each feature's absence in the cooperative game using a background distribution $\mathcal{D}$. As pointed out in [39], this means that Shapley values explain a prediction of an input ***in contrast*** *to a distribution of background points*. In practice, the background distribution is taken as a uniform distribution over unit point masses at a finite number of points, called the *background dataset*.

Therefore, the background dataset should be chosen according to the contrastive question that one aim to answer. We list some of the most common distributions that have been proposed.[2]

Training set $\mathcal{D}_{\text{TRAIN}}$ [22, 23, 38]. The training set, including the samples that are labelled and/or predicted of being of the same class of the input.

Differently-labelled samples $\mathcal{D}_{\text{D-LAB}}$ [24, 25]. The samples in the training set *labelled* differently than the input.

Differently-predicted samples $\mathcal{D}_{\text{D-PRED}}(f)$ [24, 25]. The samples in the training set *predicted* with a different class.

These choices of background dataset have in common the fact that they are defined a priori, i.e., given a model, they are equal for all the inputs. This means that we are contrasting an input $\boldsymbol{x}$ with a (input-invariant) distribution $\mathcal{D}$ that may potentially be very different from $\boldsymbol{x}$. This can give rise to explanations that are sometimes misleading for a user who is typically interested in understanding which features led to their adverse outcome (in order to reverse it) [41]. In other words the constrastive question that we are answering with the Shapley values is not tailored to the specific input (user) and therefore instead of answering the question of "Why was a user rejected when compared to similar users that were accepted?" we will be answering the more generic question of "Which features are most important in making my outcome different from that of other (accepted) users?" (who are potentially very different from $\boldsymbol{x}$).

The example in Figure 1.i shows an explanation where the background dataset is the training set, and we note that the Shapley values suggest that Feature A negatively contributed to the model output; this means that the current value of Feature A is "protective" against rejection when *put in contrast* with the expected output of the model obtained when using the background distribution $\mathcal{D}_{TRAIN}$. This may be useful information for the model developers **but it does not allow one to gain any (actionable) insight** unless we assume access to the underlying distribution $\mathcal{D}_{TRAIN}$. In fact, this explanation only informs the user that their Feature A value has a positive impact when contrasted to typical Feature A values, but it does not either (a) advise them on how they can change their features in order to overcome the (adverse) outcome; or (b) inform them which features were most important in rejecting their application.

Figures 1.ii and 1.iii show how alternative (but still input-invariant) background distributions ($\mathcal{D}_{D-LAB}$ and $\mathcal{D}_{D-PRED}$, resp.) may improve the explanations in terms of informing the user on which features were most important in rejecting their application when compared to other rejected samples, but they still lack the ability of giving useful insight on which features were the most important and therefore should be acted upon in order to reverse the adverse decision $F(\boldsymbol{x}) = 1$. This is due to the contrastive question being

[2] We note that these distributions are often too large to use in practice, and instead the background dataset is obtained by sub-sampling or using k-means to generate a number of medoids [22, 23, 38].

posed with respect to (a) samples that have much better (lower) model outputs and (b) samples that have similar model output but that are very different (in terms of distance in input space) from $\boldsymbol{x}$.

Using a set of counterfactual points as the background dataset mitigates the issues mentioned in the preceding example. In particular, using counterfactuals as the background dataset allows one to answer a contrastive question that is (a) of interest for the user because it is comparing $\boldsymbol{x}$ to samples that are similar to them (and implicitly more "reachable") and (b) more amenable in terms of access to the underlying distributions. In fact, as mentioned earlier, a useful interpretation of Shapley values-based explanations requires access to the background distribution. Arguably, a user can relate to a set of similar customers more easily than the training set (that may contain very different users). For example, it would be most useful in a credit lending context to compare customers with new accounts and no mortgage to other customers in a similar financial situation, while contrasting them to average customers, including many home-owners with older accounts, may lead to less actionable explanations.

We now formally define the Counterfactual SHAP values.

Definition 3.1. The *Counterfactual SHAP values* for an input $\boldsymbol{x} \in \mathcal{X}$ are the Shapley values for $\boldsymbol{x}$ computed using the following characteristic function.

$$v(S) = \mathbb{E}_{\boldsymbol{x'} \sim \mathcal{D}_C(f,\boldsymbol{x})} \left[f(\boldsymbol{x}_S, \boldsymbol{x'}_{\mathcal{F}\setminus S}) \right]$$

where C is (the name of) the counterfactual technique used to obtain the distribution $\mathcal{D}_C(f, \boldsymbol{x})$.

We note that, using a set of counterfactual points as a background distribution (contrary to "classic" input-invariant background distributions), means that the background distribution depends on the input $\boldsymbol{x}$ as reflected by Definition 3.1. We can appreciate the effect of using counterfactuals as background dataset for the running example in Figure 1.iv. We note how both features are deemed as contributing to the rejection when compared to similar customers that were instead accepted. And in fact, Feature A has a higher importance than Feature B since the model is locally more sensitive to Feature A than Feature B as shown by the sharper color gradient in the horizontal direction.

3.2 Guidance on the direction of the change

As remarked in [5], feature attributions do not clearly provide guidance on how to alter the features in order to change the prediction of a model. In the case of Shapley values, this is due to Shapley values providing information about the most important features solely in terms of model output. In practice this means that the Shapley value ϕ_i is not useful for understanding how one should alter the feature i (i.e., increase or decrease it) in order to change the adverse outcome. In other words, Shapley values identify features that have a strong impact on the output, **but not which inputs values are associated to output values of interest**. They are therefore not useful in informing a user on what intervention on x_i is necessary to decrease $f(\boldsymbol{x})$, and consequently overcome the adverse outcome.

To address this issue we enrich the explanation provided by Counterfactual SHAP values with the *derived trends*.

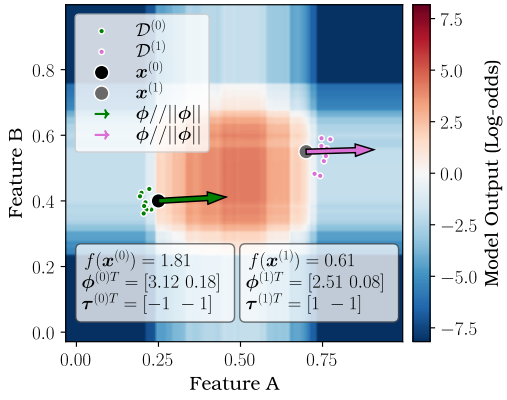

Figure 2: Effect of (the lack of) derived trends on the Shapley values of two same inputs using the 10-NN as background dataset. Coloured arrows and scatter points represent the directions of the Shapley values vector and the background dataset used for their computation, respectively. We note that the derived trends better provide guidance for the direction of change of the features that cannot be afforded by the Shapley values alone (see Section 3.2 for more details).

Definition 3.2. The *derived trends* $\boldsymbol{\tau} \in \mathrm{T}$ for an input $\boldsymbol{x} \in \mathcal{X}$ with respect to a distribution $\mathcal{D}_C(f, \boldsymbol{x})$ are such that:

$$\tau_i = sgn\left(\mathbb{E}_{\boldsymbol{x'} \in \mathcal{D}_C(f,\boldsymbol{x})} \left[x'_i \right] - x_i \right)$$

where *sgn* denotes the sign function[3] and T denotes the set of all derived trends (vectors), i.e., $\mathrm{T} = \{-1, +1, 0\}^m$.

The derived trend τ_i for feature i will be $+1$ (-1) when the value for the feature $\boldsymbol{x}_i$ is lower (higher, resp.) than that of the counterfactual distribution $\mathcal{D}_C(f, \boldsymbol{x})$. This means that, if $\tau_i = +1$ the user must act upon $\boldsymbol{x}$ increasing x_i to get closer to the expected value of $\mathcal{D}_C(f, \boldsymbol{x})$, whilst if $\tau_i = -1$ they must act upon $\boldsymbol{x}$ decreasing x_i to achieve such goal.

To better understand the definition of the derived trends we can consider the example in Figure 2.i, that shows two inputs $\boldsymbol{x}^{(0)}$ and $\boldsymbol{x}^{(1)}$ with similar Shapley values $\boldsymbol{\phi}^{(0)}$ and $\boldsymbol{\phi}^{(1)}$, respectively. It is evident that, despite Feature A being the most important feature for both inputs, in order to overcome the adverse outcome for $\boldsymbol{x}^{(0)}$ Feature A must be *decreased* while it must be *increased* for input $x^{(1)}$. This intuition is matched by the derived trends $\tau_A^{(0)} = -1$ and $\tau_A^{(1)} = +1$ correctly suggesting that Feature A must be decreased or increased to change the decision for the input $x^{(0)}$ and $x^{(1)}$, respectively.

3.3 Counterfactual SHAP

After having described how the background distribution used for the computation of Shapley values plays a key role in giving a counterfactual interpretation to Shapley values and how explanations can be enriched with derived trends to provide guidance on direction of change of the features, we now formally define Counterfactual SHAP explanations.

[3] $sgn(x) = -1$ if $x < 0$, $sgn(x) = +1$ if $x > 0$, and $sgn(x) = 0$ otherwise.

Definition 3.3. The *Counterfactual SHAP* explanation (CF-SHAP) for an input $\boldsymbol{x} \in \mathcal{X}$ with respect to a distribution $\mathcal{D}_C(f, \boldsymbol{x})$ is the tuple $(\boldsymbol{\phi}, \boldsymbol{\tau}) \in \Phi \times \mathrm{T}$ where $\boldsymbol{\phi}$ and $\boldsymbol{\tau}$ are, respectively, the *Counterfactual SHAP values* and the *derived trends* for $\boldsymbol{x}$ with respect to $\mathcal{D}_C(f, \boldsymbol{x})$.

We note that the various mathematical properties of Shapley values, and by extension SHAP values, have been studied in depth [36, 38, 52]. These properties typically fit with human intuition for feature importance and are used as part of the basis for justifying the SHAP framework. Since Counterfactual SHAP values apply the SHAP framework, Counterfactual SHAP also satisfies the key properties of *additivity*, *missingness* and *consistency* as defined in [38]. We provide a more thorough discussion in the supplementary material (see Appendix B).

We now turn to the question of how we can numerically measure the "counterfactual-ability" of a feature attribution. We will tackle this problem in Section 4.

4 COUNTERFACTUAL-ABILITY

We seek to formalise the notion that certain feature attributions will be more useful for a model user in changing features to reverse an adverse outcome. It is important to emphasise that predicting how users might engage with explanations is a very challenging problem, and behaviour may vary dramatically depending on the context. We do not claim to resolve this problem. However, we aim to set up a flexible framework to measure the ability of an explanation to help a user reverse an adverse decision (in Section 4.1), before specialising this framework under certain sensible assumptions about how a user could act on the explanations that they receive (in Section 4.2).

4.1 General Evaluation Framework

To assess the *counterfactual-ability* of a feature attribution we first use the explanation to generate what we call the *induced counterfactual* point (from the explanation). Afterwards, we measure the "goodness" of the induced counterfactual point based on the cost that a user will incur when moving from the original input $\boldsymbol{x}$ to the induced counterfactual. We expect a good feature attribution to induce counterfactual points with lower cost and therefore higher counterfactual-ability. In our evaluation framework the *induced counterfactual* from the explanation $(\boldsymbol{\phi}, \boldsymbol{\tau})$ represents the counterfactual point towards which the user will tend to move (under some sensible assumptions) with minimum cost (for the user).

To formally define such notions we will now define two concepts: the *cost function* and the *action function*.

Cost Function. We measure the cost of changing an input $\boldsymbol{x}$ into another input $\boldsymbol{x'}$ via a *cost function*. Formally, a *cost function* is a function $c : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^+$ where $c(\boldsymbol{x}, \boldsymbol{x'})$ is the cost for the user of moving from $\boldsymbol{x}$ to $\boldsymbol{x'}$. A trivial example of cost function is the Euclidean distance.

Action Function. In order to describe how a user acts upon the input $\boldsymbol{x}$ based on the information provided by the feature attribution $\boldsymbol{\phi}$ and the trends $\boldsymbol{\tau}$ we use an *action function*. Formally, an *action function* is a function $A : \mathcal{X} \times \Phi \times \mathrm{T} \to 2^{\mathcal{X}}$ where $A(\boldsymbol{x}, \boldsymbol{\phi}, \boldsymbol{\tau}) \subset \mathcal{X}$ is a subset of the input space describing sensible changes the user may enact upon $\boldsymbol{x}$ when provided with an explanation $(\boldsymbol{\phi}, \boldsymbol{\tau})$. We will refer to $A(\boldsymbol{x}, \boldsymbol{\phi}, \boldsymbol{\tau})$ as the *action subset*. Note that we do not constrain the action subset to be finite.

Intuitively the output of an action function can be interpreted as a subset of the possible options that a user may consider when changing the input based on the information provided by the explanation. For instance, a user may consider as possible options only changes to the most important feature according to the explanation. In the most extreme scenario a user may ignore the information provided by $\boldsymbol{\phi}$ and $\boldsymbol{\tau}$ and therefore consider any change as a possible option; this would correspond to a constant action function always returning the whole input space as the action subset. In a more realistic scenario though, we expect the user to use the information provided by the feature attribution and therefore we expect the action subset to be a restricted subset of the input space, e.g., allowing only changes to the top-3 most important features according to $\boldsymbol{\phi}$ and only in the directions suggested by $\boldsymbol{\tau}$.

Induced Counterfactual and Counterfactual-Ability. The action and cost functions A and c describe, respectively, (1) how a user may act upon $\boldsymbol{x}$ given an explanation and (2) how difficult it is for a user to perform such actions. Given A and c, the *induced counterfactual* for an explanation is then simply defined as the counterfactual point lying in the action subset such that a user has minimum cost to reach and the *counterfactual-ability* is the negation of this cost. We now formally define the notions of counterfactual-ability and induced counterfactual.

Definition 4.1. The *counterfactual-ability* $CF(\boldsymbol{x}, \boldsymbol{\phi}, \boldsymbol{\tau})$ of an explanation $(\boldsymbol{\phi}, \boldsymbol{\tau})$ given an input $\boldsymbol{x}$ under an action function A and a cost function c is defined as:

$$CF(\boldsymbol{x}, \phi, \tau) = -c(\boldsymbol{x}, \boldsymbol{x'}) \quad \text{where } \boldsymbol{x'} = \underset{\substack{\boldsymbol{x'} \in A(\boldsymbol{x}, \boldsymbol{\phi}, \boldsymbol{\tau}) \\ F(\boldsymbol{x'}) \neq F(\boldsymbol{x})}}{\arg\inf} \; c(\boldsymbol{x}, \boldsymbol{x'})$$

and $\boldsymbol{x'} \in \mathcal{X}$ is referred to as the *induced counterfactual* from $(\boldsymbol{\phi}, \boldsymbol{\tau})$ given $\boldsymbol{x}$ under A and c.

Note that the action function is fixed for a given user; the goal in fact is to compare how different feature attributions perform under a (given) action function rather than optimising the action function for a specific user. We note that, in the degenerate case in which the action function is a constant function always returning the whole input space, solving this optimisation problem is equivalent to finding the counterfactual point $\boldsymbol{x'}$ with minimum cost from $\boldsymbol{x}$.

Note that the more points are included in the action subset, the smaller the counterfactual cost and therefore the larger the counterfactual-ability. However, in order to realise the full potential of the counterfactual-ability a user must be able to find the minimal cost counterfactual within this subset, and this task is in general intractable for a user with limited access to the model. We must therefore make certain assumptions about the behaviour of a user. We will introduce the precise assumptions that we make in the following section, but our choices of action subsets will be restricted to half-lines originating from the query point in the input space. This choice emerges from the following informal assumption: a user will make an effort to change certain features in response to receiving an explanation, and the fraction of total effort that a user puts into changing each feature is not dependent on the total amount of effort that they expend. Assuming this, the action subset takes the form of a line in a direction determined by the manner in

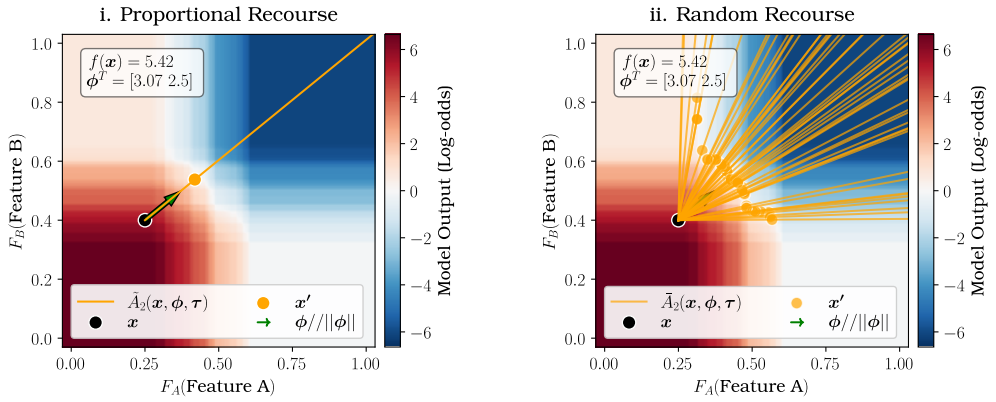

Figure 3: Algorithmic intuition of the action function. The orange lines in (i) and (ii) correspond to the action subsets $\tilde{A}_2(\boldsymbol{x}, \boldsymbol{\phi}, \boldsymbol{\tau})$ (proportional) and $\bar{A}_2(\boldsymbol{x}, \boldsymbol{\phi}, \boldsymbol{\tau})$ (random) as per definitions in Section 4.2. The corresponding orange points correspond to the points $\boldsymbol{x}'$ in the action subset with minimum cost according to L2-norm, i.e., the induced counterfactuals. Note that the proportional action subset (line) in (i) has the same direction of the Shapley values vector while the random action subset in (ii) is uniformly distributed. Note that F_A and F_B are the cumulative distribution functions of Feature A and B, respectively. This means that the feature axes are in the quantile space as per definitions of the action functions.

which the user chooses to apportion their effort in response to the explanation given.

4.2 Instances of the Evaluation Framework

After defining the general concepts of action function and cost function we now define some concrete instances that we use in this paper. To do so, we start with a number of assumptions designed to create sensible action and cost functions in the context of algorithmic recourse. Intuitively, the assumptions aim to cast the feature attribution as a suggested direction for a user to move in feature space, and the counterfactual-ability will therefore measure the (negation of the) cost to reach the decision boundary along this line. In the scope of this paper, we use the following set of assumptions:

Assumption 1: trend-aware recourse. When changing a feature a user moves its value in the direction suggested by the *derived trend* τ, e.g., if $\tau_{income} = +1$, a user will try to increase their income (as opposed to reducing it) since this change is more likely to give rise to a change in the prediction.

Assumption 2: adverse factors recourse. A user changes only features with *positive* Shapley values, i.e, the features contributing to the adverse prediction (as opposed to also improving features that are already good).

Assumption 3: sparse recourse. A user changes only the k most important features (with the highest Shapley values). In many industry applications, e.g., the rejection of a loan application, regulations require companies to provide only the most important features [63] therefore making sensible to assume that users have access only to the most important features. The parameter k will control how many features a user is allowed to change.

Assumption 4: recourse cost. We use the *quantile shift* as metric to measure the cost of the recourse, a common metric in the actionable recourse literature [64].

Regarding the way in which a user do a recourse we have 2 *alternative* assumptions that we will now introduce.

Assumption 5.A: proportional recourse. A user change the features *proportionally* to their Shapley values: the higher the Shapley value the more a feature will be changed compared to others. In some applications users may be provided not only with the list of the most important features but also with the magnitude of the Shapley values for each of the features. The intuition behind this assumption is that some users may tend to change more some features (i.e., of a greater quantile shift) than others that are not deemed as important by the Shapley values. We will denote the following action function satisfying assumptions 1, 2, 3, 4 and 5.A with $\tilde{A}_k$ where k is the number of top features that a user considers.[4]

$$\tilde{A}_k(\boldsymbol{x}, \boldsymbol{\phi}, \boldsymbol{\tau}) = \left\{\boldsymbol{x}' : Q(\boldsymbol{x}') - Q(\boldsymbol{x}) = -\lambda \boldsymbol{\phi} \odot \boldsymbol{\tau} \odot \mathbb{I}^k[\boldsymbol{\phi}], \forall \lambda > 0, \boldsymbol{x}' \in \mathcal{X}\right\} \tag{1}$$

where:

- $Q : \mathcal{X} \to [0,1]^m$ is a function computing the quantile[5] of each of the features with respect to the marginal distributions in the training data;
- $\mathbb{I}^k[\boldsymbol{\phi}] \in \{0,1\}^m$ is a (binary) indicator vector for the top-k features in $\boldsymbol{\phi}$ with positive Shapley value.[6]

[4] We use $\cdot$ and $\odot$ to indicate the dot and element-wise product, respectively.

[5] $Q : \mathcal{X} \to [0,1]^m$ takes a vector $\boldsymbol{x}$ and returns a vector $Q(\boldsymbol{x})$ with the marginal cumulative probability for each feature, i.e., $\boldsymbol{q} = Q(\boldsymbol{x})$ is such that $q_i = F_i(x_i)$ where F_i is the cumulative distribution function of feature i (estimated from the training set).

[6] Formally the indicator vector $\mathbb{I}^k[\boldsymbol{\phi}]$ for the top-k features in $\boldsymbol{\phi}$ with positive Shapley values is such that for every element $\mathbb{I}^k_i[\boldsymbol{\phi}]$ of $\mathbb{I}^k[\boldsymbol{\phi}]$:

$$\mathbb{I}^k_i[\boldsymbol{\phi}] = \begin{cases} 1 & \text{if } \phi_i > 0 \wedge i \in \arg\max_{S \subseteq \{1,\dots,m\}, |S| \leq k} \sum_{i \in S} \phi_i \\ 0 & \text{otherwise} \end{cases}$$

In other words, $\mathbb{I}^k_i[\boldsymbol{\phi}]$ is 1 iff ϕ_i is positive and ϕ_i is among the k largest Shapley value in ϕ, and 0 otherwise.

Method	Distribution	Description
Counterfactual SHAP		
CF-SHAP K-NN	$\mathcal{D}_{K\text{-NN}}(f, \boldsymbol{x})$	K-nearest neighbours of $\boldsymbol{x}$ such that $F(\boldsymbol{x}) = 1$ †
Baselines		
SHAP TRAIN	$\mathcal{D}_{\text{TRAIN}}$	Training set
SHAP D-LAB	$\mathcal{D}_{\text{DIFF-LAB}}$	Samples in the training set such that $y = 1$ (label)
SHAP D-PRED	$\mathcal{D}_{\text{DIFF-PRED}}(f)$	Samples in the training set such that $F(\boldsymbol{x}) = 1$ (prediction)

Table 1: Explanation methods used in the experiments divided among CF-SHAP and baselines. (†) We used $K = 100$, see Appendix C for additional experiments with a smaller or larger number of neighbours K.

The intuition behind this action function is that the feature attribution should provide a suggested direction to the user that takes them towards the decision boundary; we reflect this in Equation 1 by using $\boldsymbol{\tau}$ to limit change in a certain direction (Assumption 1) and by using $\boldsymbol{\phi}$ to enforce changes that are proportional to the feature attribution (Assumption 5.A). However, realistic actions will not involve changes to every feature; rather, a user may focus on making changes to only the top-k most important features (Assumption 3) that are adversely contributing to the prediction (Assumption 2); we reflect this using $\mathbb{I}^k[\boldsymbol{\phi}]$ in Equation 1. We use the quantile shift as a normalised metric for recourse cost (Assumption 4), enforced in Equation 1 by normalizing $\boldsymbol{x}$ and $\boldsymbol{x}'$ with Q. Finally, note that λ intuitively represents the amount of effort that a user put in changing their features, the higher λ the farther $\boldsymbol{x}'$ is from $\boldsymbol{x}$.

The action subset induced by our action function is a semi-infinite line in the normalized quantile input space in the direction of the Shapley vector with its sign adjusted to match the monotonic trend. To better understand this concept we can consider Figure 3.i, showing an example of the action subset induced for an input $\boldsymbol{x}$ and an attribution $\boldsymbol{\phi}$.

Assumption 5.B: random recourse. A user change the features *randomly*. For every user a random "utility vector" will be used to describe which features should be changed more than others. The objective behind this assumption is two-fold; firstly, it introduces an element of robustness in the evaluation. In fact, we do not know how for different users some features may be more or less costly to change, and using a random utility among the top-k features is an attempt to model this situation. Secondly, in some real-world applications users may not be provided with the Shapley values of each feature but only with a list of the top-k most important features, therefore making the use of a proportional recourse infeasible. We will denote the following action function satisfying assumptions 1, 2, 3, 4 and 5.B with $\bar{A}_k$ where k is the number of top features that a user considers.[4]

$$\bar{A}_k(\boldsymbol{x}, \boldsymbol{\phi}, \boldsymbol{\tau}) = \left\{\boldsymbol{x}' : Q(\boldsymbol{x}') - Q(\boldsymbol{x}) = -\lambda \boldsymbol{r}_{\mathcal{R}} \odot \boldsymbol{\tau} \odot \mathbb{I}^k[\boldsymbol{\phi}], \forall \lambda > 0, \boldsymbol{x}' \in \mathcal{X}\right\} \tag{2}$$

where $\boldsymbol{r}_{\mathcal{R}} = (R_1, \ldots, R_m)$ is a vector of random variables following the distribution $\mathcal{R}$. Q and $\mathbb{I}^k[\boldsymbol{\phi}]$ are defined as above. In our experiments $\mathcal{R}$ will be a uniform distribution between 0 and +1.

We note that our choices of action function are just two instantiations of the framework that we propose. We argue that casting the explanation as a random direction in which an input point may move (Assumption 5.B) is a more robust choice than casting explanations in a proportional fashion (Assumption 5.A) as it makes fewer assumptions on the user preferences, but we acknowledge that there is no clear answer to the question of how different users may act upon $\boldsymbol{x}$ given an explanation $(\boldsymbol{\phi}, \boldsymbol{\tau})$ in full generality.

Cost Function. We measured the cost using two alternative definitions based on the features quantile shift: the quantile shift under L1-norm (a.k.a., total quantile shift [64]) and L2-norm. Formally[5]:

$$c_{L1}(\boldsymbol{x}, \boldsymbol{x}') = \left\|Q(\boldsymbol{x}') - Q(\boldsymbol{x})\right\|_1$$

$$c_{L2}(\boldsymbol{x}, \boldsymbol{x}') = \left\|Q(\boldsymbol{x}') - Q(\boldsymbol{x})\right\|_2 .$$

We note that the instances of action and cost functions that we propose implicitly assume that the changes to features are made through interventions. We acknowledge that in some applications in which a full causal understanding of the input space is possible, one could instead resort to changes that take into consideration the causal structure. We believe that this topic represents an interesting future research direction (see Section 7).

5 EXPERIMENTS

In order to understand how Counterfactual SHAP (CF-SHAP) performs we compared it against existing feature attribution techniques. In particular, we compared CF-SHAP with SHAP using common input-invariant background distributions: $\mathcal{D}_{TRAIN}$, $\mathcal{D}_{D-PRED}$ and $\mathcal{D}_{D-LAB}$. Table 1 summarizes the baselines that we considered in our experiments. We refer to Section 3.1 for more details about these distributions.

To generate counterfactual points for Counterfactual SHAP we used K-nearest neighbours (K-NN). In practice, to generate counterfactual points for $\boldsymbol{x}$ we took the K nearest points to $\boldsymbol{x}$ in the training set, such that their predictions were different from the prediction for $\boldsymbol{x}$. In our experiments we used $K = 100$ and the Manhattan distance over the quantile space as distance metric for neighbours. This distribution will be referred to as $\mathcal{D}_{K-NN}(\boldsymbol{x}, f)$. We note that we set $K = 100$ when running our experiments in order to allow for a fair comparison with SHAP that, by default, randomly samples 100 points from the background distribution that it has been given.

We choose K-NN as technique rather than more complex counterfactual generation engines because (1) few counterfactual generation techniques are able to generate multiple counterfactuals and

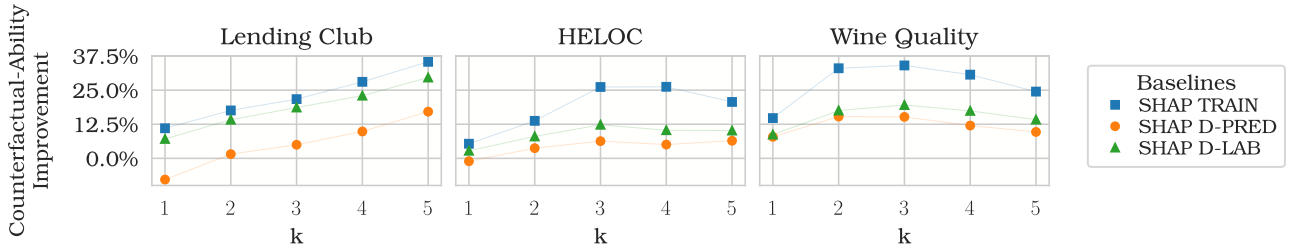

Figure 4: Counterfactual-ability improvement (as defined in Section 5) of CF-SHAP with respect to the baselines (SHAP TRAIN, SHAP D-LAB,SHAP D-PRED). The plots show how the counterfactual-ability improves when varying the number of top-k features a user has access to. In particular, this plots show the results for the improvement in counterfactual-ability under c_{L1} cost function and random recourse ($\tilde{A}_k$). Each line represents a baseline.

even fewer a diverse set thereof, especially in the context of decision tree-based models; (2) most importantly, the choice of K-NN as the technique for the generation of counterfactuals allows us to showcase the performance of Counterfactual SHAP while separating it from the performance of the counterfactual generation engine used. For example, using K-NN allows us to generate counterfactual points that are on-manifold (since they are points of the training set), an issue with which many counterfactual generation techniques struggle [32].

To run the experiments we used 3 publicly available datasets: **HELOC** (Home Equity Line Of Credit) [17], **LC** (Lending Club Loan Data) [27] and **WQ** (UCI Wines Quality) [11]. For each dataset, we trained an XGBoost model [10] and used TreeSHAP [37] to generate explanations. We refer to Appendix A for more details about the datasets and the experimental setup.

Counterfactual-ability. Counterfactual-ability represents a key metric to evaluate explanations in counterfactual terms as it measures the ability of a feature attribution to help a user reverse an adverse decision with minimal cost. We expect good feature attributions to have higher counterfactual-ability. For each dataset, we measured the percentage of times in which CF-SHAP has higher counterfactual-ability than that of the baselines and subtracted the number of times in which CF-SHAP has instead lower counterfactual-ability. We measured this over 4,000 rejected (i.e. with $F(\boldsymbol{x}) = 1$) random samples in the test set (or all of the rejected samples if less than 4,000 were available). Formally, we measure the improvement in counterfactual-ability as follows.

$$\text{Counterfactual-Ability Improvement} = \mathbb{E}_{\boldsymbol{x} \in D_{F(\boldsymbol{x})=1}} \left[\mathbb{1}\left[CF(\boldsymbol{x}, \boldsymbol{\phi}, \boldsymbol{\tau}) > CF(\boldsymbol{x}, \boldsymbol{\phi}_{BASE}, \boldsymbol{\tau}^G) \right] - \mathbb{1}\left[CF(\boldsymbol{x}, \boldsymbol{\phi}, \boldsymbol{\tau}) < CF(\boldsymbol{x}, \boldsymbol{\phi}_{BASE}, \boldsymbol{\tau}^G) \right] \right]$$

where $(\boldsymbol{\phi}, \boldsymbol{\tau})$ and $(\boldsymbol{\phi}_{BASE}, \boldsymbol{\tau}^G)$ are the the explanations for CF-SHAP and the baseline, respectively, $D_{F(\boldsymbol{x})=1}$ is the set of rejected samples and $\mathbb{1}$ is the boolean indicator[7]. Figure 4 show the results using the random action function ($\tilde{A}_k$) and the cost function with L1-norm (c_{L1}). We refer to the supplementary material (see Appendix D) for additional results with alternative definitions of action function and cost function. We report the main findings.

- Despite K-NN being a crude method for counterfactual generation, CF-SHAP beats (i.e., $> 0\%$) the baselines for all 3 datasets.
- There seems to be a positive correlation between the number of top-k features that are allowed to change and the improvement in counterfactual-ability. This suggests that CF-SHAP is able not only to find the top-1 most important feature to change to reverse the prediction, but it is also better performing than the baselines in identifying the top-2 and top-3 most important features that a user must change to reverse an adverse outcome.
- The results are robust with respect to the action function ($\tilde{A}_k$ or $\tilde{A}_k$) and cost function (c_{L1}, c_{L2}) used. We refer to Appendix D for details on this results.
- Depending on the dataset the best choice for the hyperparameter K of K-NN ranges from $K = 10$ to $K = 100$. We refer to Appendix C for detailed experiments on this hyperparameter.

We note that to allow for a fair comparison of the counterfactual-ability of CF-SHAP with that of the baselines (1) we equipped explanations that do not provide a derived trend with a "global" trend $\boldsymbol{\tau}^G$ obtained using the Pearson correlation of the features and target values in the training set[8]; (2) in order to to implement the random action function, we generated a random vector $\boldsymbol{r}_{\mathcal{R}}$ (see Equation 2) for each sample. This means that, as one would

[7] $\mathbb{1}[\cdot] = 1$ if the boolean expression $\cdot$ is true, and $\mathbb{1}[\cdot] = 0$ otherwise.

[8] Formally, given the training set $[\boldsymbol{X}_1, \ldots, \boldsymbol{X}_m]$, $\boldsymbol{y}$, we define the *global trend* as a vector $\boldsymbol{\tau}^G \in \{-1, 0+1\}^m$ such that:

$$\tau_i^G = sgn\left(\rho_{\boldsymbol{X}_i, \boldsymbol{y}}\right) \quad \forall i \in \{1, \ldots, m\}$$

where sgn is the sign function and $\rho_{\boldsymbol{X}_i, \boldsymbol{y}}$ is the Pearson's r between the column $\boldsymbol{X}_i$ of the training set (corresponding to feature i) and the labels vector $\boldsymbol{y}$.

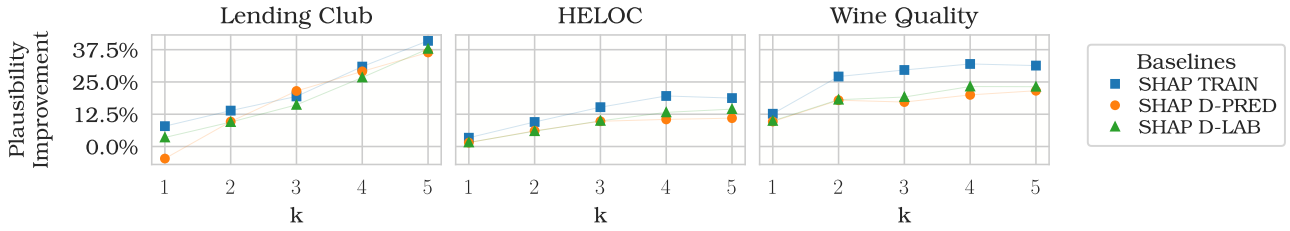

Figure 5: Plausibility improvement (as defined in Section 5) of CF-SHAP with respect to the baselines (SHAP TRAIN, SHAP D-LAB,SHAP D-PRED). The plots show how the plausibility improves when varying the number of top-k features a user has access to. In particular, this plots show the results for the improvement in plausibility under L1-norm (p_{L1}), c_{L1} cost function and random recourse ($\bar{A}_k$). Each line represents a baseline.

expect, all explanation techniques have been tested using the same "utility vectors" (one for each sample) modelling different users' preferences on the features to change; (3) the counterfactual-ability results are reported in relative terms to allow for the comparison of explanations for which counterfactual-ability is $-\infty$ (when no induced counterfactual can be found in the action set).

Plausibility. As pointed out in Section 2.2, plausibility is another important desideratum for counterfactuals. We expect a good feature attribution to induce counterfactuals that are not less plausible than those that can be found using input-invariant distributions. We measured the *plausibility* of the induced counterfactual in terms of the density of the region in which it lies so that, in practice, we computed the average distance of the induced counterfactual from its 5 nearest neighbours (measured again using quantile shifts). Formally, we measure plausibility in two ways:

$$p_{L1}(\boldsymbol{x}, \boldsymbol{x}') = -\mathbb{E}_{\boldsymbol{x}^* \in \mathcal{NN}(\boldsymbol{x}', 5)} \left[\left\| Q(\boldsymbol{x}^*) - Q(\boldsymbol{x}') \right\| \right]_1$$

$$p_{L2}(\boldsymbol{x}, \boldsymbol{x}') = -\mathbb{E}_{\boldsymbol{x}^* \in \mathcal{NN}(\boldsymbol{x}', 5)} \left[\left\| Q(\boldsymbol{x}^*) - Q(\boldsymbol{x}') \right\| \right]_2$$

where $\mathcal{NN}$ is a function $\mathcal{NN} : \mathcal{X} \times \mathbb{N}_{>0} \rightarrow 2^{\mathcal{X}}$ such that $\mathcal{NN}(\boldsymbol{x}', n)$ is the set of the n nearest neighbours of $\boldsymbol{x}'$. Similarly to the counterfactual-ability, we computed the percentage of times in which CF-SHAP has higher plausibility than that of the baselines and subtracted the number of times in which CF-SHAP has lower counterfactual-ability, i.e., formally:

$$\text{Plausibility Improvement} = \mathbb{E}_{x \in D_{F(\boldsymbol{x})=1}} \left[\mathbb{1}\left[p(\boldsymbol{x}, \boldsymbol{x}'_{CF-SHAP}) > p(\boldsymbol{x}, \boldsymbol{x}'_{BASE}) \right] - \mathbb{1}\left[p(\boldsymbol{x}, \boldsymbol{x}'_{CF-SHAP}) < p(\boldsymbol{x}, \boldsymbol{x}'_{BASE}) \right] \right]$$

where $\boldsymbol{x}_{CF-SHAP}$ and $\boldsymbol{x}_{BASE}$ are the the induced counterfactuals for CF-SHAP and the baseline, respectively, and $D_{F(\boldsymbol{x})=1}$ is the set of rejected samples. We run the experiments over $4,000$ rejected (i.e. with $F(\boldsymbol{x}) = 1$) random samples for each dataset. Figure 5 shows the results using random recourse (i.e., action function $\bar{A}_k$) and total quantile shift cost (i.e., cost function c_{L1}). We note that CF-SHAP beats (i.e., $> 0\%$) the baselines in plausibility for all datasets and it is highly robust to the choice of plausibility normalisation, action function and cost function. We refer to Appendix D for detailed results with alternative action function, cost function and plausibility normalisation.

Execution Time. The execution time of CF-SHAP directly depends on the execution time of (1) the counterfactual generation technique and (2) the execution time of SHAP. We note the computational complexity of (Tree-)SHAP scales linearly with the size of the background dataset [38]. Our experiments showed that CF-SHAP has a similar execution time to the baselines. This means that K-NN do not add a significant overhead to the overall execution time of CF-SHAP. In particular, CF-SHAP explanations could be generated (on average) in as little as $620\mu s$ for the Wine Quality dataset, $646\mu s$ for the HELOC dataset and $2646\mu s$ for the Lending Club dataset. We refer to the supplementary material (see Appendix B) for a detailed benchmark on the execution time of CF-SHAP.

6 RELATED WORK

There has been recently an increasing interest in exploring the relationship between feature importance and counterfactual explanations. A recent work [69] has proposed a Bayesian decision theory-based approach to the computation of the Shapley values. In particular the idea of [69] is to optimize the choice of the background distribution for the computation of Shapley values maximizing the expected reward for the user, i.e., $\mathcal{D}^* = \arg\max_{\mathcal{D}^* \subseteq \mathcal{D}} E_{\boldsymbol{x} \sim \mathcal{D}^*}[r(\boldsymbol{x})]$, under a certain reward function r. The work provides a theoretical framework for modelling user preference and beliefs but lacks (by design) concrete (1) guidance on how to select $\mathcal{D}^*$, (2) how to update the reward function r based on the observed Shapley values and (3) how to interpret the feature attribution $\boldsymbol{\phi}$ into practical actions on the input $\boldsymbol{x}$ in order to (automatically) solve the optimisation problem without resorting to an update in human-in-the-loop fashion.

Other works have proposed to fill the gap between feature attribution techniques and counterfactual explanations by different means than Shapley values. In particular, [42] and [55] propose

techniques to generate feature attributions for differentiable models from a set of diverse counterfactual points but (contrary to us) they use frequency-based approaches, i.e., they give higher attribution to features that are more often changed in counterfactual points. This implies that also features potentially ignored by the model may receive a high feature importance because they are correlated with other features that are really used by the model. As remarked in [9] this behaviour may be desirable in some context as medical sciences but not in others, as in the credit scoring scenario in which users are ultimately interested in understanding why they have have been rejected *by the model* rather than which features correlate with rejection *in the data*. In [8] feature attributions are generated by approximating the minimal adversarial perturbation using an adversarially trained neural network on a (differentiable) neural network-based surrogate model. This approach tends to follow the most strictest interpretation of the "true to the model" paradigm [9] enforcing only the class change but does not directly allow for the enforcement of other constraints, e.g., regarding the plausibility of such changes, as we do by providing a background distribution that is based on counterfactuals.

Other works [15, 16, 48, 49] analyze the complementary problem to that we analyze in this paper: they show how feature attributions can guide the search of counterfactuals (while we investigate how techniques for the generation of counterfactuals can empower better feature attribution).

In general, many works have explored how to evaluate counterfactual explanations (e.g., [35, 50, 64]) and feature attributions (e.g., [19, 34, 46]) but few proposed a quantitative metric to evaluate feature attributions in counterfactual terms. In [70] the authors propose to evaluate feature attributions with a *fidelity error* for each of the features that (differently from counterfactual-ability) can be computed changing only a single feature at a time. We overcome this limitation with the parameter k, controlling the number of features that are allowed to change.

7 CONCLUSION AND FUTURE WORK

Towards the more general goal of unifying feature attribution techniques and counterfactual explanations, we have shown how using a set of counterfactuals as the background distribution for the computation of Shapley values allows one to obtain feature attributions that can better advise towards useful changes of the input in order to overcome an adverse outcome. We have also highlighted that the generation of feature attributions with a counterfactual intent requires one to enrich explanations with additional information to describe the direction of the change (i.e., the derived trends). We proposed a new quantitative framework to evaluate such an effect in terms of counterfactual-ability and plausibility based on the notions of action and cost functions. We evaluated CF-SHAP on 3 publicly available datasets and highlighted that using simpler counterfactual techniques such as those based on nearest-neighbours within CF-SHAP performs better than existing feature attribution methods.

Our proposal can be extended in several directions. Firstly, it would be interesting to explore alternative notions of action and cost functions, grounding their definition with findings in psychology concerning how users interpret feature attributions and how they consequently change their behaviour. For example, one possibility would be to expand the definition of action function to take into account user preferences for certain actions – this could be achieved by coupling each "possible action" returned by the action function with a probability. Secondly, testing our approach on different models (e.g., neural networks) and using (potentially model-agnostic) counterfactual explanation techniques as [13, 21, 28, 29, 31, 50] represents another interesting future direction. Lastly, investigating how the generation of feature attribution with a counterfactual flavour connects with causality would also be of great interest. This could be achieved by considering a conditional background distribution in the computation of Shapley values [1, 18, 26, 62, 68] or using counterfactual explanation techniques that explicitly make use of causality, e.g. [30, 66].

From a wider perspective, our work draws attention to some gaps in the literature that we believe are worthy of further investigation. On the one hand, the importance of techniques for the generation of diverse counterfactuals advocated by many practitioners [43, 53, 57]. On the other hand, it highlights how few techniques have the capabilities of efficiently generating diverse counterfactual explanations in the context of non-differentiable models that are among the most widely adopted in industry, e.g., ensembles of decision trees.

ACKNOWLEDGMENTS

Disclaimer. This paper was prepared for informational purposes by the Artificial Intelligence Research group of JPMorgan Chase & Co. and its affiliates ("JP Morgan"), and is not a product of the Research Department of JP Morgan. JP Morgan makes no representation and warranty whatsoever and disclaims all liability, for the completeness, accuracy or reliability of the information contained herein. This document is not intended as investment research or investment advice, or a recommendation, offer or solicitation for the purchase or sale of any security, financial instrument, financial product or service, or to be used in any way for evaluating the merits of participating in any transaction, and shall not constitute a solicitation under any jurisdiction or to any person, if such solicitation under such jurisdiction or to such person would be unlawful.

REFERENCES

[1] Kjersti Aas, Martin Jullum, and Anders Løland. 2021. Explaining Individual Predictions When Features Are Dependent: More Accurate Approximations to Shapley Values. *Artificial Intelligence* 298 (2021), 103502.
[2] Amina Adadi and Mohammed Berrada. 2018. Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI). *IEEE Access* 6 (9 2018), 52138–52160.
[3] Emanuele Albini, Antonio Rago, Pietro Baroni, and Francesca Toni. 2020. Relation-Based Counterfactual Explanations for Bayesian Network Classifiers. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence, IJCAI*. 451–457.
[4] Emanuele Albini, Antonio Rago, Pietro Baroni, and Francesca Toni. 2021. Influence-Driven Explanations for Bayesian Network Classifiers. In *PRICAI 2021: Trends in Artificial Intelligence*. 88–100.
[5] Solon Barocas, Andrew D Selbst, and Manish Raghavan. 2020. The Hidden Assumptions Behind Counterfactual Explanations and Principal Reasons. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency, FAccT*. 80–89.
[6] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador Garcia, Sergio Gil-Lopez, Daniel

Molina, Richard Benjamins, Raja Chatila, and Francisco Herrera. 2020. Explainable Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion* 58 (2020), 82–115.

[7] James Bergstra, Daniel Yamins, and David Cox. 2013. Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures. In *Proceedings of the 30th International Conference on International Conference on Machine Learning, ICML*. I–115–I–123.

[8] Matt Chapman-Rounds, Umang Bhatt, Erik Pazos, Marc-Andre Schulz, and Konstantinos Georgatzis. 2021. FIMAP: Feature Importance by Minimal Adversarial Perturbation. *Proceedings of the 35th AAAI Conference on Artificial Intelligence* 35, 13 (5 2021), 11433–11441.

[9] Hugh Chen, Joseph D Janizek, Scott Lundberg, and Su-In Lee. 2020. True to the Model or True to the Data?. In *ICML '20 Workshop on Human Interpretability*. arXiv:2006.16234

[10] Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD*. 785–794.

[11] Paulo Cortez, António Cerdeira, Fernando Almeida, Telmo Matos, and José Reis. 2009. Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems* 47, 4 (11 2009), 547–553.

[12] Kristijonas Čyras, Antonio Rago, Emanuele Albini, Pietro Baroni, and Francesca Toni. 2021. Argumentative XAI: A Survey. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence, IJCAI*, Vol. 5. 4392–4399.

[13] Susanne Dandl, Christoph Molnar, Martin Binder, and Bernd Bischl. 2020. Multi-objective counterfactual explanations. In *Proceedings of the 16th International Conference on Parallel Problem Solving from Nature*, Vol. 12269 LNCS. 448–469.

[14] High-Level Expert Group on Artificial Intelligence European Commission. 2019. *Ethics Guidelines for Trustworthy AI*. Technical Report.

[15] Xiaoli Fern and Quintin Pope. 2021. Text Counterfactuals via Latent Optimization and Shapley-Guided Search. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. 5578–5593.

[16] Carlos Fernández-Loría, Foster Provost, and Xintian Han. 2021. Explaining Data-Driven Decisions Explaining Data-Driven Decisions made by AI Systems: The Counterfactual Approach. arXiv:2001.07417

[17] FICO Community. 2019. Explainable Machine Learning Challenge. https://community.fico.com/s/explainable-machine-learning-challenge

[18] Christopher Frye, Damien de Mijolla, Tom Begley, Laurence Cowton, Megan Stanley, and Ilya Feige. 2021. Shapley Explainability on the Data Manifold. In *Proceedings of the 9th International Conference on Learning Representations (ICLR)*. 14.

[19] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Dino Pedreschi, Franco Turini, and Fosca Giannotti. 2018. Local Rule-Based Explanations of Black Box Decision Systems. arXiv:1805.10820

[20] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. 2019. A Survey of Methods for Explaining Black Box Models. *Comput. Surveys* 51, 5 (2019), 1–42.

[21] Masoud Hashemi and Ali Fathi. 2020. PermuteAttack: Counterfactual Explanation of Machine Learning Credit Scorecards. arXiv:2008.10138

[22] GitHub Issues. 2018. Interpretation of Kernel SHAP and Its Hyperparameters - Issue #23 https://github.com/slundberg/shap.

[23] GitHub Issues. 2019. Choosing the Background Set · Issue #391 · https://github.com/slundberg/shap.

[24] GitHub Issues. 2019. Interpretation of SHAP Values Away from the Mean · Issue #435 · https://github.com/slundberg/shap.

[25] GitHub Issues. 2019. ZestFinance Writeup on SHAP and Why It Shouldn't Be Used on Its Own · Issue #624 · https://github.com/slundberg/shap.

[26] Dominik Janzing, Lenon Minorics, and Patrick Bloebaum. 2020. Feature Relevance Quantification in Explainable AI: A Causal Problem. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics*. 2907–2916.

[27] Kaggle. 2019. Lending Club Loan Data. https://www.kaggle.com/wordsforthewise/lending-club

[28] Kentaro Kanamori, Takuya Takagi, Ken Kobayashi, and Hiroki Arimura. 2020. DACE: Distribution-Aware Counterfactual Explanation by Mixed-Integer Linear Optimization. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence, IJCAI*. 2855–2862.

[29] Amir-Hossein Karimi, Gilles Barthe, Borja Balle, and Isabel Valera. 2020. Model-Agnostic Counterfactual Explanations for Consequential Decisions. In *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics, AISTATS*. 895–905.

[30] Amir-Hossein Karimi, Julius von Kügelgen, Bernhard Schölkopf, and Isabel Valera. 2020. Algorithmic Recourse under Imperfect Causal Knowledge: A Probabilistic Approach. In *Proceedings of the 34th Conference on Neural Information Processing Systems, NeurIPS*.

[31] Amir-Hossein Karimi, Eth Zürich, Switzerland Bernhard Schölkopf, and Isabel Valera. 2021. Algorithmic Recourse: from Counterfactual Explanations to Interventions. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, FAccT*. 353—-362.

[32] Mark T Keane, Eoin M Kenny, Eoin Delaney, and Barry Smyth. 2021. If Only We Had Better Counterfactual Explanations: Five Key Deficits to Rectify in the Evaluation of Counterfactual XAI Techniques. In *Proceeding of the 30th International Joint Conference on Artificial Intelligence, IJCAI*. 4466–4474.

[33] I Elizabeth Kumar, Suresh Venkatasubramanian, Carlos Scheidegger, and Sorelle A Friedler. 2020. Problems with Shapley-value-based explanations as feature importance measures. In *Proceedings of the 37th International Conference on Machine Learning , ICML*. 5491–5500.

[34] Himabindu Lakkaraju, Ece Kamar, Rich Caruana, and Jure Leskovec. 2019. Faithful and Customizable Explanations of Black Box Models. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society, FAccT*. 131–138.

[35] Thibault Laugel, Marie-Jeanne Lesot, Christophe Marsala, Xavier Renard, and Marcin Detyniecki. 2019. The Dangers of Post-Hoc Interpretability: Unjustified Counterfactual Explanations. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence, IJCAI*. 2801–2807.

[36] Scott Lundberg. 2017. Supplementary Material to a Unified Approach to Interpreting Model Predictions: The Monotonicity Axiom Implies the Symmetry Axiom for Shapley Values.

[37] Scott M. Lundberg, Gabriel Erion, Hugh Chen, Alex DeGrave, Jordan M. Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee. 2020. From local explanations to global understanding with explainable AI for trees. *Nature Machine Intelligence* 2, 1 (1 2020), 56–67.

[38] Scott M Lundberg and Su-In Lee. 2017. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems, NeurIPS*. 4768–-4777.

[39] Luke Merrick and Ankur Taly. 2020. The Explanation Game: Explaining Machine Learning Models Using Shapley Values. In *International Cross-Domain Conference for Machine Learning and Knowledge Extraction, CD-MAKE*. 17–38.

[40] John W L Merrill, Geoff M Ward, Sean J Kamkar, Jay Budzik, and Douglas C Merrill. 2019. Generalized Integrated Gradients: A practical method for explaining diverse ensembles. *Journal of Machine Learning Research - Under Review* (2019). arXiv:1909.01869

[41] Tim Miller. 2019. Explanation in Artificial Intelligence: Insights from the Social Sciences. *Artificial Intelligence* 267 (6 2019), 1–38.

[42] R. K. Mothilal, Divyat Mahajan, Chenhao Tan, and Amit Sharma. 2021. Towards Unifying Feature Attribution and Counterfactual Explanations: Different Means to the Same End. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society, AIES*. 652–663.

[43] Ramaravind K. Mothilal, Amit Sharma, and Chenhao Tan. 2020. Explaining Machine Learning Classifiers through Diverse Counterfactual Explanations. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency, FAccT*. 607–617.

[44] Martin Pawelczyk, Sascha Bielawski, Johannes van den Heuvel, Tobias Richter, and Gjergji Kasneci. 2021. CARLA: A Python Library to Benchmark Algorithmic Recourse and Counterfactual Explanation Algorithms. In *Benchmark & Data Sets Track at the 36th Conference on Neural Information Processing Systems, NeurIPS*.

[45] Martin Pawelczyk, Klaus Broelemann, and Gjergji Kasneci. 2020. Learning Model-Agnostic Counterfactual Explanations for Tabular Data. In *Proceedings of The Web Conference 2020, WWW*. 3126—-3132.

[46] Gregory Plumb, Denali Molitor, and Ameet S Talwalkar. 2018. Model Agnostic Supervised Local Explanations. In *Advances in Neural Information Processing Systems, NeurIPS*. 2520—-2529.

[47] Rafael Poyiadzi, Kacper Sokol, Raul Santos-Rodriguez, Tijl De Bie, and Peter Flach. 2020. FACE: Feasible and Actionable Counterfactual Explanations. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society, AIES*. 344–350.

[48] Yanou Ramon, David Martens, Foster Provost, and Theodoros Evgeniou. 2020. A comparison of instance-level counterfactual explanation algorithms for behavioral and textual data: SEDC, LIME-C and SHAP-C. *Advances in Data Analysis and Classification* 14 (2020), 801–819.

[49] Shubham Rathi. 2019. Generating Counterfactual and Contrastive Explanations using SHAP. In *2nd Workshop on Humanizing AI (HAI) at IJCAI '19*. arXiv:1906.09293

[50] Kaivalya Rawal and Himabindu Lakkaraju. 2020. Beyond Individualized Recourse: Interpretable and Interactive Summaries of Actionable Recourses. In *Advances in Neural Information Processing Systems, NeurIPS*. 12187–12198.

[51] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why Should I Trust You?". In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD*. 1135–1144.

[52] Alvin E. Roth. 1988. *The Shapley Value: Essays in Honor of Lloyd S. Shapley*.

[53] Chris Russell. 2019. Efficient Search for Diverse Coherent Explanations. In *Proceedings of the 2019 Conference on Fairness, Accountability, and Transparency, FAccT*. 20–28.

[54] Lloyd Stowell Shapley. 1951. Notes on the n-Person Game-II: The Value of an n-Person Game.

[55] Shubham Sharma, Jette Henderson, and Joydeep Ghosh. 2020. CERTIFAI: A Common Framework to Provide Explanations and Analyse the Fairness and Robustness of Black-box Models. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society, AIES*. 166–172.

[56] Ravid Shwartz-Ziv and Amitai Armon. 2021. Tabular Data: Deep Learning Is Not All You Need. arXiv:2106.03253

[57] Barry Smyth and Mark T. Keane. 2021. A Few Good Counterfactuals: Generating Interpretable, Plausible and Diverse Counterfactual Explanations. arXiv:2101.09056

[58] Thomas Spooner, Danial Dervovic, Jason Long, Jon Shepard, Jiahao Chen, and Daniele Magazzeni. 2021. Counterfactual Explanations for Arbitrary Regression Models. In *ICML'21 Workshop on Algorithmic Recourse.* arXiv:2106.15212

[59] Ilia Stepin, Jose M. Alonso, Alejandro Catala, and Martin Pereira-Farina. 2021. A Survey of Contrastive and Counterfactual Explanation Generation Methods for Explainable Artificial Intelligence. *IEEE Access* 9 (2021), 11974–12001.

[60] Erik Strumbelj and Igor Kononenko. 2010. An Efficient Explanation of Individual Classifications using Game Theory. *Journal of Machine Learning Research* 11 (2010), 1–18.

[61] Agus Sudjianto and Scott Zoldi. 2021. The Case for Interpretable Models in Credit Underwriting. https://soundcloud.com/finreglab/agus-sudjiantoscott-zoldi-the-case-for-interpretable-models-in-credit-underwriting

[62] Mukund Sundararajan and Amir Najmi. 2020. The Many Shapley Values for Model Explanation. In *Proceedings of the 37th International Conference on Machine Learning*. 9269–9278.

[63] U.S. Congress. 2018. 12 CFR Part 1002 - Equal Credit Opportunity Act (Regulation B). https://www.consumerfinance.gov/rules-policy/regulations/1002/9/

[64] Berk Ustun, Alexander Spangher, and Yang Liu. 2019. Actionable Recourse in Linear Classification. In *Proceedings of the Conference on Fairness, Accountability, and Transparency, FAccT*. 10–19.

[65] Sahil Verma, Arthur Ai, John Dickerson, and Keegan Hines. 2020. Counterfactual Explanations for Machine Learning: A Review. arXiv:2010.10596

[66] Julius von Kügelgen, Amir-Hossein Karimi, Umang Bhatt, Isabel Valera, Adrian Weller, and Bernhard Schölkopf. 2022. On the Fairness of Causal Algorithmic Recourse. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence.*

[67] Sandra Wachter, Brent Mittelstadt, and Chris Russell. 2018. Counterfactual Explanations Without Opening the Black Box: Automated Decisions and the GDPR. *Harvard Journal of Law & Technology* 31 (2018), 1–52.

[68] Jiaxuan Wang, Jenna Wiens, and Scott Lundberg. 2021. Shapley Flow: A Graph-based Approach to Interpreting Model Predictions. In *Proocedings of the 24th International Conference on Artificial Intelligence and Statistics, AISTATS.*

[69] David S. Watson. 2022. Rational Shapley Values. In *Proceedings of the 2022 Conference on Fairness, Accountability, and Transparency, FAccT.*

[70] Adam White and Artur d'Avila Garcez. 2019. Measurable Counterfactual Local Explanations for Any Classifier. In *Proceedings of the 24th European Conference on Artificial Intelligence, ECAI.* 2529–2535.

A EXPERIMENTAL SETUP AND REPRODUCIBILITY

A.1 Datasets and Models

To run the experiments we used 3 publicly available datasets. Table 2 describes in details the datasets.

We split the data using a stratified 70/30 *random* train/test split for HELOC and WINE. For LC we split the data using a non-random 70/30 train/test split based on the loan issuance date (available in the original data).

We trained an XGBoost model [10] for each dataset. In particular, we hyper-trained the parameters using Bayesian optimization via hyperopt [7] for 2000 iterations maximizing the average validation ROC-AUC under a 5-fold cross validation. To reduce model over-parameterization during the hyper-parameters optimization we penalized high model variance, i.e., for each cross-validation fold, instead of using AUC_{val}, we used $AUC_{val} + (AUC_{val} - AUC_{train})$ where AUC_{train} and AUC_{val} are the training and validation ROC-AUC, respectively.

To compute the decision threshold (t) we used the ROC-AUC curve: we maximized the sum of the false positive rate (fall-out) and true positive rate (recall). Table 2 shows the decision threshold and the performance of each model.

A.2 Technical setup

The experiments were run using a `c6i.8xlarge` AWS virtual machine with 32 vCPUs (16 cores of 3.5 GHz 3rd generation Intel Xeon Scalable processor) and 64GB of RAM. XGBoost parameter `nthread` was set to 15.

We used a machine running `Ubuntu 20.04`. We used `Python 3.6.13`, `shap 0.39.0`, `sklearn 0.24.2` and `xgboost 1.3.3`.

A.3 Source Code

The code to reproduce the experiments will be made available at https://www.emanuelealbini.com/cfshap-facct2022.

A.3.1 SHAP Explanations. In order to compute Shapley values we used the TreeSHAP [37] available through the `TreeExplainer` class in the `shap` package[9] (for Python). We note that we computed the Shapley values on the model output (default setting of `shap`). We also remark that we used the interventional (a.k.a., non-conditional) version of SHAP (default setting of `shap`).

A.3.2 K-Nearest Neighbours. To compute the K-nearest neighbours implementation in `sklearn.neighbours`. To make our results indifferent to the size of the dataset we limited the k-nearest neighbours to be selected among a random sample of $10,000$ samples from the training set.

B COUNTERFACTUAL SHAP ALGORITHM: PROPERTIES AND EXECUTION TIME

We report in Algorithm 1 the procedure to compute Counterfactual SHAP explanations.

[9]The `shap` package can be found at https://github.com/slundberg/shap

B.1 Properties of Counterfactual SHAP

We will now discuss some of the properties of Counterfactual SHAP.

We note that Counterfactual SHAP is an *additive feature attribution method* as defined by [38] (or, equivalently, it satisfies *additivity*), i.e., $f(\boldsymbol{x}) = \phi_0 + \sum_{i \in \mathcal{F}} \phi_i$ where $\phi_0 = E_{\boldsymbol{x}' \in \mathcal{D}_C(f,\boldsymbol{x})}[f(\boldsymbol{x}')]$. We also note that Counterfactual SHAP values satisfy all the properties (local accuracy, missingness and consistency) satisfied by SHAP values. This is true because such properties are satisfied by SHAP values **independently of the background distribution**. In fact, the definition of Counterfactual SHAP values diverges from the definition of SHAP values only in terms of the background distribution used in the characteristic function.[10]

Additivity. As noted in Section 2.2 one of the objectives of counterfactual generation techniques is to try to minimize the distance between the input $\boldsymbol{x}$ and the counterfactual point $\boldsymbol{x}'$ and indirectly the distance of $\boldsymbol{x}'$ from the decision boundary. Since Counterfactual SHAP uses a set of counterfactual points as background distribution, we note that the closer (in terms of output) the generated counterfactuals are to the decision boundary, the better the sum of the Shapley values approximates the distance in model output of the query instance from the decision boundary. Indeed, rewriting the additivity property for Counterfactual SHAP we have

$$\sum_{i \in \mathcal{F}} \phi_i = f(\boldsymbol{x}) - \mathbb{E}_{\boldsymbol{x}' \in \mathcal{D}_C(\boldsymbol{x},f)}\left[f(\boldsymbol{x}')\right].$$

When using Counterfactual SHAP the average model output on the counterfactual distribution, $\mathbb{E}_{\boldsymbol{x}' \in \mathcal{D}_C(\boldsymbol{x},f)}[f(\boldsymbol{x}')]$, should approximate the threshold output value t, although this does depend on properties of the model and the counterfactual distribution. If the model is continuous and the counterfactuals are selected to be on the decision boundary then we will have $\mathbb{E}_{\boldsymbol{x}' \in \mathcal{D}_C(\boldsymbol{x},f)}[f(\boldsymbol{x}')] = t$, but this cannot be guaranteed theoretically for tree-based models. Empirically we see that $\mathbb{E}_{\boldsymbol{x}' \in \mathcal{D}_C(\boldsymbol{x},f)}[f(\boldsymbol{x}')]$ approximates t very closely for certain choices of counterfactual distribution. For example, by projecting the K-nearest neighbours onto the decision boundary along the line to the query instance, we can obtain a counterfactual distribution $D_C(\boldsymbol{x}, f)$ with $\mathbb{E}_{\boldsymbol{x}' \in \mathcal{D}_C(\boldsymbol{x},f)}[f(\boldsymbol{x}')]$ very close to the threshold (see Table 3). Using this counterfactual distribution instead of the K-nearest neighbours themselves results in extremely similar performance to that presented in Section 5, as shown in Figure 6.

Linearity[11]. For Shapley values, linearity states that given coalitional games Γ and Γ' with value functions v and w respectively, then the Shapley values $\phi_i(v+w)$ for the summed game $\Gamma + \Gamma'$ with value $v+w$ are given by the sum $\phi_i(v) + \phi_i(w)$ of the Shapley values of Γ and Γ'. In the context of machine learning models, this axiom states that for two models F_1 and F_2 and a **fixed background dataset** the SHAP values for the summed model $f_\Sigma = f_1 + f_2$ are given by the sum of the SHAP values for f_1 and f_2. This property is essential for rapidly calculating the SHAP values for ensemble models, as for example in the TreeSHAP algorithm [37].

CF-SHAP involves the calculation of SHAP values, and as such the linearity property as described above is inherited automatically

[10]In [38] the characteristic function is denoted with f_x while in this paper we use the canonical notation v from the game theory literature.

[11]The linearity property should not be confused with additivity used earlier in this paper (a.k.a., local accuracy, or efficiency), i.e., $f(\boldsymbol{x}) = \sum_{i \in \{0\} \cup \mathcal{F}} \phi_i$.

Dataset	Features	Size Train Set	Size Test Set	Decision Threshold*	Model Performance† ROC-AUC	Model Performance† Recall	Model Performance† Accuracy‡
HELOC (Home Equity Line Of Credit) [17]	23	6,909	2,962	0.3985	79.6%	81.6%	72.9%
LC (Lending Club Loan Data) [27]	20	961,326	411,998	0.3824	69.6%	79.8%	56.0%
WINE (UCI Wine Quality) [11]	11	3,428	1,470	0.4614	83.2%	80.7%	78.2%

Table 2: Characteristics of the datasets and models used in the experiments. (∗) The decision threshold is reported here in probability space (i.e., after passing the model output through a sigmoid); (†) performance metrics are computed on the test set; (‡) we note that the AUC-ROC and recall are better suited metrics for this applications (i.e., a "bad" customer being accepted is a more undesirable outcome than a "good" customer being rejected).

Algorithm 1 Counterfactual SHAP algorithm

procedure CF-SHAP(x, f)
  $D_C \leftarrow C(\boldsymbol{x}, f)$ ▷ The set of counterfactuals D_C is computed for x wrt. the model f using technique C.
  $\boldsymbol{\phi} \leftarrow SHAP(\boldsymbol{x}, f, D_C)$ ▷ The Shapley values $\boldsymbol{\phi}$ for $\boldsymbol{x}$ wrt. background dataset D_C and model f using SHAP.
  $\boldsymbol{\tau} \leftarrow Trends(\boldsymbol{x}, D_C)$ ▷ The trends $\boldsymbol{\tau}$ for $\boldsymbol{x}$ wrt. background dataset D_C are computed.
  return $(\boldsymbol{\phi}, \boldsymbol{\tau})$ ▷ Shapley values and trends are returned.
end procedure

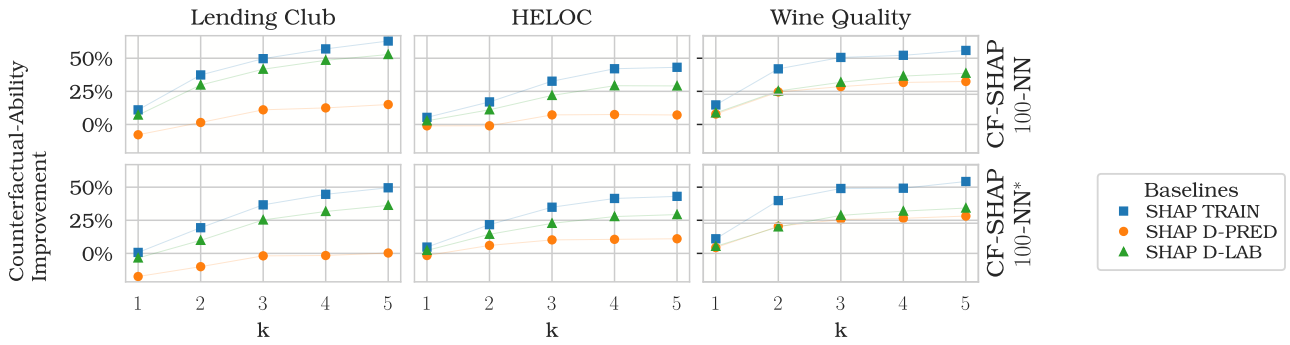

Figure 6: Counterfactual-ability improvement (as defined in Section 5) of CF-SHAP 100-NN and CF-SHAP 100-NN∗ with respect to the baselines. We note that CF-SHAP 100-NN and CF-SHAP 100-NN∗ have very similar performance in terms of counterfactual-ability. (∗) indicates the variant of CF-SHAP K-NN that generates counterfactuals by projecting the K-nearest neighbours onto the decision boundary along the line to the query instance.

	HELOC	LC	WQ
CF-SHAP 100-NN	0.524610	0.268826	0.663118
CF-SHAP 100-NN *	**0.067102**	0.080836	**0.076111**
SHAP D-LAB	0.330960	**0.037210**	0.567720
SHAP D-PRED	0.940211	0.693693	0.880631
SHAP TRAIN	0.437677	0.102876	0.456406

Table 3: Divergence of the average model output of the points in the background dataset from the threshold t for different distributions, i.e., $|t - \mathbb{E}_{\boldsymbol{x} \in \mathcal{D}}[f(\boldsymbol{x})]|$ where $\mathcal{D}$ is the background distribution used to compute the Shapley values. (∗) indicates the variant of CF-SHAP K-NN that generates counterfactuals by projecting the K-nearest neighbours onto the decision boundary along the line to the query instance.

for this calculation (allowing the use of efficient algorithms such as TreeSHAP). However, the use of a fixed background dataset is an important caveat. In CF-SHAP it is key that the background dataset is tailored both to the individual query instance and also to the model. Counterfactuals for one model may not be counterfactuals for another model on the same dataset, and it is therefore not the case that the CF-SHAP values for $f_1 + f_2$ are given by summing the CF-SHAP values for f_1 and f_2. Thus, when viewing the Counterfactual SHAP algorithm as a whole, we see that the linearity property will typically not hold since the counterfactual distributions will differ for different models.

This should not be seen as a limitation of Counterfactual SHAP, but rather as a necessary consequence of tailoring the contrastive explanations in a counterfactual way. Indeed, even using the SHAP algorithm with the background dataset given by $\mathcal{D}_{\text{D-PRED}}(f)$, the samples predicted differently by f, results in the failure of the linearity property in this sense. This can be seen in Figure 7; this figure shows two models in red and blue and the model that results

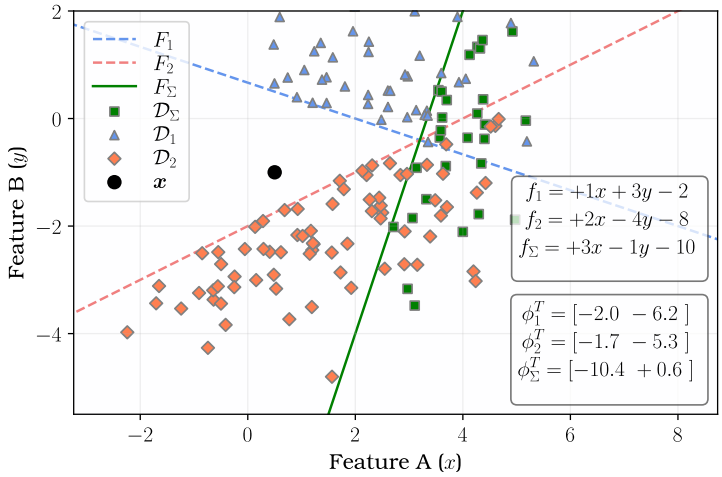

Figure 7: Linearity for Counterfactual SHAP explanations: solid lines indicate the models' (f_1, f_2 and $f_\Sigma = f_1 + f_2$) decision boundaries (F_1, F_2 and F_Σ); the black point indicates the input x; the coloured squares are samples of the background distribution $\mathcal{D}_{D-PRED}(f)$.

from their sum in green. For each model, we show points from the data that are predicted differently by each of these models. All of these points are counterfactual for their respective model, but the points which are counterfactual for the blue model (say) are not necessarily counterfactual for the red or green models.

B.2 Computational complexity

As one can deduce from Algorithm 1, the computation time of CF-SHAP is simply given by the the sum of the time to compute the counterfactual explanations, the Shapley values and the trends.

$$T(n) = T_C(n) + T_{SHAP}(n) + T_{Trends}(n) = O(n)$$

where n is the number of (counterfactual) points in the background dataset.

The computation time of the interventional variant of TreeSHAP (that we use in this paper) depends linearly on the number of samples in the background distribution [37], i.e., $T_{SHAP}(n) = O(n)$. The computation time of the counterfactuals explanations depends on the counterfactual generation technique that is being used. In the case of k-nearest neighbours, the overall computation time depends linearly on the number of neighbours generated, i.e., once again $T_{K-NN}(n) = O(n)$. The computation of the trends is also linear with respect to the number of counterfactual explanations since it is a simple average, therefore $T_{Trends}(n) = O(n)$.

B.3 Execution Time: Experiments

To experimentally test the execution time of the explanation techniques we recorded the (average) time taken by each explanation technique to generate a single explanation.

Experiment setup. In particular, in order to estimate the execution time for a single explanation, we run each explanation algorithm on all the samples in the dataset for a time $t > 0.1$ seconds; we then divide t by the number of explanations that the method computed in such time frame. To account for error introduced by the OS scheduler we run the experiments 10 times and, since all the explanations techniques that we experimented with are deterministic, we take the minimum execution time.

	HELOC	LC	WQ
Counterfactual SHAP			
CF-SHAP 100-NN	$646\mu s$	$2,646\mu s$	$620\mu s$
CF-SHAP 10-NN	$189\mu s$	$419\mu s$	$165\mu s$
Baselines			
SHAP D-LAB (n = 100)	$619\mu s$	$3,024\mu s$	$652\mu s$
SHAP D-PRED (n = 100)	$597\mu s$	$3,015\mu s$	$640\mu s$
SHAP TRAIN (n = 100)	$380\mu s$	$2,209\mu s$	$408\mu s$

Table 4: Execution time of different explanations techniques. We report the (average) execution time to generate the explanation of a single sample. n represents the number of points in the random sample drawn from the background distribution. Refer to Appendix B.3 for more details about the setup.

Results. The results are reported in Table 4. The main findings are reported as follows.

- The impact of K-nearest neighbour computation on the Counterfactual SHAP values computation is minimal. For example, we can see from Table 4 that there is only a 4.3% increase in execution time of CF-SHAP 100-NN with respect to that of SHAP D-PRED ($n = 100$).
- The execution time of (Tree-)SHAP scales linearly with the size of the background dataset confirming the theoretical results in [38]. This means that explanations techniques using the full training or variants thereof (SHAP D-LAB/PRED) have a considerably larger run-times than other techniques. We remark that, in practise, to reduce the execution time bottle-neck, such distribution are replaced with others approximating them, e.g., a random sample or of k-means medoids.

K	Counterfactual-Ability Improvement (%)														
	HELOC					Lending Club					Wine Quality				
$k=$	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
1	−8.6	−6.4	−18.1	−26.7	−31.7	−28.0	−13.8	−5.0	2.6	8.0	−0.4	11.5	8.2	8.2	8.5
3	−4.9	2.4	1.0	−2.4	−3.7	−19.2	−4.2	3.8	8.8	18.3	6.1	14.3	15.0	15.2	16.1
5	−3.1	3.9	4.2	2.6	2.4	−16.6	−0.6	6.6	10.2	20.3	8.2	17.2	17.0	16.7	18.7
10	−1.9	**6.4**	7.0	4.7	5.3	−14.0	**2.4**	**6.7**	**13.0**	**21.6**	9.6	**19.0**	17.8	**18.9**	**19.1**
20	−1.0	**6.4**	**8.0**	**5.4**	5.4	−12.2	2.1	6.2	12.4	19.5	**10.5**	18.0	**18.3**	18.8	18.5
50	−1.0	5.2	7.2	5.3	6.2	−9.5	**2.4**	5.7	11.2	18.4	9.4	18.3	17.8	15.3	13.7
100	−1.0	3.7	6.3	5.1	**6.4**	−7.8	1.6	5.0	9.8	17.1	7.9	15.3	15.2	12.0	9.7
250	−1.1	2.0	4.7	4.6	6.0	−4.4	2.1	3.9	8.9	13.2	4.6	8.6	5.2	0.9	−0.1
500	−1.2	1.4	3.7	4.0	5.8	−2.7	1.2	3.2	8.1	10.6	1.2	−1.6	−8.2	−14.8	−18.9
1000	**−0.7**	0.5	1.6	2.2	3.7	**−1.2**	0.8	1.8	5.5	7.7	0.7	−13.2	−26.6	−34.1	−37.0

Table 5: Counterfactual-ability improvement (as defined in Section 5) of CF-SHAP with respect to SHAP D-PRED (higher is better) for different datasets when varying the number of nearest neighbors (K) selected as counterfactuals and the number of features allowed to change (k).

C CHANGING THE NUMBER OF NEAREST NEIGHBOURS (K)

The techniques based on K-nearest neighbours that we used to compute counterfactuals in our experiments have as main parameter the number of neighbours K. In this appendix we will provide the results showing how the counterfactual-ability of explanations changes when varying K. We remark, as already noted in Section 5, that to allow for a fair comparison with Counterfactual SHAP, we set $K = 100$ when running our main experiments. In fact, by default, `shap` randomly sample 100 points from the a background dataset it has provided with.

Experimental Setup. We measured the improvement in counterfactual-ability as described in Section 5 over 4,000 explanations. For this experiment we chose to compare against the toughest baseline to beat: SHAP D-PRED. We varied the number of neighbors used by CF-SHAP from 1 to 1000 and recorded the results.

Results. Table 5 reports the improvement in counterfactual-ability of CF-SHAP when compared to SHAP D-PRED.

In particular, we note that:

- Depending on the dataset the best choice for the hyper-parameter K of K-NN ranges from $K = 10$ to $K = 100$.
- Increasing the number of neighbours over 100 does not result in an improvement in counterfactual-ability. This matches with the intuition that increasing the number of counterfactuals included in the background dataset means that the explanation will be "less tailored" to the specific user (associated with the input) $\boldsymbol{x}$.
- Decreasing the number of neighbours under 10 does not result in an improvement in counterfactual-ability. This again matches with the intuition that proving (too) few points as background distribution makes the explanation less robust because it reduces the ability of CF-SHAP to describe the decision boundary. In particular, we can see how providing a single counterfactual point results in a very sharp fall in counterfactual-ability of the explanation.

D ROBUSTNESS TO DIFFERENT SETTINGS

To show the robustness of our evaluation we report additional results for the experiments on counterfactual-ability improvement and plausibility improvement under different action functions and cost functions. In the main text we

Experimental Setup. We run the same experiments for the counterfactual-ability and plausibility as reported in Section 5 using alternative definitions of action function and cost function that we have introduced in the paper. In particular in Section 5 we reported the results for the improvement in counterfactual-ability (Figure 4) and the improvement in plausibility (Figure 5) under the assumption of:

- random recourse (i.e. using action function $\bar{A}_k$) and total quantile shift cost (i.e., using cost function c_{L1}).

In this appendix we report the results under the following alternative assumptions:

- random recourse (action function $\bar{A}_k$) and quantile shift cost with L2 norm (i.e., using cost function c_{L2});
- proportional recourse (i.e., using action function $\tilde{A}_k$) and total quantile shift cost (i.e., using cost function c_{L1});
- proportional recourse (i.e., using action function $\tilde{A}_k$) and total quantile shift cost under L2 norm (i.e., using cost function c_{L2}).

Results. Figure 8 and 9 shows the results for the improvement in counterfactual-ability and plausibility, respectively, under these different assumptions. We note that both the counterfactual-ability and the plausibility experiments results are highly robust to the choice of action and cost function. This suggests that the CF-SHAP performs better than the baselines also when changing the underlying assumptions on how users may act on the explanation (action function) and how users measure the cost of the recourse (cost function).

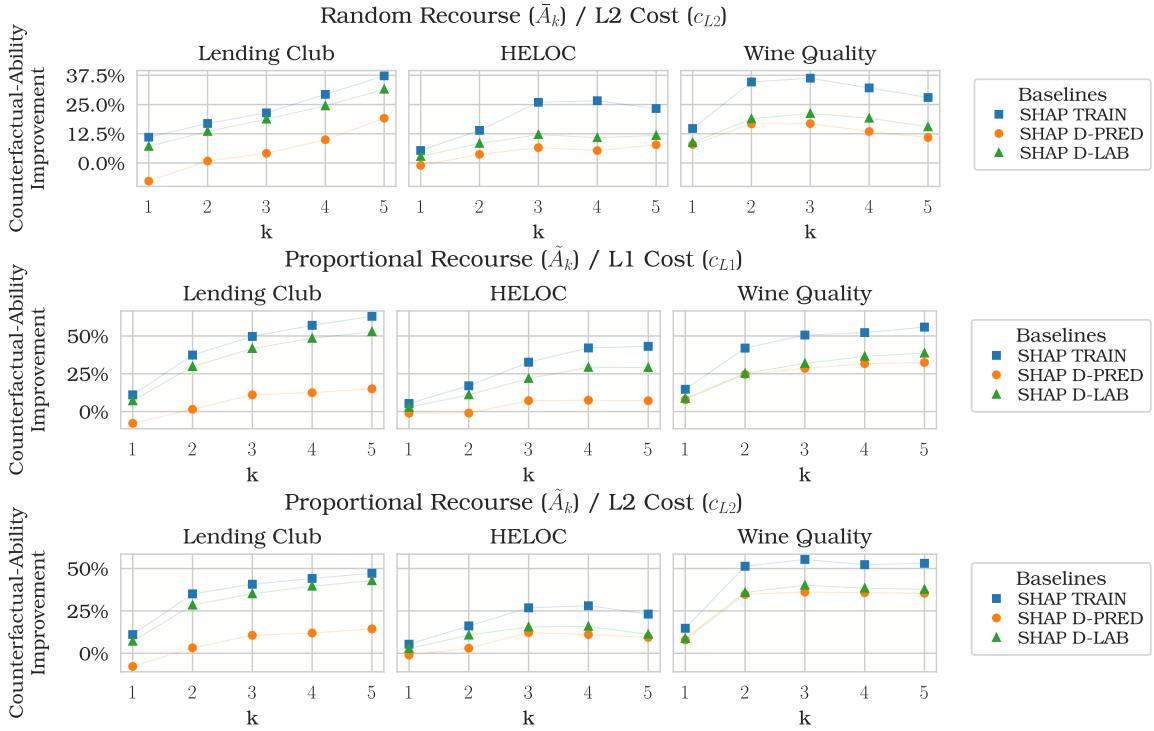

Figure 8: Improvement in counterfactual-ability under different assumptions (i.e., using alternative definitions of action function and cost function). This is the equivalent of Figure 4 under different assumptions. See Appendix D for more details.

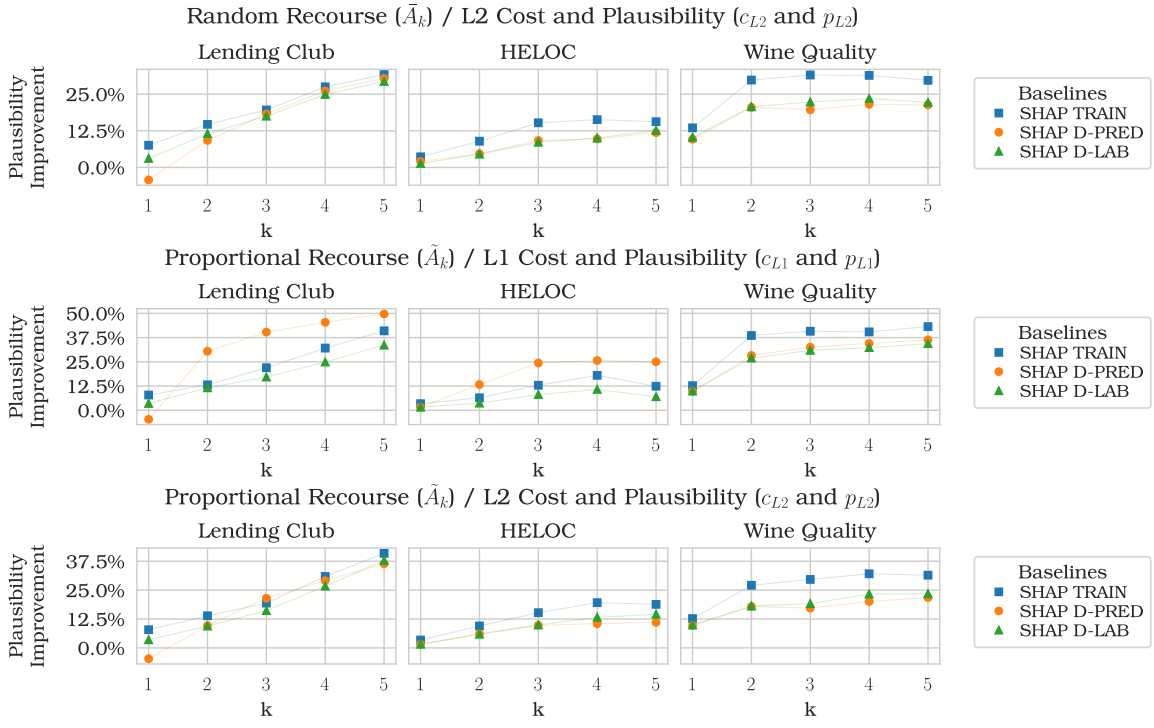

Figure 9: Improvement in plausibility under different assumptions (i.e., using alternative definitions of action function and cost function). This is the equivalent of Figure 5 under different assumptions. See Appendix D for more details.