FEATURE RESPONSIVENESS SCORES: MODEL-AGNOSTIC EXPLANATIONS FOR RECOURSE

Anonymous authors
Paper under double-blind review

ABSTRACT

Machine learning models are often used to automate or support decisions in applications such as lending and hiring. In such settings, consumer protection rules mandate that we provide consumers who receive adverse decisions with a list of "principal reasons." In practice, lenders and employers identify principal reasons as the top-scoring features from a *feature attribution* method. In this work, we study how such practices align with one of the underlying goals of consumer protection – *recourse* – i.e., educating individuals on how to achieve a desired outcome. We show that standard attribution methods can highlight features that will not lead to recourse – providing individuals with *reasons without recourse*. We propose to score features on the basis of *responsiveness*, i.e., the proportion of interventions that can lead to a desired outcome. We develop efficient methods to compute responsiveness scores for any model and any dataset under complex actionability constraints. We present an empirical study on the responsiveness of explanations in lending, and demonstrate how responsiveness scores can highlight features that support recourse and mitigate harm by flagging instances with fixed predictions.

1 INTRODUCTION

Machine learning models are now routinely used to automate or support decisions about people in domains such as employment [8, 44], consumer finance [26], and public services [59, 17, 24]. In such applications, explanations are often seen as an essential tool to protect consumers who are adversely affected by the predictions of a machine learning model [57, 52, 47, 5]. Existing and proposed laws and regulations include provisions that require lenders or employers to provide explanations to individuals in such situations [1, 57, 52, 18]. In the United States, for example, the adverse action notice requirement in the Equal Credit Opportunity Act mandates that lenders provide "principal reasons" explaining why individuals are denied credit [1]. In the European Union, Article 86 of the AI Act [18] grants individuals a right to obtain explanations to describe the "main elements" of decisions in areas such as employment, education, financial systems, government benefits, law enforcement, and border control.

Our reliance on explanations as a tool for consumer protection reflects widespread beliefs about the value of transparency in such settings [14] – i.e., that revealing information can protect and empower consumers [47]. In the United States, for example, the adverse action requirement is motivated by the fact that presenting consumers with "principal reasons" can: (1) promote *anti-discrimination* by revealing that a prediction was based on protected characteristics; (2) streamline *rectification*, by revealing that a prediction was based on incorrect feature values; and; (3) support *recourse* by educating individuals on how to improve their decision in a future application. Regulators provide lenders with substantial flexibility in complying with these requirements [51]. In practice, lenders who use machine learning create adverse action notices by applying methods for feature attribution such as SHAP or LIME [20]. Given a model, these methods explain its predictions by assigning scores to each feature. In this way, model deployers identify the top-scoring features for an adverse prediction and present them to consumers as the "principal reasons" for their decision (see Fig. 1).

In this work, we study how to explain model predictions to support one of the main goals of consumer protection: *recourse*. We focus on achieving recourse through the use of feature attribution – techniques that are widely used in practice. Our work is motivated by the fact that regulations seek to achieve multiple goals; we claim that it is useful to align the design of an explanatory method with the goals it seeks to achieve. To this end, we study how well existing approaches for feature attribution

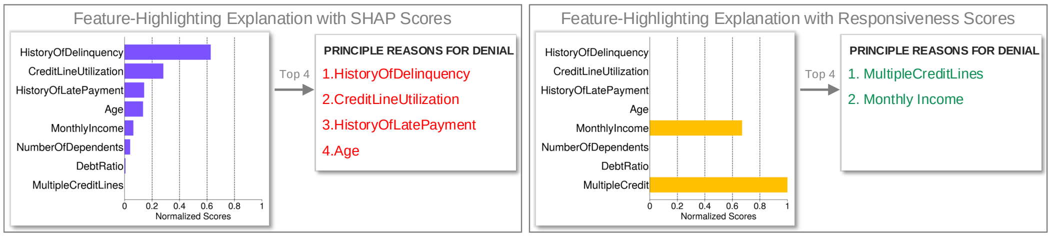


Figure 1: Feature-highlighting explanations for a person denied credit by a logistic regression model for a lending task (see givemecredit, Section 4). We show explanations from top-scoring features using SHAP [38] (left) and responsiveness scores (right). As shown, SHAP highlights 4 features, of which 3 are immutable (Age, HistoryOfLatePayment, HistoryOfDeliquency) and 1 is unresponsive (CreditLineUtilization). In contrast, explanations built from responsiveness scores (right) only highlight up to 4 features that an individual could change to attain a desired prediction.

methods support recourse, and develop an approach tailored to communicating with respect to this goal. Our main contributions include:

1. We present a feature attribution method to measure the responsiveness of predictions from a model. The *responsiveness score* measures the proportion of interventions on a specific feature that attain a desired outcome. Our approach highlights features that can be changed to achieve a desired outcome, and flags instances where recourse is impossible or difficult to obtain.

2. We develop model-agnostic methods to compute feature responsiveness scores using *reachable sets*. Our methods can evaluate scores for any model, which can be paired with theoretical guarantees to flag harm, and that can be readily adapted to achieve other goals.

3. We conduct a comprehensive empirical study on the responsiveness of feature attribution in consumer finance. Our results demonstrate that common feature attribution methods output *reasons without recourse* by highlighting features that do not provide recourse, and underscore the benefits of our approach. Namely, returning responsive features and refraining from presenting explanations when individuals do not have recourse.

4. We include a Python library to measure feature responsiveness available at our anonymized repository.

Related Work Our work is related to a stream of research on post-hoc explanations [45, 38, 46, 37, 61, 3, 39]. We focus on methods for feature attribution, which are designed to evaluate the importance of features in the prediction of a model at a given point. Many methods are built for use cases in model development [e.g., 45, 38], but are now used to construct “feature-highlighting explanations” to comply with regulations on explanations in consumer applications [see e.g., 5, 20].

Our work reveals a critical failure mode in feature attribution methods: *reasons without recourse*. These methods highlight features that agents cannot change to attain their target prediction. This failure expands the list of critiques of local explanation methods, including their susceptibility to manipulation [e.g., 36, 49, 50, 4, 25] and their indeterminancy [40, 56, 10, 7]. Our work complements recent impossibility results showing that common attribution methods (e.g., SHAP) cannot be used to reliably characterize salient behavior (e.g., recourse) [see e.g., 6, 21], establishes the prevalence of this effect in practice, and develops a principled approach to detect and mitigate it.

We study a new task related to algorithmic recourse [54, 30] and feature attribution: measuring the *responsiveness* of features. Our task differs from traditional approaches to recourse; we seek to estimate the prevalence of actions in each dimension, rather than identify the closest action [see e.g., 31]. These actions guarantee recourse only when features change by a prescribed amount. When explanations omit the required feature change magnitude, agents may not attain recourse – falling short or overshooting the target range. On the contrary, our approach provides a more valuable perspective by highlighting features that maximize the chances of recourse. We build on a line of work that elicits and enforces complex actionability constraints [54, 35] to construct feature responsiveness scores that are model-agnostic and can adapt to address practical challenges in providing recourse – e.g., robustness [42, 43, 53] or causality [32, 12, 23] – and beyond (e.g., adversarial robustness).

2 Problem Statement

We formalize the problem of explaining the predictions of a machine learning model through feature attribution. We consider a standard classification task where we wish to predict a label $y \in \mathcal{Y}$ from a set of d *features* $\boldsymbol{x} = [x_1, x_2, \ldots, x_d] \in \mathcal{X} \subseteq \mathbb{R}^d$. We assume that we are given a model $h : \mathcal{X} \to \mathcal{Y}$ where each instance, $\boldsymbol{x}_i \in \mathcal{X}$, represents a person, and each feature, $j \in [d]$, represents a semantically meaningful characteristic for the task at hand (e.g., `age` or `income`). [1]

We consider a task where we must explain the predictions of a model to individuals who fail to receive a *target prediction*, y^{t}. For example, in a lending task where a model would predict $y \in \{0, 1\}$ and $y = 1$ indicates that an applicant will repay their loan, we would set the target prediction as $y^{\mathrm{t}} = 1$ and explain the predictions for all applicants for whom $h(\boldsymbol{x}_i) = 0$.

Feature-Highlighting Explanations Our goal is to construct explanations where each feature is *responsive* – i.e., can be changed independently to attain the target prediction y^{t}. The standard practice of explaining predictions is to use *feature-highlighting explanations* [see e.g. 5]. These explanations consist of a list of "most important" features from a specified method that we convert into a natural language description [e.g., a reason code 20].

Feature Attribution Methods The standard approach in constructing feature-highlighting explanations is to use feature attribution methods [20].

Definition 1. Given a model $h : \mathcal{X} \to \mathcal{Y}$ and its training dataset $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, a *feature attribution method* for point $\boldsymbol{x}_i$ is a function $\boldsymbol{\phi}(\boldsymbol{x}_i \mid h, \mathcal{D}) : \mathcal{X} \to \mathbb{R}^d$, where the jth element of the output, $\phi_j(\boldsymbol{x}_i \mid h, \mathcal{D})$ is the attribution for feature $j \in [d]$.

In what follows, we write $\boldsymbol{\phi}(\boldsymbol{x}_i)$ instead of $\boldsymbol{\phi}(\boldsymbol{x}_i \mid h, \mathcal{D})$ when h and $\mathcal{D}$ are clear from context. This function captures the behavior of several methods that are used to explain the prediction of a model:

- *Local Linear Explainers* [see e.g., 45, 62, 60, 15]: Given a model h and a point $\boldsymbol{x}_i$, these methods fit a linear model $g : \mathbb{R}^d \to \mathbb{R}$ to approximate the decision boundary surrounding $\boldsymbol{x}_i$ such that $g(\boldsymbol{x}') = \langle \boldsymbol{\phi}(\boldsymbol{x}_i), \boldsymbol{x}' \rangle$. The resulting attribution for each feature is its weight in g.
- *Shapley Value Methods* [see e.g., 38, 27, 22]: Given a model h and a point $\boldsymbol{x}_i$, these methods cast features as players in a cooperative game, and estimate $\phi_j(\boldsymbol{x}_i)$ as the marginal contribution of feature j to the prediction $h(\boldsymbol{x}_i)$ under axioms of social choice [48].

Given a model h and its training dataset $\mathcal{D}$, these methods output a vector of scores $\boldsymbol{\phi}(\boldsymbol{x}_i)$ that capture the importance of each feature for the prediction $h(\boldsymbol{x}_i)$. In general, these scores satisfy the following properties:

- *Relevance*: A feature with an attribution score $\phi_j(\boldsymbol{x}_i) = 0$ is not relevant to the prediction for $\boldsymbol{x}_i$ – i.e., it can be changed arbitrarily without changing the prediction [see e.g., the "missingness" axiom in 38].
- *Strength*: Features with larger attribution scores have a larger impact on the prediction – i.e., if $|\phi_j(\boldsymbol{x}_i)| > |\phi_{j'}(\boldsymbol{x}_i)|$, then feature j has a stronger contribution to the prediction than feature j'.

These properties allow model developers to comply with consumer protection rules, but can promote misinterpretation among consumers [33].

Reasons without Recourse One of the key failure modes of machine learning in consumer-facing applications is that models can assign *fixed predictions* – predictions that cannot be changed by their decision subjects (see e.g., Table 1). In lending, for example, models that assign fixed predictions can inflict harm through *preclusion* – i.e., when an applicant is denied a loan based on a fixed prediction, they are permanently barred from credit access.

Reasons without recourse are important for the context of explanations because it is impossible to provide feature-highlighting explanations for recourse to someone who is assigned a fixed prediction. It may be that they cannot act upon any of the features. Alternatively, they may be able to change

[1] We assume that feature values are bounded so that $x_j \in [l_j, u_j]$ and $\|\boldsymbol{x}\| \leq B$ for all $\boldsymbol{x} \in \mathcal{X}$ and B is sufficiently large. This assumption holds for most semantically meaningful features [see 54]. Some features have bounds by construction (e.g., binary features). In other cases, we can set loose bounds (e.g., for `income`).

their features, but none of them allows them to obtain the target prediction. However, existing feature attribution methods can generate an explanation, presenting individuals with *reasons without recourse*. This can lead to harm by misleading individuals to invest effort into cases that they cannot change.

Accounting for Actionability Models will assign fixed predictions when features cannot change or can change in specific ways. In principle, we can detect these instances by explicitly considering actionability constraints. Given these challenges, we introduce machinery to capture how features can change at the instance level.

Definition 2. An *action* is a vector $\boldsymbol{a} = [a_1, \ldots, a_d] \in \mathbb{R}^d$ that a person can perform to change their features from $\boldsymbol{x}_i$ to $\boldsymbol{x}_i + \boldsymbol{a} = \boldsymbol{x}' \in \mathcal{X}$. Given a point $\boldsymbol{x}_i \in \mathcal{X}$, the *action set* $A(\boldsymbol{x}_i)$ contains all possible actions for $\boldsymbol{x}_i$. We assume that every action set contains the *null action* $\boldsymbol{\emptyset} \in A(\boldsymbol{x}_i)$.

Action sets captures how we can change features from a given point as a set of *actionability constraints*. As shown in Table 2, we can elicit complex constraints from human experts in natural language, and convert them into equations that we can embed into an optimization problem. In this way, we can enforce actionability in – for example – algorithms to find recourse actions [see e.g., 54, 35].

Features		Label Counts		Best Model
`age` $\geq$ `60`	`has_IRA`	n_0	n_1	$h(\boldsymbol{x})$
0	0	51	10	0
0	1	7	30	1
1	0	21	8	0
1	1	31	17	0

Table 1: Stylized lending task where the best model assigns fixed predictions to two points, highlighted in red. We predict a binary label $y = \texttt{repayment}$ from two binary features (x_1, x_2) = (`age` $\geq$ `60`, `has_IRA`). We fit a classifier data with n_0 negative labels and n_1 positive labels for each $(x_1, x_2) \in \{0, 1\}^2$. Individuals with $x_1 = 1$ can only change their features to $(x_1, x_2) \in \{(1, 0), (1, 1)\}$ since `age` $\geq$ `60` is immutable.

Class	Example	Features	Actionability Constraint
Immutability	`age` cannot change	$x_j = \texttt{age}$	$a_j = 0$
Monotonicity	`recent_payment` can only increase	$x_j = \texttt{recent\_payment}$	$a_j \geq 0$
Integrality	`late_payments` must be positive integer ≤ 12	$x_j = \texttt{late\_payments}$	$a_j \in \mathbb{Z}^+ \cap [0 - x_j, 12 - x_j]$
Encoding Validity	preserve one-hot encoding of categorical feature `housing` $\in \{\texttt{own}, \texttt{rent}, \texttt{other}\}$	$x_k = \mathbb{1}[\texttt{housing=own}]$ $x_l = \mathbb{1}[\texttt{housing=rent}]$ $x_m = \mathbb{1}[\texttt{housing=other}]$	$a_j + x_j \in \{0, 1\}$ for $j \in \{k, l, m\}$ $\sum_{j \in \{k,l,m\}} a_j + x_j = 1$

Logical Implication	if `has_savings_account` $=$ `TRUE`
	then `savings_balance` ≥ 0
	else `savings_balance` $= 0$
		$x_j = \texttt{has\_savings\_account}$
		$x_k = \texttt{savings\_balance}$
			$a_j + x_j \in \{0, 1\}$
			$a_k + x_k \in [0, 10^{12}]$
			$a_j + x_j \leq 10^{12}(x_k + a_k)$

Causal Implication	if `years_of_account_history` increases then `age` will increase commensurately	$x_j = \texttt{years\_of\_account\_history}$ $x_k = \texttt{age}$	$x_j + a_j \leq x_k + \delta_k$ $\delta_k \in [0, 100]$

Table 2: Examples of actionability constraints on semantically meaningful features for a lending task (see Appendix B for additional examples). Each constraint can be expressed in natural language and embedded into an optimization problem using standard techniques in mathematical programming [see, e.g., 58].

To highlight features that are responsive, we must assign a score to features that accounts for actionability constraints. In practice, the actionability constraints for a given feature will include constraints that pertain to the feature as well as other features. We refer to the features that may change as a result of interventions on feature j as *downstream features*, C_j.

Definition 3. Given an action set $A(\boldsymbol{x}_i)$ for a point $\boldsymbol{x}_i \in \mathcal{X}$, the *action set* for feature $j \in [d]$ is:

$$A_j(\boldsymbol{x}_i) := \{\boldsymbol{a} \in A(\boldsymbol{x}_i) \mid \boldsymbol{a}_j \neq 0 \wedge \boldsymbol{a}_k = 0, k \in [d] \setminus C_j\}.$$

Here, the *downstream set* $C_j := \{k \in [d] \setminus \{j\} \mid \boldsymbol{a}_j \neq 0 \implies \boldsymbol{a}_k \neq 0 \;\; \forall \boldsymbol{a} \in A(\boldsymbol{x})\}$ is the subset of all features that must change as a result of interventions on feature j.

Definition 3 captures cases where actions on a feature can induce changes in other features. Such cases can stem from deterministic causal relationships – e.g., increasing `years_of_account_history` should lead to a commensurate change in `age`. In general, they can capture dependencies that would not be included in a traditional causal graph – e.g., changing a categorical attribute will require switching a binary feature "off" while turning another binary feature "on" (so that $x_j = 1 \rightarrow 0 \implies x'_j = 0 \rightarrow 1$).

3 Measuring Feature Responsiveness

In this section, we introduce our main technical contribution – the *responsiveness score*. We first define the responsiveness score, then discuss its interpretation and computation.

3.1 Responsiveness Scores

Our goal is to measure the *responsiveness* of the prediction of a model at a point $\boldsymbol{x}_i$ with respect to the set of feasible actions on specific features. We propose to measure the sensitivity for each feature through the *feature responsiveness score*.

Definition 4. Given a model $h : \mathcal{X} \to \mathcal{Y}$, a point $\boldsymbol{x}_i$ with action set $A(\boldsymbol{x}_i)$ and feature $j \in [d]$, the *responsiveness score* for feature j is defined as:

$$\mu_j(\boldsymbol{x}_i \mid h, A(\boldsymbol{x}_i)) := \Pr(h(\boldsymbol{x}') = y^{\mathrm{t}} \mid \boldsymbol{x}' = \boldsymbol{x}_i + \boldsymbol{a}, \boldsymbol{a} \in A_j(\boldsymbol{x}_i))$$

The responsiveness score for feature j captures the proportion of intervention on feature j that changes the prediction of a model h at $\boldsymbol{x}_i$. In what follows, we write $\mu_{\boldsymbol{x}}(j)$ instead of $\mu_{\boldsymbol{x}|h,A(\boldsymbol{x}_i)}(j)$ when h and $A(\boldsymbol{x}_i)$ are clear from context. Given a feature where $\mu_j(\boldsymbol{x}_i) = p$, we know that $100(p)\%$ of the interventions on j, $\boldsymbol{a} \in A_j(\boldsymbol{x}_i)$ will change the prediction of the model. Thus, all actions to a feature where $\mu_j(\boldsymbol{x}_i) = 0$ would not change the prediction while all actions on a feature where $\mu_j(\boldsymbol{x}_i) = 1$ would result in a different prediction.

These interpretations are contingent on the actionability constraints used to compute the responsiveness score. In the simplest case, actionability constraints encode indisputable constraints on how a feature can be changed (e.g., feature encoding or physical limits) and so the responsiveness score for a given feature represents an upper bound on its responsiveness: "*at most* $100\mu_j(\boldsymbol{x}_i)\%$ of interventions on feature j attain a target prediction." Such constraints let us flag undeniable instances of harm.

Safeguards for Consumer Protection One benefit of responsiveness scores is that we can reliably use them to detect when consumers are assigned fixed predictions, and when feature-based explanations can provide recourse.

Remark 1. *Given a model $h : \mathcal{X} \to \mathcal{Y}$, let $\mu_{\boldsymbol{x}_i}(1), \ldots, \mu_{\boldsymbol{x}_i}(d)$ denote the responsiveness scores of $\boldsymbol{x}_i \in \mathcal{X}$ with respect to the action set $A(\boldsymbol{x}_i)$.*

a) If $\mu_j(\boldsymbol{x}_i) > 0$ for some feature $j \in [d]$, then h can provide recourse to $\boldsymbol{x}_i$ through an intervention on j.

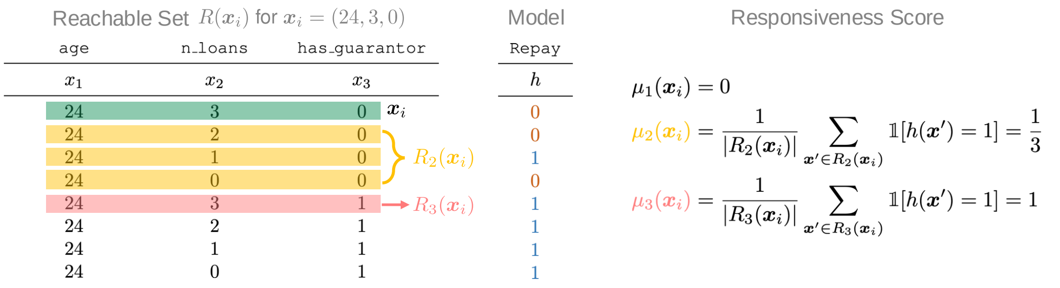
b) If $\mu_j(\boldsymbol{x}_i) = 0$ for all features $j \in [d]$, then either: (i) h assigns a fixed prediction to $\boldsymbol{x}_i$, or (ii) h can only provide recourse to $\boldsymbol{x}_i$ through actions that alter two or more features.

Remark 1a) states that every person ($\boldsymbol{x}_i$) who receives a positive responsiveness score for at least one feature has recourse. This implies that when we construct feature-highlighting explanations using the top-k responsiveness scores, we will *only* provide explanations to individuals who have recourse. Remark 1b) also illustrates how the responsiveness scores can flag for potential harm when $\mu_j(\boldsymbol{x}_i) = 0$ and allows us to mitigate harm on a case-by-case basis. In case (i) – where a person is assigned fixed predictions – we would refrain from providing explanations to avoid misleading consumers, and flag the issue so that model development can be potentially revisited. In case (ii) – where a person is assigned predictions that can change through multiple actions – we could proceed in a similar manner to case (i), or provide explanations that highlight subsets of responsive features or that include an explicit warning against presumptions of feature independence. We can mitigate harm if instead of providing misleading explanations of fixed predictions they are flagged to model owners or auditors.

3.2 Computing Scores with Reachable Sets

We compute responsiveness scores using a *reachable set*:

Definition 5. Given a point $\boldsymbol{x}_i$ and its action set $A(\boldsymbol{x}_i)$, we refer to the set of all points that are attainable through actions in $A(\boldsymbol{x})$ as the *reachable set*: $R(\boldsymbol{x}_i) := \{\boldsymbol{x}_i + \boldsymbol{a} \mid \boldsymbol{a} \in A(\boldsymbol{x}_i)\}$. We refer to the subset of points that are reachable through actions on feature $j \in [d]$ as the *reachable set* for feature j and denote it as: $R_j(\boldsymbol{x}_i) := \{\boldsymbol{x}_i + \boldsymbol{a}' \mid \boldsymbol{a}' \in A_j(\boldsymbol{x}_i)\}$.



age	n_loans	has_guarantor	Repay
x_1	x_2	x_3	h
24	3	0	0
24	2	0	0
24	1	0	1
24	0	0	0
24	3	1	1
24	2	1	1
24	1	1	1
24	0	1	1

$$\mu_1(\boldsymbol{x}_i) = 0$$

$$\mu_2(\boldsymbol{x}_i) = \frac{1}{|R_2(\boldsymbol{x}_i)|} \sum_{\boldsymbol{x}' \in R_2(\boldsymbol{x}_i)} \mathbb{1}[h(\boldsymbol{x}') = 1] = \frac{1}{3}$$

$$\mu_3(\boldsymbol{x}_i) = \frac{1}{|R_3(\boldsymbol{x}_i)|} \sum_{\boldsymbol{x}' \in R_3(\boldsymbol{x}_i)} \mathbb{1}[h(\boldsymbol{x}') = 1] = 1$$

Figure 2: Simple example of how to compute responsiveness scores involving three independent features. age is an immutable feature, n_loans is a discrete feature taking values from 0 to 3 and has_guarantor is a binary feature. The original prediction of 0 is shown in the row highlighted in green. Interventions for n_loans, has_guarantor are highlighted yellow and red respectively. The responsiveness score for age is 0 since it is immutable. Although the full reachable set is not required for computation, we include it for demonstrative purposes.

Reachable sets represents an alternative way to store and process information about actionability at the instance level. In particular, a reachable set $R(\boldsymbol{x}_i)$ encodes this information as a set of feature vectors that can be reached through feasible actions.

Given reachable sets $R_j(\boldsymbol{x}_i)$ for each feature $j \in [d]$, we can calculate responsiveness scores for any model by querying its predictions (see Fig. 2). This has the benefits that:

- it is *model-agnostic* and hence can be readily used in place of any approach in existing frameworks.
- we only need to compute the reachable set *once* and can re-use the same set multiple times (e.g., with different models).
- it allows for *other notions* of responsiveness (see Section 3.3). This is because a reachable set is a collection of feature vectors. We can easily implement these through matrix-vector operations.

Algorithm 1 Sample Reachable Set for Feature j

```
Require: point $\boldsymbol{x}_i$
Require: action set $A_j(\boldsymbol{x}_i)$ for feature $j \in [d]$
Require: sample size $N \in \mathbb{N}$
  $\hat{R}_j \leftarrow \varnothing, \quad A_j \leftarrow A(\boldsymbol{x}_i)$
  repeat
  $\boldsymbol{a}^* \leftarrow$ Sample1DAction($\boldsymbol{x}_i, A_j$)
      if CheckFeasibility($\boldsymbol{a}^*, A_j$) then
          $\hat{R}_j \leftarrow \hat{R}_j \cup \{\boldsymbol{x}_i + \boldsymbol{a}^*\}$
      end if
  until $|\hat{R}_j| = N$
Output $\hat{R}_j$, $N$ reachable points via actions on $j$
```

Sampling for Continuous Features Algorithm 1 presents a method to generate a sampled reachable set $\hat{R}_j(\boldsymbol{x}_i)$ for feature j. $\mathsf{Sample1DAction}(\boldsymbol{x}_i, A_j)$ samples uniformly according to feature j's separable actionability constraint – i.e. constraints that only apply to j such as lower and upper bounds. Then we perform rejection sampling by checking whether the sampled action is feasible with $\mathsf{CheckFeasibility}(\boldsymbol{x}_i, A_j)$.

The independent samples in $\hat{R}_j(\boldsymbol{x}_i)$ allow us to *estimate* the responsiveness score and operationalize its safety guarantees through the following result:

Definition 6. Given a point $\boldsymbol{x}_i \in \mathcal{X}$, let $\hat{R}_j(\boldsymbol{x}_i) \subseteq R_j(\boldsymbol{x}_i)$ denote a sample of N points drawn uniformly from the reachable set for feature j. Given any model $h : \mathcal{X} \to \mathcal{Y}$, we can estimate the responsiveness score for feature j as

$$\hat{\mu}_j(\boldsymbol{x}_i) := \frac{1}{N} \sum_{\boldsymbol{x}' \in \hat{R}_j(\boldsymbol{x}_i)} \mathbb{1}[h(\boldsymbol{x}') = y^{\mathsf{t}}].$$

Given a level of significance $\alpha \in (0, 1)$, we have that:

$$\Pr(\mu_j(\boldsymbol{x}_i) \in [\tilde{\mu}_j(\boldsymbol{x}_i) - \mathcal{E}, \tilde{\mu}_j(\boldsymbol{x}_i) + \mathcal{E}]) \geq 1 - \alpha$$

Here: $\mathcal{E} = \kappa\sqrt{\frac{1}{\tilde{N}}\tilde{\mu}_j(\boldsymbol{x}_i)(1 - \tilde{\mu}_j(\boldsymbol{x}_i))}$, $S = |\{\boldsymbol{x}' \in \hat{R}_j(\boldsymbol{x}_i) \mid h(\boldsymbol{x}') = y^{\mathsf{t}}\}|$ denotes the number of responsive points, $\kappa := \Phi^{-1}(1 - \frac{\alpha}{2})$, $\Phi(\cdot)$ is the Normal CDF, and $\tilde{\mu}_j(\boldsymbol{x}_i) := \frac{1}{N+\kappa^2}\left(S + \frac{\kappa^2}{2}\right)$ is the corrected estimator to improve coverage when $\mu_j(\boldsymbol{x}_i) = 0$ or 1 [see 9].

Here, the significance level $\alpha \in (0,1)$ represents the probability that the true $\mu_j(\boldsymbol{x}_i)$ lies outside of the interval $[\tilde{\mu}_j(\boldsymbol{x}_i) - \mathcal{E}, \tilde{\mu}_j(\boldsymbol{x}_i) + \mathcal{E}]$. In practice, practitioners can apply the result above by setting α and determining a minimal sample size to estimate $\mu_j(\boldsymbol{x}_i)$ with the desired level of guarantees on consumer safety claims (i.e. Remark 1).

This is a general-purpose approach to compute responsiveness scores for both continuous and discrete features. However, we cannot identify fixed predictions with certainty using a sampled reachable set.

Enumeration for Discrete Features Algorithm 2 presents a method to generate $R_j(\boldsymbol{x}_i)$ for a given point $\boldsymbol{x}_i$ and feature $j \in [d]$ in discrete space. We solve the optimization problem:

$$\mathsf{Find1DAction}(\boldsymbol{x}_i, A_j) := \operatorname{argmin} \|\boldsymbol{a}\| \text{ s.t. } \boldsymbol{a} \in A_j(\boldsymbol{x}_i).$$

We can formulate $\mathsf{Find1DAction}(\boldsymbol{x}_i, A_j)$ as a mixed-integer program (see Appendix A). This approach extends that of Kothari et al. [35] to generate a single-feature reachable set j. This approach is limited to discrete feature spaces and has larger computational and storage overhead compared to the aforementioned sampling approach. Having said that, enumeration allows us to calculate exact responsiveness scores and certify recourse feasibility.

Algorithm 2 Enumerate Reachable Set for Feat. j

```
Require: point $\boldsymbol{x}_i$
Require: action set $A_j(\boldsymbol{x}_i)$ for feature $j \in [d]$
  $R_j \leftarrow \varnothing, \quad A_j \leftarrow A(\boldsymbol{x}_i)$
  repeat
  $\boldsymbol{a}^* \leftarrow \mathsf{Find1DAction}(\boldsymbol{x}_i, A_j)$
  $R_j \leftarrow R_j \cup \{\boldsymbol{x}_i + \boldsymbol{a}^*\}$
  $A_j \leftarrow A_j \setminus \{\boldsymbol{a}^*\}$
  until $\mathsf{Find1DAction}(\boldsymbol{x}_i, A_j)$ is infeasible
Output $R_j$, set of reachable points via actions in $j$
```

3.3 Discussion

One of the benefits of working with reachable sets is that they can readily be extended to handle other desiderata by weighing and filtering reachable points. Here, we list a few variants of the responsiveness score:

- *Robust Responsiveness* [i.e. $\nu_j(\boldsymbol{x}) = \Pr(h(\boldsymbol{x} + \boldsymbol{a}) = y^{\mathrm{t}} | \boldsymbol{a}_j \neq 0,\ \boldsymbol{a} \in A(\boldsymbol{x}_i))$]: Responsiveness of feature j even as other features change [see e.g., 42]. If we are listing features for recourse, we might want them to be robust against changes in other features. This is because consumers may (inadvertently) act upon other features.
- *Cost-Weighted Responsiveness* [i.e. $\gamma_j(\boldsymbol{x};\ \mathrm{cost}) = \sum_{\boldsymbol{a} \in A_j(\boldsymbol{x}_i)} \mathrm{cost}(\boldsymbol{a}) \cdot \mathbb{1}[h(\boldsymbol{x} + \boldsymbol{a}) = y^{\mathrm{t}}]$]: Many methods that return recourse actions with the smallest cost. This suggests the "easiest" path to recourse for individuals. In a similar fashion, we can calculate responsiveness weighted by the cost of actions. However, one should note that eliciting a meaningful cost function may be challenging in practice.

Some variants will require additional quantities (i.e., we need the full reachable set to calculate robust responsiveness), while others may not have the same interpretation as the original responsiveness scores (i.e. cost-weighted responsiveness can no longer be interpreted as a proportion).

More broadly, the reachable set allows us to assess model characteristics like the monotonicity of the model with respect to a feature. This is useful because there may be instances where an individual can attain the target prediction by increasing a feature value, but can invalidate recourse by increasing it beyond a threshold. These properties are difficult to capture with one or several scores, but remain accessible through analyzing the reachable set.

4 Experiments

We present an empirical study on the responsiveness of explanations. Our goals are: (1) to evaluate how our approach can support recourse and flag fixed predictions; and (2) to demonstrate the limitations of existing feature attribution methods in practice. We include additional results and details in Appendix B, and code to reproduce these results at anonymized repository.

Setup We work with three classification datasets from consumer finance that are publicly available and used in prior work (see Appendix B for details). Here, each instance represents a consumer and each label indicates whether they will repay a loan. For each dataset, we define *inherent* *actionability constraints* that capture indisputable requirements and that apply to all individuals – e.g., no changes for immutable and protected attributes, changes must preserve feature encoding and adhere to deterministic causal effects.

We split each dataset into a training sample (80%; to train models and tune hyperparameters) and a test sample (20%; to evaluate out-of-sample performance). We train classifiers using (1) *logistic regression* (LR), (2) *XGBoost* (XGB), and (3) *random forests* (RF). For each model, we construct a feature-based explanation for each individual who is denied credit by listing the top-k highest-scoring features from the following methods:

- *Feature Responsiveness Score* (RESP): We compute the score in Definition 4 using the procedure in Section 3.2, and the actionability constraints in Appendix B.
- *Standard Feature Attribution*: We consider local feature attribution methods that are model-agnostic and widely used in the lending industry [20]: *Shapley additive explanation* (SHAP) [38]; and *local interpretable model-agnostic explanations* (LIME) [45].
- *Actionable Feature Attribution*: We also consider *action-aware* variants of feature attribution methods SHAP-AW and LIME-AW, which seek to promote responsiveness by setting the scores for immutable features to 0 such that $\phi_j(\boldsymbol{x}_i) \leftarrow 0$ when feature j is immutable.

Results We summarize the viability of promoting recourse using feature-highlighting explanations in Table 3, and the responsiveness of explanations from each method in Table 4. We evaluate explanations built using the top-4 scoring features from each method, which reflects the recommended number of reasons to include in an adverse action notice required by the U.S. Equal Credit Opportunity Act [see 2, 5].

Our results in Table 3 show that models admit features that allow *some* individuals to change them to attain a desired prediction (29.8% to 93.2% across models and datasets). At the same time, they reveal their potential to mislead individuals who are assigned fixed predictions (i.e., 0.2% to 49.1% across all models and datasets). For example, given the LR model for the heloc dataset, we would present an explanation to 56.1% of individuals who are a denied loan. Among them, 44.4% can achieve recourse through single-feature actions; 35.6% can only achieve recourse through joint actions; and 19.1% have no path to recourse because they receive a fixed prediction.

Dataset	Metrics	LR	RF	XGB
heloc	% Denied	56.1%	58.3%	57.0%
$n = 5,842$	↳ % Fixed	19.1%	28.1%	49.1%
$d = 43$ ($d_A = 31$)	↳ % 1-D Rec	44.4%	34.6%	29.8%
FICO [19]	↳ % n-D Rec	36.6%	37.4%	21.2%
german	% Denied	22.9%	17.5%	22.0%
$n = 1,000$	↳ % Fixed	7.4%	29.1%	15.5%
$d = 36$ ($d_A = 9$)	↳ % 1-D Rec	73.4%	51.4%	65.5%
Dua and Graff [13]	↳ % n-D Rec	19.2%	19.4%	19.1%
givemecredit	% Denied	24.6%	24.7%	24.8%
$n = 120,268$	↳ % Fixed	15.6%	0.2%	11.5%
$d = 23$ ($d_A = 13$)	↳ % 1-D Rec	72.4%	93.2%	76.0%
Kaggle [28]	↳ % n-D Rec	12.0%	6.6%	12.5%

Table 3: Recourse feasibility across datasets and model classes. Here, d_A is the number of mutable features. *% Denied* – the fraction of individuals denied credit by a model; *% 1-D* – the fraction of denied individuals who can achieve recourse with actions that alter a single feature; *% n-D* – the fraction of denied individuals who can only achieve recourse with actions that alter 2 or more features; and *% Fixed* – the fraction of denied individuals who are assigned a fixed prediction (in red if > 0).

On Responsiveness Scores Our results in Table 4 show how our approach can support consumers by highlightin g responsive features and by flagging instances where explanations may be misleading. Explanations are only provided to individuals who can achieve recourse through a single-feature action, and are given to *all* such individuals (the values for *% Presented with Reasons* in Table 4 match the values for *% 1-D Rec* in Table 3). When we construct feature-based explanations using responsiveness scores, we present individuals with explanations that only contain responsive features, achieving 100% on the *% All Reasons Responsive* metric across datasets and models. This may result in explanations that highlight fewer reasons on average – for example, individuals receiving explanations from the LR model on german receive 1.9 out of 4 reasons on average. This behavior can mitigate harm as we avoid presenting explanations to individuals with fixed predictions or those who require joint actions to change their outcomes.

On Feature Attribution Scores Our results show how standard methods for feature attribution can output explanations that are ineffective and potentially misleading. For example, under the LR model for the heloc dataset, we find that 82% and 75.6% of explanations from LIME and SHAP include 4/4

		LR					XGB				
		All Features		Actionable Features			All Features		Actionable Features		
Dataset	Metrics	LIME	SHAP	LIME-AW	SHAP-AW	RESP	LIME	SHAP	LIME-AW	SHAP-AW	RESP
heloc $n = 5,842$ $d = 43$ ($d_A = 31$) FICO [19]	% Presented with Explanations	100.0%	100.0%	100.0%	100.0%	44.4%	100.0%	100.0%	100.0%	100.0%	29.8%
	↳ % All Unresponsive	82.0%	75.6%	64.7%	64.7%	0.0%	92.6%	80.7%	77.5%	75.1%	0.0%
	↳ % At Least 1 Responsive	18.0%	24.4%	35.3%	35.3%	100.0%	7.4%	19.3%	22.5%	24.9%	100.0%
	↳ % All Responsive	0.0%	0.0%	0.2%	0.2%	**100.0%**	0.0%	0.0%	0.0%	0.0%	**100.0%**
	↳ Mean # of Features	4.0	4.0	4.0	4.0	2.4	4.0	4.0	4.0	4.0	2.7
german $n = 1,000$ $d = 36$ ($d_A = 9$) Dua and Graff [13]	% Presented with Explanations	100.0%	100.0%	100.0%	100.0%	73.4%	100.0%	100.0%	100.0%	100.0%	65.5%
	↳ % All Unresponsive	100.0%	100.0%	62.9%	66.4%	0.0%	100.0%	83.2%	64.5%	66.8%	0.0%
	↳ % At Least 1 Responsive	0.0%	0.0%	37.1%	33.6%	100.0%	0.0%	16.8%	35.5%	33.2%	100.0%
	↳ % All Responsive	0.0%	0.0%	0.0%	0.0%	**100.0%**	0.0%	0.0%	0.0%	0.0%	**100.0%**
	↳ Mean # of Features	4.0	4.0	4.0	4.0	1.9	4.0	4.0	4.0	4.0	2.0
givemecredit $n = 120,268$ $d = 23$ ($d_A = 13$) Kaggle [28]	% Presented with Explanations	100.0%	100.0%	100.0%	100.0%	72.4%	100.0%	100.0%	100.0%	100.0%	76.0%
	↳ % All Unresponsive	55.8%	45.5%	50.7%	31.8%	0.0%	40.9%	51.3%	30.9%	40.6%	0.0%
	↳ % At Least 1 Responsive	44.2%	54.5%	49.3%	68.2%	100.0%	59.1%	48.7%	69.1%	59.4%	100.0%
	↳ % All Responsive	0.0%	0.0%	5.5%	23.1%	**100.0%**	0.0%	0.0%	5.4%	3.7%	**100.0%**
	↳ Mean # of Features	4.0	4.0	4.0	4.0	2.4	4.0	4.0	4.0	4.0	2.6

Table 4: Responsiveness of feature-based explanations for LR and XGB models across all methods and datasets (We defer results for RF to Appendix B.5 for clarity). For each model, we generate feature-based explanations for individuals denied a loan, highlighting up to 4 top-scoring features from a given feature attribution method. For each method, we report the proportion of individuals receiving an explanation (*% Presented with Explanations*); the mean number of features per explanation (*Mean # of Features*); and the proportion of explanations that highlight only unresponsive features (*% All Unresponsive*), include at least one responsive feature (*At Least 1 Responsive*), or highlight only responsive features (*All Responsive*, in **bold**). Methods that return only unresponsive explanations are marked in red.

unresponsive features respectively. This behavior arises as a result of algorithm design, as the scores do not account for responsiveness nor actionability. This results in two key problems:

Low Scores for Responsive Features: Methods can assign low scores to responsive features. On the heloc dataset, for example, 44.4% of denied individuals by the LR model can achieve recourse by altering a single feature. However, explanations built using LIME and SHAP fail to include them since their scoring mechanisms do not account for feature responsiveness. For instance, an individual could achieve recourse by acting on NumRevolvingTrades, but a feature-based explanation from LIME does not include it, as it assigns higher scores to four other features that are unresponsive. We also observe this phenomenon beyond the top-4 features in Fig. 3.

Reasons without Recourse: Methods provide explanations to individuals with fixed predictions. On the heloc dataset, the LR model assigns a fixed prediction to 19.1% of denied individuals. In such cases, LIME and SHAP and their variants offer explanations, even though it is impossible for them to achieve recourse. These explanations may mislead individuals by highlighting features that are salient to the prediction and could be changed, but would not lead to recourse. For example, an explanation from SHAP for an individual with a fixed prediction includes AvgYearsInFile and NetFractionRevolvingBurden – both of which are mutable but not actionable.

On Adapting Existing Methods Seeing how responsiveness is inherently tied to actionability, we study the potential to improve responsiveness through *action-aware* variants of SHAP and LIME – SHAP-AW and LIME-AW. We construct explanations using only mutable features, following common a belief surrounding actionability that we can account for it through post-processing [e.g., 41, 29].

The action-aware variants show some modest improvements. For the LR model in heloc, 35.3% of explanations contained at least one responsive feature, up from SHAP's 24.4%, potentially helping more consumers achieve recourse. Fig. 3 confirms SHAP-AW ranks responsive features higher than SHAP, showing an upward shift in responsiveness distribution across ranks.

Nevertheless, SHAP-AW and LIME-AW explanations still contain unresponsive reasons. On heloc with the LR model, only 0.2% of the explanations were fully responsive. This means that 99.8% of denied applicants received explanations with at least one unresponsive feature. This occurs because LIME-AW and SHAP-AW still assign scores to unresponsive features when other responsive features exist or have exhausted the list of such features. Therefore, our results highlight that post-processing fails to properly account for actionability.

5 Concluding Remarks

Explanations are often seen as a strategy to protect individuals from harm when machine learning models are applied in domains like lending and hiring. Our work reveals how this strategy can backfire by highlighting unresponsive features and overlooking fixed predictions. We find that common feature attribution methods exhibit both of these failure modes, leading to situations where consumers are given reasons without recourse. Our work addresses these limitations by developing a feature attribution method that measures *responsiveness*—i.e., the probability that a feature can be changed in a way that leads to recourse. These scores can readily replace the scores currently used to comply with regulations. In doing so, we can strengthen consumer protection by highlighting features that enable recourse when possible and flagging instances where recourse is unattainable. Our results demonstrate the benefits of developing standalone methods to address specific goals—whether for recourse, rectification, or anti-discrimination. By adopting specialized approaches, we can achieve more effective consumer protection.

Extensions While our work focused on consumer finance and recourse, the responsiveness score has broader applications across various domains. In healthcare, it can evaluate decision models for organ transplant allocation and triage systems, where it is essential to make prompt yet fair decisions. In criminal justice, it can assess risk scoring models used in pretrial and sentencing decisions. Although "a right to recourse" does not apply in these domains, the responsiveness score serves as a valuable diagnostic tool to identify potentially harmful model behaviors.

Limitations The main limitations of our work stem from assumptions about actionability. Our approach relies on the validity of actionability assumptions within an action set. When defining this set to encode indisputable constraints, as in Section 4, responsiveness scores can flag individuals with fixed predictions. However, presented features may not achieve recourse due to individual constraints. To mitigate this, we can highlight features achieving a threshold responsiveness or elicit constraints from decision subjects [see e.g., 16, 11, 34].

Our machinery only represents a subset of constraints considered in causal algorithmic recourse literature. It can represent cases with deterministic causal effects but excludes scenarios where interventions induce probabilistic effects on downstream features [32, 12, 55]. In principle, our approach can incorporate such assumptions: given an individual probabilistic graphical model, we can compute a responsiveness score reflecting the expected recourse rate. The key challenge lies in validating causal assumptions at an individual level. This reflects a practical bottleneck that requires further study and may require an approach to measure responsiveness in a way that is robust to misspecification.

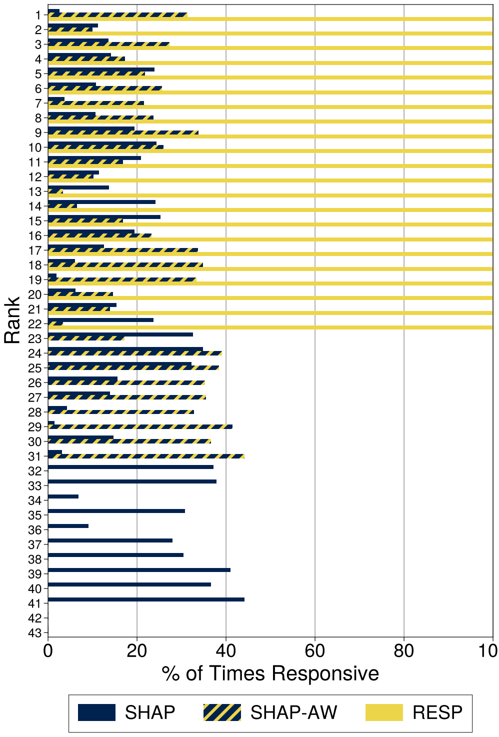


Figure 3: Responsiveness of top-scoring features for individuals who are denied credit by the LR model on the heloc dataset for using SHAP, SHAP-AW and RESP. For each method, we report the mean responsiveness of the feature with the k-th largest score – i.e. the proportion of individuals who can attain a target prediction through a single-feature action on this feature. As shown, the top features highlighted by SHAP are rarely responsive. SHAP-AW highlights more responsive features by assigning low scores to features that are immutable. We report responsiveness forindividuals who are denied (i.e., 24%), only include a if a feature receive a non-zero attribution score. We provide analogous plots for other datasets, model classes and methods in Appendix B.6.

REFERENCES

[1] 12 cfr part 1002 - equal credit opportunity act (regulation b). https://www.consumerfinance.gov/rules-policy/regulations/1002/2/, . Accessed: 2024-07-16.

[2] Comment for 1002.9 - notifications. https://www.consumerfinance.gov/rules-policy/regulations/1002/interp-9/#9-b-1-Interp-1, . Accessed: 2024-07-16.

[3] Adler, Philip, Casey Falk, Sorelle A Friedler, Tionney Nix, Gabriel Rybeck, Carlos Scheidegger, Brandon Smith, and Suresh Venkatasubramanian. Auditing black-box models for indirect influence. *Knowledge and Information Systems*, 54:95–122, 2018.

[4] Aïvodji, Ulrich, Hiromi Arai, Olivier Fortineau, Sébastien Gambs, Satoshi Hara, and Alain Tapp. Fairwashing: the risk of rationalization. In *International Conference on Machine Learning*, pages 161–170. PMLR, 2019.

[5] Barocas, Solon, Andrew D Selbst, and Manish Raghavan. The hidden assumptions behind counterfactual explanations and principal reasons. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 80–89, 2020.

[6] Bilodeau, Blair, Natasha Jaques, Pang Wei Koh, and Been Kim. Impossibility theorems for feature attribution. *Proceedings of the National Academy of Sciences*, 121(2):e2304406120, 2024.

[7] Black, Emily, Manish Raghavan, and Solon Barocas. Model multiplicity: Opportunities, concerns, and solutions. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 850–863, 2022.

[8] Bogen, Miranda and Aaron Rieke. Help wanted: An examination of hiring algorithms, equity, and bias. *Upturn, December*, 7, 2018.

[9] Brown, Lawrence D, T Tony Cai, and Anirban DasGupta. Interval estimation for a binomial proportion. *Statistical science*, 16(2):101–133, 2001.

[10] Brunet, Marc-Etienne, Ashton Anderson, and Richard Zemel. Implications of model indeterminacy for explanations of automated decisions. *Advances in Neural Information Processing Systems*, 35:7810–7823, 2022.

[11] De Toni, Giovanni, Paolo Viappiani, Stefano Teso, Bruno Lepri, and Andrea Passerini. Personalized algorithmic recourse with preference elicitation. *arXiv preprint arXiv:2205.13743*, 2022.

[12] Dominguez-Olmedo, Ricardo, Amir H Karimi, and Bernhard Schölkopf. On the adversarial robustness of causal algorithmic recourse. In *International Conference on Machine Learning*, pages 5324–5342. PMLR, 2022.

[13] Dua, Dheeru and Casey Graff. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

[14] Edwards, Lilian and Michael Veale. Slave to the algorithm: Why a right to an explanation is probably not the remedy you are looking for. *Duke L. & Tech. Rev.*, 16:18, 2017.

[15] ElShawi, Radwa, Youssef Sherif, Mouaz Al-Mallah, and Sherif Sakr. Ilime: local and global interpretable model-agnostic explainer of black-box decision. In *Advances in Databases and Information Systems: 23rd European Conference, ADBIS 2019, Bled, Slovenia, September 8–11, 2019, Proceedings 23*, pages 53–68. Springer, 2019.

[16] Esfahani, Seyedehdelaram, Giovanni De Toni, Bruno Lepri, Andrea Passerini, Katya Tentori, and Massimo Zancanaro. Exploiting preference elicitation in interactive and user-centered algorithmic recourse: An initial exploration. *arXiv preprint arXiv:2404.05270*, 2024.

[17] Eubanks, Virginia. *Automating inequality: How high-tech tools profile, police, and punish the poor*. St. Martin's Press, 2018.

[18] European Parliament, Council of the European Union. Regulation (eu) 2024/1689. https://eur-lex.europa.eu/eli/reg/2024/1689/oj. Accessed: 2024-08-30.

[19] FICO. Explainable machine learning challenge, 2018. URL https://community.fico.com/s/explainable-machine-learning-challenge.

[20] FinRegLab. Empirical white paper: Explainability and fairness: Insights from consumer lending. Technical report, FinRegLab, July 2023. URL https://finreglab.org/wp-content/uploads/2023/12/FinRegLab_2023-07-13_Empirical-White-Paper_Explainability-and-Fairness_Insights-from-Consumer-Lending.pdf.

[21] Fokkema, Hidde, Rianne De Heide, and Tim Van Erven. Attribution-based explanations that provide recourse cannot be robust. *Journal of Machine Learning Research*, 24(360):1–37, 2023.

[22] Fumagalli, Fabian, Maximilian Muschalik, Patrick Kolpaczki, Eyke Hüllermeier, and Barbara Hammer. Shap-iq: Unified approximation of any-order shapley interactions. *Advances in Neural Information Processing Systems*, 36, 2024.

[23] Galhotra, Sainyam, Romila Pradhan, and Babak Salimi. Explaining black-box algorithms using probabilistic contrastive counterfactuals. In *Proceedings of the 2021 International Conference on Management of Data*, pages 577–590, 2021.

[24] Gilman, Michele E. Poverty lawgorithms: A poverty lawyer's guide to fighting automated decision-making harms on low-income communities. *Data & Society*, 2020.

[25] Goethals, Sofie, David Martens, and Theodoros Evgeniou. Manipulation risks in explainable ai: The implications of the disagreement problem. *arXiv preprint arXiv:2306.13885*, 2023.

[26] Hurley, Mikella and Julius Adebayo. Credit scoring in the era of big data. *Yale JL & Tech.*, 18:148, 2016.

[27] Jethani, Neil, Mukund Sudarshan, Ian Connick Covert, Su-In Lee, and Rajesh Ranganath. Fastshap: Real-time shapley value estimation. In *International conference on learning representations*, 2021.

[28] Kaggle. Give Me Some Credit. http://www.kaggle.com/c/GiveMeSomeCredit/, 2011.

[29] Karimi, Amir-Hossein, Gilles Barthe, Borja Balle, and Isabel Valera. Model-agnostic counterfactual explanations for consequential decisions. In *International Conference on Artificial Intelligence and Statistics*, pages 895–905. PMLR, 2020.

[30] Karimi, Amir-Hossein, Julius Von Kügelgen, Bernhard Schölkopf, and Isabel Valera. Algorithmic recourse under imperfect causal knowledge: a probabilistic approach. *Advances in neural information processing systems*, 33:265–277, 2020.

[31] Karimi, Amir-Hossein, Gilles Barthe, Bernhard Schölkopf, and Isabel Valera. A survey of algorithmic recourse: definitions, formulations, solutions, and prospects. 2021.

[32] Karimi, Amir-Hossein, Bernhard Schölkopf, and Isabel Valera. Algorithmic recourse: from counterfactual explanations to interventions. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, pages 353–362, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383097. doi: 10.1145/3442188.3445899. URL https://doi.org/10.1145/3442188.3445899.

[33] Kaur, Harmanpreet, Harsha Nori, Samuel Jenkins, Rich Caruana, Hanna Wallach, and Jennifer Wortman Vaughan. Interpreting interpretability: understanding data scientists' use of interpretability tools for machine learning. In *Proceedings of the 2020 CHI conference on human factors in computing systems*, pages 1–14, 2020.

[34] Koh, Seunghun, Byung Hyung Kim, and Sungho Jo. Understanding the user perception and experience of interactive algorithmic recourse customization. *ACM Transactions on Computer-Human Interaction*, 2024.

[35] Kothari, Avni, Bogdan Kulynych, Tsui-Wei Weng, and Berk Ustun. Prediction without preclusion: Recourse verification with reachable sets. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=SCQfYpdoGE.

[36] Lakkaraju, Himabindu and Osbert Bastani. "how do i fool you?": Manipulating user trust via misleading black box explanations. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, AIES '20, pages 79–85, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450371100. doi: 10.1145/3375627.3375833. URL https://doi.org/10.1145/3375627.3375833.

[37] Lei, Jing, Max G'Sell, Alessandro Rinaldo, Ryan J Tibshirani, and Larry Wasserman. Distribution-free predictive inference for regression. *Journal of the American Statistical Association*, 113(523):1094–1111, 2018.

[38] Lundberg, Scott M and Su-In Lee. A unified approach to interpreting model predictions. *NeurIPS*, 2017.

[39] Marx, Charles, Richard Phillips, Sorelle Friedler, Carlos Scheidegger, and Suresh Venkatasubramanian. Disentangling influence: Using disentangled representations to audit model predictions. *Advances in Neural Information Processing Systems*, 32, 2019.

[40] Marx, Charles, Flavio Calmon, and Berk Ustun. Predictive multiplicity in classification. In *Proceedings of Machine Learning and Systems 2020*, pages 9215–9224. 2020.

[41] Mothilal, Ramaravind K, Amit Sharma, and Chenhao Tan. Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*, pages 607–617, 2020.

[42] Nguyen, Duy, Ngoc Bui, and Viet Anh Nguyen. Distributionally robust recourse action. *arXiv preprint arXiv:2302.11211*, 2023.

[43] Pawelczyk, Martin, Teresa Datta, Johan HeuvelVan den , Gjergji Kasneci, and Himabindu Lakkaraju. Probabilistically robust recourse: Navigating the trade-offs between costs and robustness in algorithmic recourse. In *The Eleventh International Conference on Learning Representations*, 2023.

[44] Raghavan, Manish, Solon Barocas, Jon Kleinberg, and Karen Levy. Mitigating bias in algorithmic hiring: Evaluating claims and practices. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*, pages 469–481, 2020.

[45] Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin. Why should I trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144. ACM, 2016.

[46] Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin. Anchors: High-precision model-agnostic explanations. In *AAAI Conference on Artificial Intelligence*, 2018.

[47] Selbst, Andrew D and Solon Barocas. The intuitive appeal of explainable machines. 2018.

[48] Shapley, Lloyd S. A value for n-person games. *Contribution to the Theory of Games*, 2, 1953.

[49] Slack, Dylan, Sophie Hilgard, Emily Jia, Sameer Singh, and Himabindu Lakkaraju. Fooling lime and shap: Adversarial attacks on post hoc explanation methods. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 180–186, 2020.

[50] Slack, Dylan, Anna Hilgard, Himabindu Lakkaraju, and Sameer Singh. Counterfactual explanations can be manipulated. *Advances in neural information processing systems*, 34:62–75, 2021.

[51] Taylor, Winnie F. Meeting the equal credit opportunity act's specificity requirement: Judgmental and statistical scoring systems. *Buff. L. Rev.*, 29:73, 1980.

[52] The Lawyers' Committee for Civil Rights Under Law. Online civil rights act, December, 2023. URL https://www.lawyerscommittee.org/online-civil-rights-act.

[53] Upadhyay, Sohini, Shalmali Joshi, and Himabindu Lakkaraju. Towards robust and reliable algorithmic recourse. *arXiv preprint arXiv:2102.13620*, 2021.

[54] Ustun, Berk, Alexander Spangher, and Yang Liu. Actionable recourse in linear classification. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, FAT* '19, pages 10–19. ACM, 2019. ISBN 978-1-4503-6125-5. doi: 10.1145/3287560.3287566.

[55] Kügelgen, Juliusvon , Amir-Hossein Karimi, Umang Bhatt, Isabel Valera, Adrian Weller, and Bernhard Schölkopf. On the fairness of causal algorithmic recourse. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 9584–9594, 2022.

[56] Watson-Daniels, Jamelle, David C. Parkes, and Berk Ustun. Predictive multiplicity in probabilistic classification. In *AAAI Conference on Artificial Intelligence*, 06 2023.

[57] White House. Blueprint for an AI bill of rights: Making automated systems work for the American people. The White House Office of Science and Technology Policy, October, 2022. URL https://www.whitehouse.gov/ostp/ai-bill-of-rights/.

[58] Wolsey, Laurence A. *Integer programming*. John Wiley & Sons, 2020.

[59] Wykstra, S. Government's use of algorithm serves up false fraud charges. undark, 6 january, 2020.

[60] Zafar, Muhammad Rehman and Naimul Mefraz Khan. Dlime: A deterministic local interpretable model-agnostic explanations approach for computer-aided diagnosis systems. *arXiv preprint arXiv:1906.10263*, 2019.

[61] Zhou, Yilun, Serena Booth, Marco Tulio Ribeiro, and Julie Shah. Do feature attribution methods correctly attribute features? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 9623–9633, 2022.

[62] Zhou, Zhengze, Giles Hooker, and Fei Wang. S-lime: Stabilized-lime for model explanation. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pages 2429–2438, 2021.

A Supporting Material for Section Section 3

A.1 MIP Formulation for Find1DAction

In what follows, we describe the implementation for the Find1DAction routine in Algorithm 2. This routine enumerates the set of valid actions by repeatedly solving a mixed-integer program.

Given a point $\boldsymbol{x}_i \in \mathcal{X}$, a feature $j \in [d]$, a single-feature action set $A_j(\boldsymbol{x}_i)$, and a set previous optima from the past L iterations $\mathcal{A}_j^{\text{opt}}$ where $|\mathcal{A}_j^{\text{opt}}| = L$, routine returns a one-dimensional action $\boldsymbol{a} \in A_j(\boldsymbol{x}_i) \setminus \mathcal{A}_j^{\text{opt}}$ by solving a mixed-integer program of the form:

$$
\begin{array}{llll r}
\min\limits_{\boldsymbol{a}} & \sum\limits_{k \in C_j} a_k^+ + a_k^- & & & \\
\text{s.t.} & a_k^+ \geq a_k & k \in C_j & \textit{positive component of } a_k & \text{(1a)} \\
 & a_k^- \geq -a_k & k \in C_j & \textit{negative component of } a_k & \text{(1b)} \\
 & a_k = a_{k,l} + \delta_{k,l}^+ - \delta_{k,l}^- & k \in C_j, \boldsymbol{a}_l \in \mathcal{A}_j^{\text{opt}} & \textit{distance from prior actions} & \text{(1c)} \\
 & \varepsilon_{\text{min}} \leq \sum\limits_{j \in C_j} (\delta_{k,l}^+ + \delta_{k,l}^-) & \boldsymbol{a}_l \in \mathcal{A}_j^{\text{opt}} & \textit{any solution is } \varepsilon_{min} \textit{ away from } \boldsymbol{a}_l & \text{(1d)} \\
 & \delta_{k,l}^+ \leq M_{k,l}^+ u_{k,l}^+ & k \in C_j, \boldsymbol{a}_l \in \mathcal{A}_j^{\text{opt}} & \delta_{k,l}^+ > 0 \implies u_{k,l}^+ = 1 & \text{(1e)} \\
 & \delta_{k,l}^- \leq M_{k,l}^- u_{k,l}^- & k \in C_j, \boldsymbol{a}_l \in \mathcal{A}_j^{\text{opt}} & \delta_{k,l}^- > 0 \implies u_{k,l}^- = 1 & \text{(1f)} \\
 & u_{k,l} = u_{k,l}^+ + u_{k,l}^- & k \in C_j, l \in [L] & & \text{(1g)} \\
 & u_{j,l} = 1 & k \in C_j, \boldsymbol{a}_l \in \mathcal{A}_j^{\text{opt}} & \textit{must change feature } j & \text{(1h)} \\
 & a_k \in A'_k(\boldsymbol{x}_i) & k \in C_j & \textit{separable actionability constraints on } k & \text{(1i)} \\
 & \boldsymbol{a} \in A_j(\boldsymbol{x}_i) & & \textit{joint actionability constraints on } j & \text{(1j)} \\
 & a_k^+, a_k^- \in \mathbb{R}_+ & k \in C_j & \textit{absolute value of } a_k & \text{(1k)} \\
 & \delta_{k,l}^+, \delta_{k,l}^- \in \mathbb{R}_+ & k \in C_j & \textit{signed distances from } a_{k,l} & \text{(1l)} \\
 & u_{k,l}, u_{k,l}^+, u_{k,l}^- \in \{0, 1\} & k \in C_j & & \text{(1m)}
\end{array}
$$

For each feature $k \in C_j$, the formulation separates the absolute value into a_k^+ and a_k^- to convert the optimization problem into a linear one.

We ensure that the solution is not already in $\mathcal{A}_j^{\text{opt}}$ – at least ε_{min} away. By default $\varepsilon_{\text{min}} = 10^{-6}$, but is set to $\varepsilon_{\text{min}} = 1$ for discrete feature sets.

This formulation is designed to output single-feature actions from an action set in the set $\boldsymbol{a} \in A_j(\boldsymbol{x}_i) \setminus \mathcal{A}_j^{\text{opt}}$. It contains two kinds of constraints: (i) constraints to ensure the actionability of changes $\boldsymbol{a} \in A_j(\boldsymbol{x}_i)$ and (ii) constraints to rule out actions in $\boldsymbol{a} \in \mathcal{A}_j^{\text{opt}}$. The formulation represents a special case of the optimization problem presented in Kothari et al. [35] – where we are explicitly enforcing actions to only alter single feature interventions.

A.2 MIP Formulation for CheckFeasibility

We describe the implementation for the CheckFeasibility$(\boldsymbol{a}^*, A_j)$ in Algorithm 1. Contrary to the MIP formulation in Appendix A.1, given the original point $\boldsymbol{x}_i \in \mathcal{X}$ and the sampled action $\boldsymbol{a}^*$, we solve the MIP once.

The formulation is a variant of the problem in Appendix A.1, where:

- $\boldsymbol{a} = \boldsymbol{a}^*$,
- $\mathcal{A}_j^{\text{opt}} = \varnothing$,
- and set the objective to $\min_{\boldsymbol{a}} \; 1$

Hence CheckFeasibility$(\boldsymbol{a}^*, A_j) = 1$ if $\boldsymbol{a}^*$ is feasible under actionability constraints and 0 otherwise.

B Supplementary Experiment Results

B.1 Description and Actionability Constraints for the `heloc` Dataset

Description The FICO dataset was created to predict repayment on Home Equity Line of Credit (HELOC) applications. HELOC credit lines are loans that use people's homes as collateral. The dataset is used by lenders to determine how much credit should be granted. The anonymized version of the HELOC dataset was created by FICO to present an explainable machine learning challenge for a prize.

Each instance in the dataset is a real credit application for HELOC credit; it's an application that a single person submitted and contains information about that person. There are $n = 10,459$ instances, each consisting of $d = 23$ features. These features are either binary or discrete. The label, `RiskPerformance`, is a binary assessment of the risk of repayment based on the 23 predictors. A value of 1 means the person hasn't been more than 90 days overdue on their payments in the last 2 years; a value of 0 means they have at least once. There are some repeated instances; there are 9,871 unique rows. The dataset is self-contained and has been anonymized for public use in the explainability challenge. It doesn't use any protected attributes like race and gender.

Actionability Constraints The joint actionability constraints include:

1. DirectionalLinkage: Actions on `NumRevolvingTradesWBalance≥2` will induce to actions on ['`NumRevolvingTrades≥2`']. Each unit change in `NumRevolvingTradesWBalance≥2` leads to:1.00-unit change in `NumRevolvingTrades≥2`
2. DirectionalLinkage: Actions on `NumInstallTradesWBalance≥2` will induce to actions on ['`NumInstallTrades≥2`']. Each unit change in `NumInstallTradesWBalance≥2` leads to:1.00-unit change in `NumInstallTrades≥2`
3. DirectionalLinkage: Actions on `NumRevolvingTradesWBalance≥3` will induce to actions on ['`NumRevolvingTrades≥3`']. Each unit change in `NumRevolvingTradesWBalance≥3` leads to:1.00-unit change in `NumRevolvingTrades≥3`
4. DirectionalLinkage: Actions on `NumInstallTradesWBalance≥3` will induce to actions on ['`NumInstallTrades≥3`']. Each unit change in `NumInstallTradesWBalance≥3` leads to:1.00-unit change in `NumInstallTrades≥3`
5. DirectionalLinkage: Actions on `NumRevolvingTradesWBalance≥5` will induce to actions on ['`NumRevolvingTrades≥5`']. Each unit change in `NumRevolvingTradesWBalance≥5` leads to:1.00-unit change in `NumRevolvingTrades≥5`
6. DirectionalLinkage: Actions on `NumInstallTradesWBalance≥5` will induce to actions on ['`NumInstallTrades≥5`']. Each unit change in `NumInstallTradesWBalance≥5` leads to:1.00-unit change in `NumInstallTrades≥5`
7. DirectionalLinkage: Actions on `NumRevolvingTradesWBalance≥7` will induce to actions on ['`NumRevolvingTrades≥7`']. Each unit change in `NumRevolvingTradesWBalance≥7` leads to:1.00-unit change in `NumRevolvingTrades≥7`
8. DirectionalLinkage: Actions on `NumInstallTradesWBalance≥7` will induce to actions on ['`NumInstallTrades≥7`']. Each unit change in `NumInstallTradesWBalance≥7` leads to:1.00-unit change in `NumInstallTrades≥7`
9. DirectionalLinkage: Actions on `YearsSinceLastDelqTrade≤1` will induce to actions on ['`YearsOfAccountHistory`']. Each unit change in `YearsSinceLastDelqTrade≤1` leads to:-1.00-unit change in `YearsOfAccountHistory`
10. DirectionalLinkage: Actions on `YearsSinceLastDelqTrade≤3` will induce to actions on ['`YearsOfAccountHistory`']. Each unit change in `YearsSinceLastDelqTrade≤3` leads to:-3.00-unit change in `YearsOfAccountHistory`
11. DirectionalLinkage: Actions on `YearsSinceLastDelqTrade≤5` will induce to actions on ['`YearsOfAccountHistory`']. Each unit change in `YearsSinceLastDelqTrade≤5` leads to:-5.00-unit change in `YearsOfAccountHistory`

Name	Type	LB	UB	mutability
ExternalRiskEstimate_geq_40	$\{0,1\}$	0	1	no
ExternalRiskEstimate_geq_50	$\{0,1\}$	0	1	no
ExternalRiskEstimate_geq_60	$\{0,1\}$	0	1	no
ExternalRiskEstimate_geq_70	$\{0,1\}$	0	1	no
ExternalRiskEstimate_geq_80	$\{0,1\}$	0	1	no
YearsOfAccountHistory	$\mathbb{Z}$	0	50	no
AvgYearsInFile_geq_3	$\{0,1\}$	0	1	only increases
AvgYearsInFile_geq_5	$\{0,1\}$	0	1	only increases
AvgYearsInFile_geq_7	$\{0,1\}$	0	1	only increases
MostRecentTradeWithinLastYear	$\{0,1\}$	0	1	yes
MostRecentTradeWithinLast2Years	$\{0,1\}$	0	1	yes
AnyDerogatoryComment	$\{0,1\}$	0	1	no
AnyTrade120DaysDelq	$\{0,1\}$	0	1	no
AnyTrade90DaysDelq	$\{0,1\}$	0	1	no
AnyTrade60DaysDelq	$\{0,1\}$	0	1	no
AnyTrade30DaysDelq	$\{0,1\}$	0	1	no
NoDelqEver	$\{0,1\}$	0	1	no
YearsSinceLastDelqTrade_leq_1	$\{0,1\}$	0	1	only increases
YearsSinceLastDelqTrade_leq_3	$\{0,1\}$	0	1	only increases
YearsSinceLastDelqTrade_leq_5	$\{0,1\}$	0	1	only increases
NumInstallTrades_geq_2	$\{0,1\}$	0	1	only increases
NumInstallTradesWBalance_geq_2	$\{0,1\}$	0	1	only increases
NumRevolvingTrades_geq_2	$\{0,1\}$	0	1	only increases
NumRevolvingTradesWBalance_geq_2	$\{0,1\}$	0	1	only increases
NumInstallTrades_geq_3	$\{0,1\}$	0	1	only increases
NumInstallTradesWBalance_geq_3	$\{0,1\}$	0	1	only increases
NumRevolvingTrades_geq_3	$\{0,1\}$	0	1	only increases
NumRevolvingTradesWBalance_geq_3	$\{0,1\}$	0	1	only increases
NumInstallTrades_geq_5	$\{0,1\}$	0	1	only increases
NumInstallTradesWBalance_geq_5	$\{0,1\}$	0	1	only increases
NumRevolvingTrades_geq_5	$\{0,1\}$	0	1	only increases
NumRevolvingTradesWBalance_geq_5	$\{0,1\}$	0	1	only increases
NumInstallTrades_geq_7	$\{0,1\}$	0	1	only increases
NumInstallTradesWBalance_geq_7	$\{0,1\}$	0	1	only increases
NumRevolvingTrades_geq_7	$\{0,1\}$	0	1	only increases
NumRevolvingTradesWBalance_geq_7	$\{0,1\}$	0	1	only increases
NetFractionInstallBurden_geq_10	$\{0,1\}$	0	1	only increases
NetFractionInstallBurden_geq_20	$\{0,1\}$	0	1	only increases
NetFractionInstallBurden_geq_50	$\{0,1\}$	0	1	only increases
NetFractionRevolvingBurden_geq_10	$\{0,1\}$	0	1	only increases
NetFractionRevolvingBurden_geq_20	$\{0,1\}$	0	1	only increases
NetFractionRevolvingBurden_geq_50	$\{0,1\}$	0	1	only increases
NumBank2NatlTradesWHighUtilizationGeq2	$\{0,1\}$	0	1	only increases

Table 5: Table of Separable Actionability Constraints for the `heloc` dataset. Includes bounds and monotonicity constraints.

12. ReachabilityConstraint: The values of [`MostRecentTradeWithinLastYear`, `MostRecentTradeWithinLast2Years`] must belong to one of 4 values with custom reachability conditions.

13. ThermometerEncoding: Actions on [`YearsSinceLastDelqTrade≤1`, `YearsSinceLastDelqTrade≤3`, `YearsSinceLastDelqTrade≤5`] must preserve thermometer encoding of YearsSinceLastDelqTradeleq., which can only decrease. Actions can only turn off higher-level dummies that are on, where `YearsSinceLastDelqTrade≤1` is the lowest-level dummy and `YearsSinceLastDelqTrade≤5` is the highest-level-dummy.

14. ThermometerEncoding: Actions on [`AvgYearsInFile≥3`, `AvgYearsInFile≥5`, `AvgYearsInFile≥7`] must preserve thermometer encoding of AvgYearsInFilegeq., which can only increase. Actions can only turn on higher-level dummies that are off, where

`AvgYearsInFile≥3` is the lowest-level dummy and `AvgYearsInFile≥7` is the highest-level-dummy.

15. ThermometerEncoding: Actions on [`NetFractionRevolvingBurden≥10`, `NetFractionRevolvingBurden≥20`, `NetFractionRevolvingBurden≥50`] must preserve thermometer encoding of NetFractionRevolvingBurdengeq., which can only decrease. Actions can only turn off higher-level dummies that are on, where `NetFractionRevolvingBurden≥10` is the lowest-level dummy and `NetFractionRevolvingBurden≥50` is the highest-level-dummy.
16. ThermometerEncoding: Actions on [`NetFractionInstallBurden≥10`, `NetFractionInstallBurden≥20`, `NetFractionInstallBurden≥50`] must preserve thermometer encoding of NetFractionInstallBurdengeq., which can only decrease. Actions can only turn off higher-level dummies that are on, where `NetFractionInstallBurden≥10` is the lowest-level dummy and `NetFractionInstallBurden≥50` is the highest-level-dummy.
17. ThermometerEncoding: Actions on [`NumRevolvingTradesWBalance≥2`, `NumRevolvingTradesWBalance≥3`, `NumRevolvingTradesWBalance≥5`, `NumRevolvingTradesWBalance≥7`] must preserve thermometer encoding of NumRevolvingTradesWBalancegeq., which can only decrease. Actions can only turn off higher-level dummies that are on, where `NumRevolvingTradesWBalance≥2` is the lowest-level dummy and `NumRevolvingTradesWBalance≥7` is the highest-level-dummy.
18. ThermometerEncoding: Actions on [`NumRevolvingTrades≥2`, `NumRevolvingTrades≥3`, `NumRevolvingTrades≥5`, `NumRevolvingTrades≥7`] must preserve thermometer encoding of NumRevolvingTradesgeq., which can only decrease. Actions can only turn off higher-level dummies that are on, where `NumRevolvingTrades≥2` is the lowest-level dummy and `NumRevolvingTrades≥7` is the highest-level-dummy.
19. ThermometerEncoding: Actions on [`NumInstallTradesWBalance≥2`, `NumInstallTradesWBalance≥3`, `NumInstallTradesWBalance≥5`, `NumInstallTradesWBalance≥7`] must preserve thermometer encoding of NumInstallTradesWBalancegeq., which can only decrease. Actions can only turn off higher-level dummies that are on, where `NumInstallTradesWBalance≥2` is the lowest-level dummy and `NumInstallTradesWBalance≥7` is the highest-level-dummy.
20. ThermometerEncoding: Actions on [`NumInstallTrades≥2`, `NumInstallTrades≥3`, `NumInstallTrades≥5`, `NumInstallTrades≥7`] must preserve thermometer encoding of NumInstallTradesgeq., which can only decrease. Actions can only turn off higher-level dummies that are on, where `NumInstallTrades≥2` is the lowest-level dummy and `NumInstallTrades≥7` is the highest-level-dummy.

B.2 Description and Actionability Constraints for the `german` Dataset

Description The `german` dataset was created in 1994 and contains information about loan history, demographics, occupation, payment history, and whether or not somebody is a good customer.

Each instance is credit applicant. There are $n = 1,000$ instances, each consisting of $d = 20$ features. The features are all either categorical or discrete. The label a binary indicator of whether somebody is a "good" ($y_i = 1$) or "bad" ($y_i = 2$) applicant. We changed these labels to be 0 and 1.

There are no missing values in the dataset. We renamed some of the features to be indicative of the values they represent. The dataset is self-contained and anonymous, and it includes features describing gender, age, and marital status.

Name	Type	LB	UB	Actionability	Sign
`Age`	$\mathbb{Z}$	19	75	No	
`Male`	$\{0,1\}$	0	1	No	
`Single`	$\{0,1\}$	0	1	No	
`ForeignWorker`	$\{0,1\}$	0	1	No	
`YearsAtResidence`	$\mathbb{Z}$	0	7	Yes	$+$
`LiablePersons`	$\mathbb{Z}$	1	2	No	
`Housing=Renter`	$\{0,1\}$	0	1	No	
`Housing=Owner`	$\{0,1\}$	0	1	No	
`Housing=Free`	$\{0,1\}$	0	1	No	
`Job=Unskilled`	$\{0,1\}$	0	1	No	
`Job=Skilled`	$\{0,1\}$	0	1	No	
`Job=Management`	$\{0,1\}$	0	1	No	
`YearsEmployed≥1`	$\{0,1\}$	0	1	Yes	$+$
`CreditAmt≥1000K`	$\{0,1\}$	0	1	No	
`CreditAmt≥2000K`	$\{0,1\}$	0	1	No	
`CreditAmt≥5000K`	$\{0,1\}$	0	1	No	
`CreditAmt≥10000K`	$\{0,1\}$	0	1	No	
`LoanDuration≤6`	$\{0,1\}$	0	1	No	
`LoanDuration≥12`	$\{0,1\}$	0	1	No	
`LoanDuration≥24`	$\{0,1\}$	0	1	No	
`LoanDuration≥36`	$\{0,1\}$	0	1	No	
`LoanRate`	$\mathbb{Z}$	1	4	No	
`HasGuarantor`	$\{0,1\}$	0	1	Yes	$+$
`LoanRequiredForBusiness`	$\{0,1\}$	0	1	No	
`LoanRequiredForEducation`	$\{0,1\}$	0	1	No	
`LoanRequiredForCar`	$\{0,1\}$	0	1	No	
`LoanRequiredForHome`	$\{0,1\}$	0	1	No	
`NoCreditHistory`	$\{0,1\}$	0	1	No	
`HistoryOfLatePayments`	$\{0,1\}$	0	1	No	
`HistoryOfDelinquency`	$\{0,1\}$	0	1	No	
`HistoryOfBankInstallments`	$\{0,1\}$	0	1	Yes	$+$
`HistoryOfStoreInstallments`	$\{0,1\}$	0	1	Yes	$+$
`CheckingAcct_exists`	$\{0,1\}$	0	1	Yes	$+$
`CheckingAcct≥0`	$\{0,1\}$	0	1	Yes	$+$
`SavingsAcct_exists`	$\{0,1\}$	0	1	Yes	$+$
`SavingsAcct≥100`	$\{0,1\}$	0	1	Yes	$+$

Table 6: Table of Separable Actionability Constraints for the `german` dataset. Includes bounds and monotonicity constraints.

Actionability Constraints The joint actionability constraints include

1. DirectionalLinkage: Actions on `YearsAtResidence` will induce to actions on ['`Age`']. Each unit change in `YearsAtResidence` leads to:1.00-unit change in `Age`
2. DirectionalLinkage: Actions on `YearsEmployed≥1` will induce to actions on ['`Age`']. Each unit change in `YearsEmployed≥1` leads to:1.00-unit change in `Age`

3. ThermometerEncoding: Actions on [`CheckingAcctexists`, `CheckingAcct≥0`] must preserve thermometer encoding of CheckingAcct., which can only increase. Actions can only turn on higher-level dummies that are off, where `CheckingAcctexists` is the lowest-level dummy and `CheckingAcct≥0` is the highest-level-dummy.

4. ThermometerEncoding: Actions on [`SavingsAcctexists`, `SavingsAcct≥100`] must preserve thermometer encoding of SavingsAcct., which can only increase. Actions can only turn on higher-level dummies that are off, where `SavingsAcctexists` is the lowest-level dummy and `SavingsAcct≥100` is the highest-level-dummy.

B.3 Description and Actionability Constraints for the givemecredit Dataset

Description The givemecredit dataset is used to determine whether a loan should be given or denied. The label indicates whether someone was 90 days past due in the two years following data collection. Delinquency refers to a debt with an overdue payment; this dataset is used to predict if someone will experience financial distress in the next two years.

It contains information about $n = 120,268$ loan recipients, and each instance represents a borrower. There are $d = 10$ features before preprocessing. The label is SeriousDlqin2yrs, meaning serious delinquency in two years. In preprocessing, we change the label to NotSeriousDlqin2yrs so that $y_i = 1$ is a positive classification and $y_i = 0$ is negative.

The data is self-contained and anonymous, and it contains features describing age, income, and the number of dependents.

Name	Type	LB	UB	mutability
Age_leq_24	$\{0,1\}$	0	1	no
Age_bt_25_to_30	$\{0,1\}$	0	1	no
Age_bt_30_to_59	$\{0,1\}$	0	1	no
Age_geq_60	$\{0,1\}$	0	1	no
NumberOfDependents_eq_0	$\{0,1\}$	0	1	no
NumberOfDependents_eq_1	$\{0,1\}$	0	1	no
NumberOfDependents_geq_2	$\{0,1\}$	0	1	no
NumberOfDependents_geq_5	$\{0,1\}$	0	1	no
DebtRatio_geq_1	$\{0,1\}$	0	1	only increases
MonthlyIncome_geq_3K	$\{0,1\}$	0	1	only increases
MonthlyIncome_geq_5K	$\{0,1\}$	0	1	only increases
MonthlyIncome_geq_10K	$\{0,1\}$	0	1	only increases
CreditLineUtilization_geq_10.0	$\{0,1\}$	0	1	yes
CreditLineUtilization_geq_20.0	$\{0,1\}$	0	1	yes
CreditLineUtilization_geq_50.0	$\{0,1\}$	0	1	yes
CreditLineUtilization_geq_70.0	$\{0,1\}$	0	1	yes
CreditLineUtilization_geq_100.0	$\{0,1\}$	0	1	yes
AnyRealEstateLoans	$\{0,1\}$	0	1	only increases
MultipleRealEstateLoans	$\{0,1\}$	0	1	only increases
AnyCreditLinesAndLoans	$\{0,1\}$	0	1	only increases
MultipleCreditLinesAndLoans	$\{0,1\}$	0	1	only increases
HistoryOfLatePayment	$\{0,1\}$	0	1	no
HistoryOfDelinquency	$\{0,1\}$	0	1	no

Table 7: Table of Separable Actionability Constraints for the givemecredit dataset. Includes bounds and monotonicity constraints.

Actionability Constraints The joint actionability constraints include

1. ThermometerEncoding: Actions on [MonthlyIncome≥3K, MonthlyIncome≥5K, MonthlyIncome≥10K] must preserve thermometer encoding of MonthlyIncomegeq., which can only increase. Actions can only turn on higher-level dummies that are off, where MonthlyIncome≥3K is the lowest-level dummy and MonthlyIncome≥10K is the highest-level-dummy.
2. ThermometerEncoding: Actions on [CreditLineUtilization≥10.0, CreditLineUtilization≥20.0, CreditLineUtilization≥50.0, CreditLineUtilization≥70.0, CreditLineUtilization≥100.0] must preserve thermometer encoding of CreditLineUtilizationgeq., which can only decrease. Actions can only turn off higher-level dummies that are on, where CreditLineUtilization≥10.0 is the lowest-level dummy and CreditLineUtilization≥100.0 is the highest-level-dummy.

3. ThermometerEncoding: Actions on [`AnyRealEstateLoans`, `MultipleRealEstateLoans`] must preserve thermometer encoding of continuousattribute., which can only decrease. Actions can only turn off higher-level dummies that are on, where `AnyRealEstateLoans` is the lowest-level dummy and `MultipleRealEstateLoans` is the highest-level-dummy.
4. ThermometerEncoding: Actions on [`AnyCreditLinesAndLoans`, `MultipleCreditLinesAndLoans`] must preserve thermometer encoding of continuousattribute., which can only decrease. Actions can only turn off higher-level dummies that are on, where `AnyCreditLinesAndLoans` is the lowest-level dummy and `MultipleCreditLinesAndLoans` is the highest-level-dummy.

B.4 Overview of Model Performance

	LR		XGB		RF	
Dataset	Train	Test	Train	Test	Train	Test
heloc $n = 5,842$ $d = 43$ ($d_A = 31$) FICO [19]	0.772	0.788	0.859	0.785	0.780	0.790
german $n = 1,000$ $d = 36$ ($d_A = 9$) Dua and Graff [13]	0.819	0.760	0.971	0.794	0.828	0.766
givemecredit $n = 120,268$ $d = 23$ ($d_A = 13$) Kaggle [28]	0.841	0.844	0.875	0.793	0.864	0.835

Table 8: Train and Test AUC for models across all datasets. We optimized the model's hyperparameters through randomized search and divided the data into training and testing sets at an 80% and 20% ratio.

B.5 Responsiveness of Explanations for RF Models

		RF				
		All Features		Actionable Features		
Dataset	Metrics	LIME	SHAP	LIME	SHAP	RESP
heloc $n = 5,842$ $d = 43$ ($d_A = 31$) FICO [19]	% Presented with Explanations	100.0%	100.0%	100.0%	100.0%	34.6%
	↳ % All Unresponsive	85.1%	78.2%	74.1%	74.4%	0.0%
	↳ % At Least 1 Responsive	14.9%	21.8%	25.9%	25.6%	100.0%
	↳ % All Responsive	0.0%	0.0%	0.0%	0.0%	**100.0%**
	↳ Mean # of Features	4.0	4.0	4.0	4.0	2.5
german $n = 1,000$ $d = 36$ ($d_A = 9$) Dua and Graff [13]	% Presented with Explanations	100.0%	100.0%	100.0%	100.0%	51.4%
	↳ % All Unresponsive	100.0%	87.4%	71.4%	60.0%	0.0%
	↳ % At Least 1 Responsive	0.0%	12.6%	28.6%	40.0%	100.0%
	↳ % All Responsive	0.0%	0.0%	0.0%	0.0%	**100.0%**
	↳ Mean # of Features	4.0	4.0	4.0	4.0	2.5
givemecredit $n = 120,268$ $d = 23$ ($d_A = 13$) Kaggle [28]	% Presented with Explanations	100.0%	100.0%	100.0%	100.0%	93.2%
	↳ % All Unresponsive	60.0%	39.6%	28.7%	17.6%	0.0%
	↳ % At Least 1 Responsive	40.0%	60.4%	71.3%	82.4%	100.0%
	↳ % All Responsive	0.0%	0.0%	0.8%	12.7%	**100.0%**
	↳ Mean # of Features	4.0	4.0	4.0	4.0	2.9

Table 9: Responsiveness of feature-based explanations for RF models for all methods and all datasets. Given a model, we construct an explanation for each individuals who are denied a loan using the top-4 scoring features from a specific feature attribution method. We report: *% Presented with Explanations*, the proportion of individuals who receive an explanation; *Mean # of Features*, the number of features in each explanation; and *% All Unresponsive / At Least 1 Responsive / All Responsive*, the proportion of explanations where all features are unresponsive/at least 1 feature is responsive/all features are responsive. For each dataset and model class, we show the approach that provides the most responsive explanations in **bold**, and highlight instances where all explanations are unresponsive in red.

B.6 Responsiveness of Top-Scoring Features

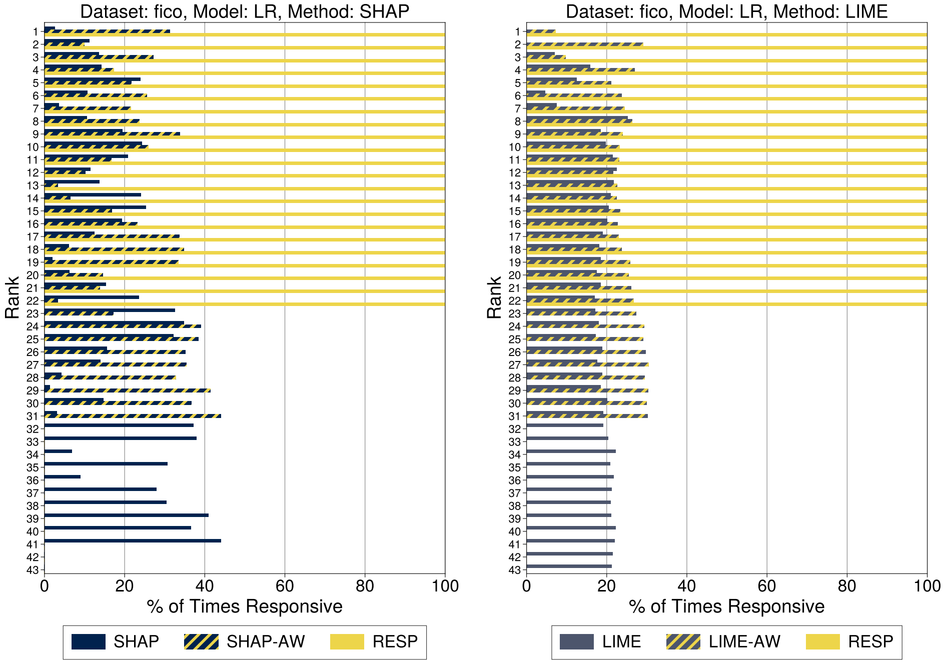


Figure 4: The percent of times where the feature at the shown rank from LIME, LIME-AW, SHAP, SHAP-AW and RESP is responsive – i.e. has at least one single-feature action that leads to recourse – for denied individuals. Only features with a non-zero score under the feature attribution method are shown. Individuals who receive a score of zero do not appear in the chart.

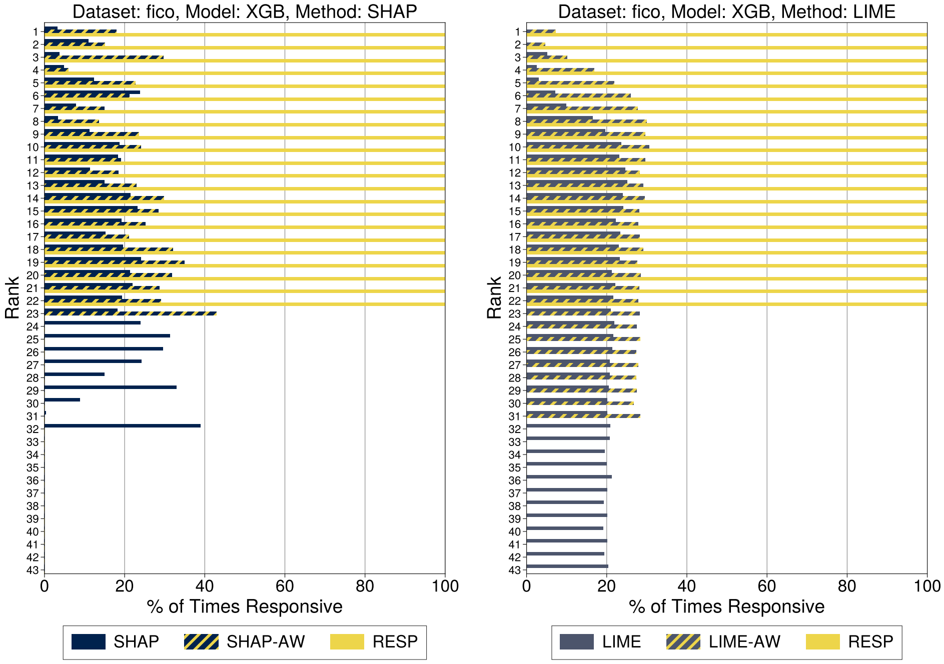


Figure 5: The percent of times where the feature at the shown rank from LIME, LIME-AW, SHAP, SHAP-AW and RESP is responsive – i.e. has at least one single-feature action that leads to recourse – for denied individuals. Only features with a non-zero score under the feature attribution method are shown. Individuals who receive a score of zero do not appear in the chart.

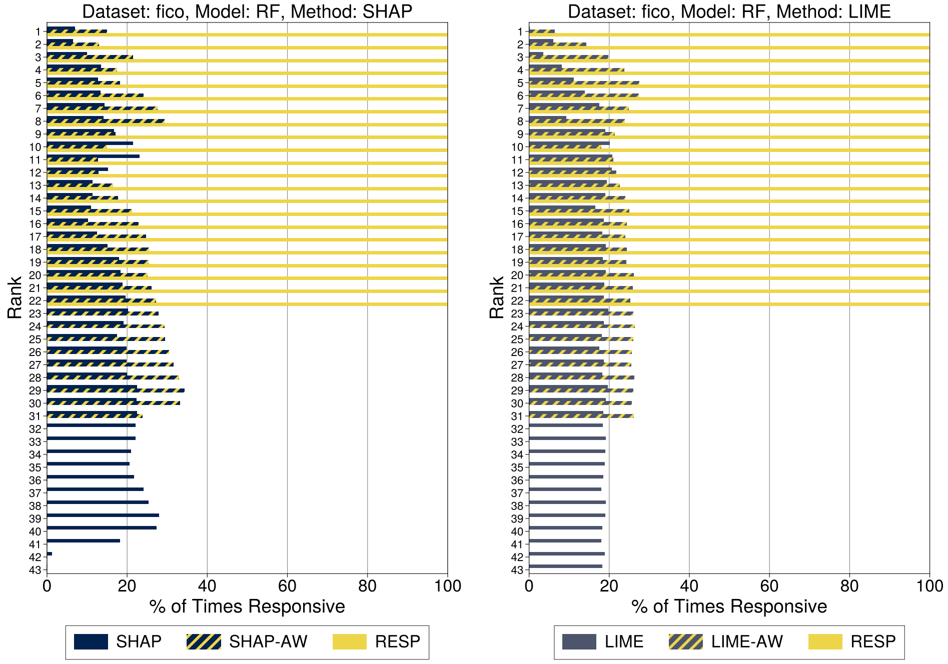


Figure 6: The percent of times where the feature at the shown rank from LIME, LIME-AW, SHAP, SHAP-AW and RESP is responsive – i.e. has at least one single-feature action that leads to recourse – for denied individuals. Only features with a non-zero score under the feature attribution method are shown. Individuals who receive a score of zero do not appear in the chart.

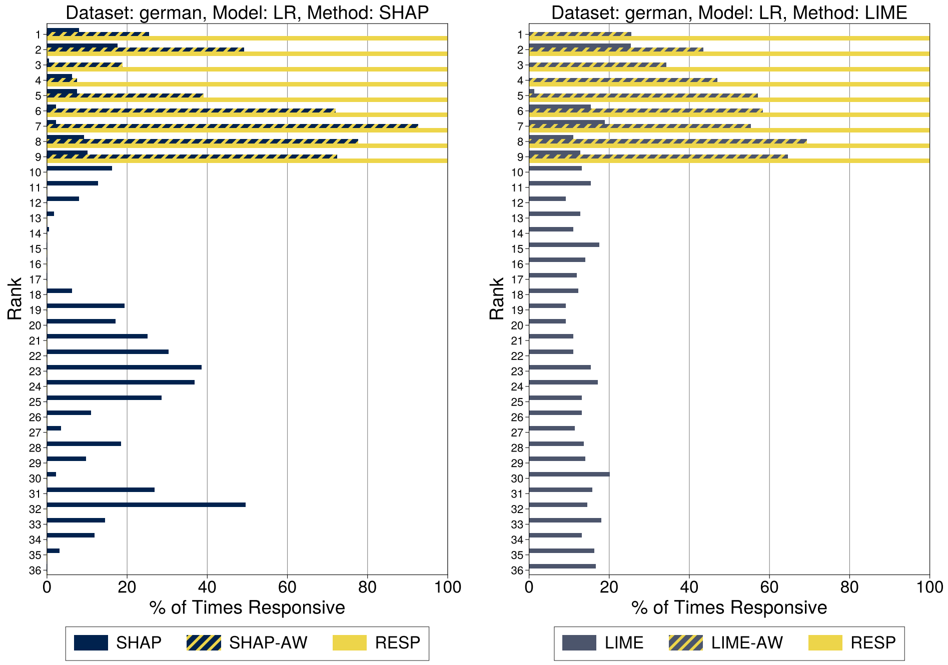


Figure 7: The percent of times where the feature at the shown rank from LIME, LIME-AW, SHAP, SHAP-AW and RESP is responsive – i.e. has at least one single-feature action that leads to recourse – for denied individuals. Only features with a non-zero score under the feature attribution method are shown. Individuals who receive a score of zero do not appear in the chart.

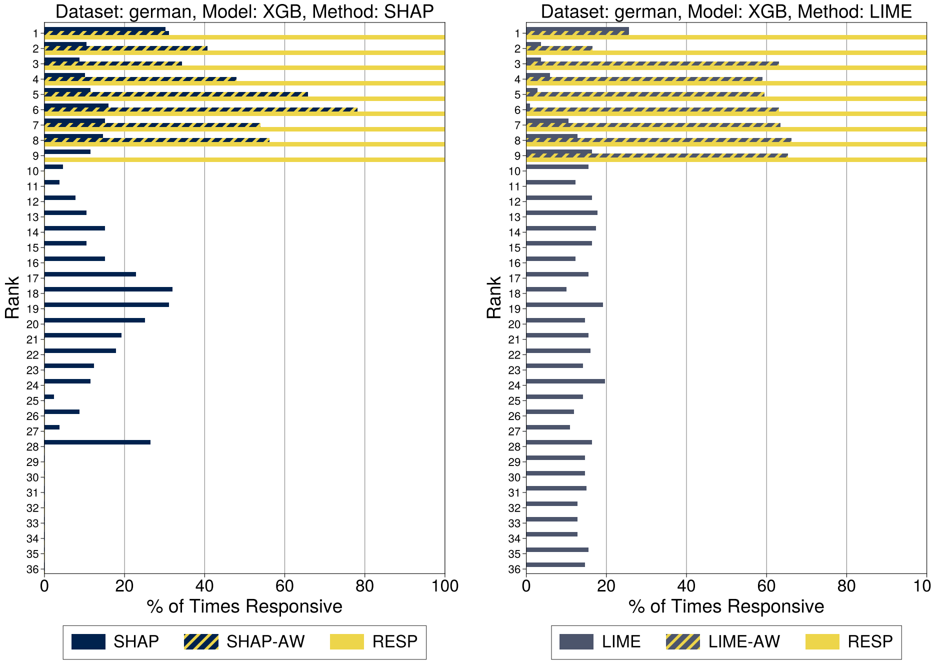


Figure 8: The percent of times where the feature at the shown rank from LIME, LIME-AW, SHAP, SHAP-AW and RESP is responsive – i.e. has at least one single-feature action that leads to recourse – for denied individuals. Only features with a non-zero score under the feature attribution method are shown. Individuals who receive a score of zero do not appear in the chart.

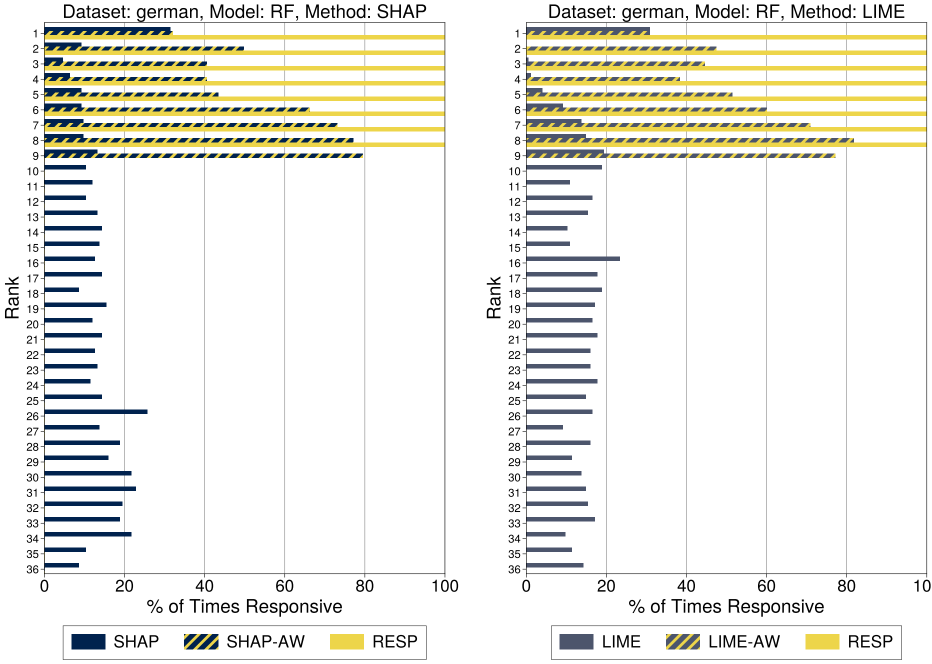


Figure 9: The percent of times where the feature at the shown rank from LIME, LIME-AW, SHAP, SHAP-AW and RESP is responsive – i.e. has at least one single-feature action that leads to recourse – for denied individuals. Only features with a non-zero score under the feature attribution method are shown. Individuals who receive a score of zero do not appear in the chart.

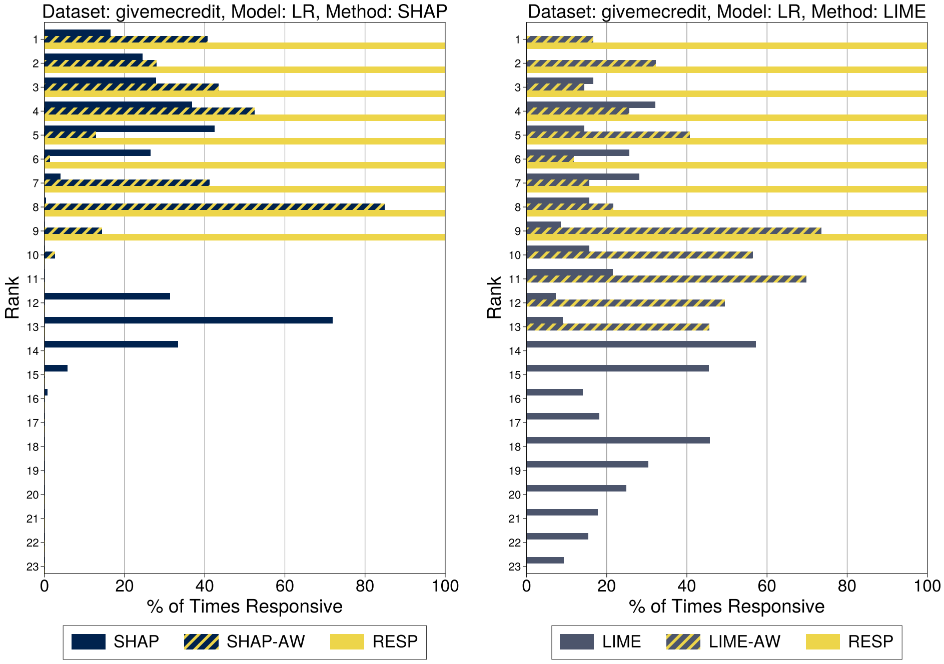


Figure 10: The percent of times where the feature at the shown rank from LIME, LIME-AW, SHAP, SHAP-AW and RESP is responsive – i.e. has at least one single-feature action that leads to recourse – for denied individuals. Only features with a non-zero score under the feature attribution method are shown. Individuals who receive a score of zero do not appear in the chart.

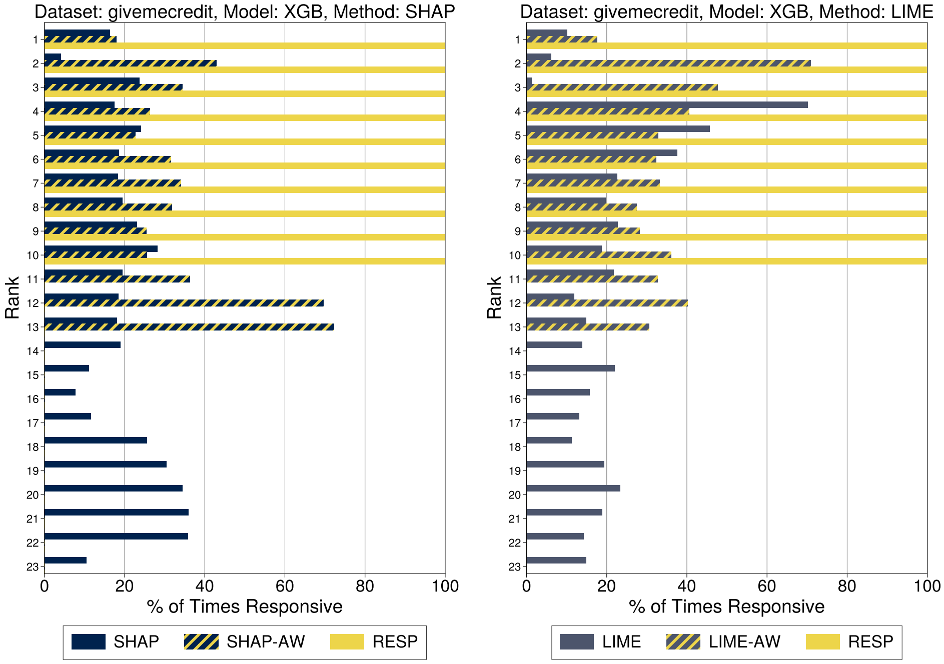


Figure 11: The percent of times where the feature at the shown rank from LIME, LIME-AW, SHAP, SHAP-AW and RESP is responsive – i.e. has at least one single-feature action that leads to recourse – for denied individuals. Only features with a non-zero score under the feature attribution method are shown. Individuals who receive a score of zero do not appear in the chart.

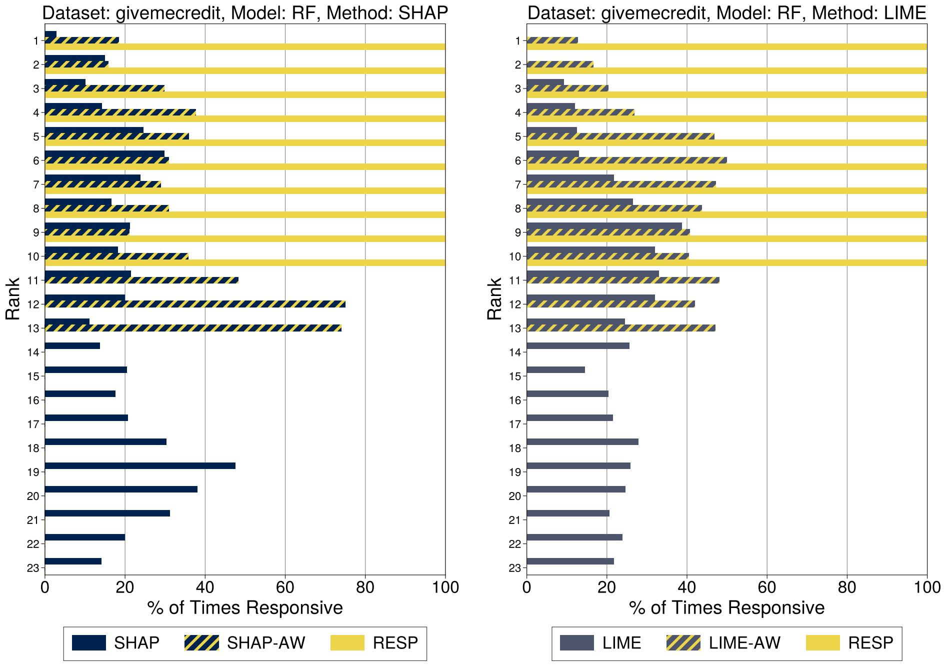


Figure 12: The percent of times where the feature at the shown rank from LIME, LIME-AW, SHAP, SHAP-AW and RESP is responsive – i.e. has at least one single-feature action that leads to recourse – for denied individuals. Only features with a non-zero score under the feature attribution method are shown. Individuals who receive a score of zero do not appear in the chart.