Feature Responsiveness Scores: Model-Agnostic Explanations for Recourse

Seung Hyun Cheon[1], Anneke Wernerfelt[2], Sorelle A. Friedler[2], and Berk Ustun[1]

[1]University of California, San Diego
[2]Haverford College

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 a list of "principal reasons" to consumers who receive adverse decisions. In practice, lenders and employers identify principal reasons by returning 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 they can attain a desired outcome. We show that standard attribution methods can mislead individuals by highlighting *reasons without recourse* – i.e., by presenting consumers with features that cannot be changed to achieve recourse. We propose to address these issues by scoring features on the basis of *responsiveness* – i.e., the probability that an individual can attain a desired outcome by changing a specific feature. We develop efficient methods to compute responsiveness scores for any model and any dataset under complex actionability constraints. We present an extensive empirical study on the responsiveness of explanations in lending, and demonstrate how responsiveness scores can be used to construct feature-highlighting explanations that lead to recourse and to 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 [9, 46], consumer finance [27], and public services [18, 25, 62]. 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 [6, 49, 54, 60]. Existing and proposed laws and regulations include provisions that require lenders or employers to provide explanations to individuals in such situations [1, 19, 54, 60]. 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 [19] grants individuals a right to obtain explanations to describe the "main elements" of their decision in high-risk applications regarding employment, education, financial systems, government benefits, law enforcement, and border control [see Annex III 19, for a definition of "high risk"].

The use of explanations in such settings reflects widespread beliefs about the effectiveness of transparency for consumer protection [15] – i.e., that revealing information can protect and empower consumers [49]. For example, the adverse action notice is motivated by the fact that presenting consumers with "principal reasons" can: (1) promote *anti-discrimination* by highlighting potential bias; (2) facilitate *rectification*, by allowing individuals to identify and correct data errors, and; (3) support *recourse* by educating individuals on how to improve future outcomes [53]. Regulators provide model deployers with substantial flexibility in complying with these requirements. In practice, model owners comply with these regulations using feature attribution methods such as LIME and SHAP [21]. These are general-purpose methods that can explain the predictions of a model after it

2nd Workshop on Regulatable ML at NeurIPS 2024.

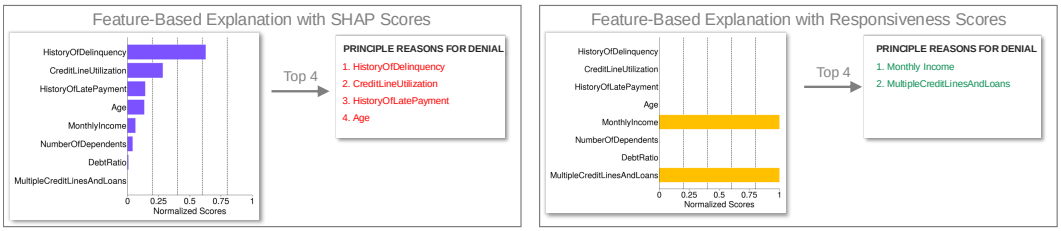

Figure 1: Feature-highlighting explanations for an individual denied credit by a logistic regression model for a lending task (see `heloc`, Section 4). We show explanations from top-scoring features using SHAP [40] (left) and responsiveness scores (right). As shown, SHAP highlights features immutable (`Age`, `HistoryOfLatePayment`, and `HistoryOfDeliquency`) or unresponsive (`CreditLineUtilization`). In contrast, explanations build using responsiveness score (right) only two features that provide recourse for the individual.

has been trained and generate explanations that can be communicated to consumers. These methods output scores that reflect the importance of each feature for a given prediction. Given these scores, model deployers can retrieve the top scoring features and present them to consumers as a list of "principal reasons" or "main elements" (see Fig. 1).

In this work, we study how to explain model predictions in a way that can achieve 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 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 probability of changing the prediction of a model by intervening on a given feature. Our approach highlights features that can be changed to receive a better model outcome, and identifies instances that may lead to harm.
2. We develop model-agnostic methods to compute feature responsiveness scores using *reachable sets*. Our methods can evaluate scores for any model, paired with theoretical guarantees that support our ability to flag harm, and 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.

Related Work Our work is related to a stream of research on post-hoc explanations [4, 39, 40, 41, 47, 48, 64]. We focus on methods for feature attribution, which are designed to evaluate the importance of feature for a given prediction. Many methods are built for use cases in model development [e.g., 3, 34, 37], but are now used to construct "feature-highlighting explanations" to comply with regulations on explanations in consumer applications [see e.g., 6, 21].

Our work shows how feature attribution methods can inflict harm in such cases by highlighting *reasons without recourse* – i.e., by highlighting features, when acted upon, do not change the prediction – sometimes to consumers who are assigned fixed predictions. This is a failure mode that affects a broad class of local explainability techniques – adding to a growing literature on how local explanations are prone to manipulation [e.g., 5, 26, 38, 51, 52], and indeterminate [e.g., due to multiplicity 8, 11, 42, 59]. Our work complements impossibility results on recourse and feature attribution showing that attribution methods that satisfy completeness and linearity (e.g., SHAP) perform no better than random guessing when inferring model behavior (e.g., recourse) [see e.g., 7, 22]. Here, we establish the prevalence of this effect empirically and develop a principled approach to mitigate it.

Our approach is related to a stream of work on algorithmic recourse [31, 57]. The vast majority of work on this topic develops algorithms for recourse provision – i.e., to present consumers with *actions* that can change the prediction a specific model [see e.g., 32]. Our goal is to highlight features

that can be reliably changed to achieve recourse. To this end, responsiveness scores measure the number of actions on a *single* feature. Our approach builds on a line of work that elicits and enforces complex actionability constraints [36, 57]. Here, we use this machinery to represent actionability constraints at an instance level, and to generate a set of all points that a person can reach under a set of actionability constraints [36]. Our approach outputs feature responsiveness scores that can be used with any model, and can be adapted to address practical challenges in providing recourse – e.g., robustness [44, 45, 56] and causality [13, 24, 33].

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 $y \in \mathcal{Y} = \{0, 1\}$ 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 given a model $h : \mathcal{X} \to \mathcal{Y}$ where each instance represents a person, and their features $\boldsymbol{x}_i \in \mathcal{X}$ encode semantically meaningful characteristics for the task at hand (e.g., `age` or `income`). We assume that the 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 sufficiently large. [1]

We consider a task where we provide explanations to individuals who are adversely affected by the prediction of a given model (see e.g., [1, 55]). We assume that $h(\boldsymbol{x}_i) = 1$ represents a *target prediction* that is desirable – e.g., $h(\boldsymbol{x}_i) = 1$ if applicant i is predicted to repay their loan within 2 years – and thus will explain the predictions for individuals where $h(\boldsymbol{x}_i) = 0$.

Feature-Highlighting Explanations Our goal is to construct explanations where each feature is *responsive* – i.e., can be changed independently to achieve recourse.

The standard practice of explaining predictions is to use *feature-highlighting explanations* [see e.g. 6]. These explanations consist of a list of "most important" features using a given method that we convert into a natural language description [e.g., a reason code 21].

Feature Attribution Methods The standard approach to construct feature-highlighting explanation is to use feature attribution method [21].

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 capture the behavior of several methods that are used to explain the prediction of a model in terms of its features:

- *Local Linear Explainers* [see e.g., 16, 47, 63, 65]: 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., 23, 28, 40]: 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 basic axioms of social choice [50].

Given a model h and its training dataset $\mathcal{D}$, the scores $\boldsymbol{\phi}(\boldsymbol{x}_i)$ capture how each feature captures the prediction of a model at the $\boldsymbol{x}_i$ in different ways. In all cases, the 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 40].
- *Strength*: Features with larger attribution scores have 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'.

[1] This assumption holds for most semantically meaningful features [see 57]. Some features have bounds by construction (i.e. binary features). In other cases, we can set loose bounds (e.g., for `income`).

These properties allow model developers to comply with consumer protection rules, but can promote misinterpretation among consumers [34].

Reasons without Recourse One failure mode of machine learning in consumer-facing applications is that models can assign *fixed predictions* – i.e., predictions that cannot be changed by their decision subjects (see e.g., Table 1). In lending, for example, fixed predictions inflict harm through *preclusion* – i.e., permanently barring consumers from access to credit. Fixed predictions are a recently-discovered failure mode [36], which must be mitigated through changes in feature construction, model development, or implementation (e.g.,using a separate model for re-applicants who are assigned fixed predictions). In practice, these instances are often left undetected. We can mitigate harm if instead of providing misleading explanations of fixed predictions they are flagged to model owners or auditors.

Table 1: Stylized lending task where the best model assigns fixed predictions. We predict $y \in \{0, 1\} =$ `repayment` from two binary features $(x_1, x_2) =$ (`age` ≥ 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` ≥ 60 is immutable and `has_IRA` is binary.

Features		Label Counts		Best Model
`age` ≥ 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

These issues stem from an oversight of *actionability* – how we can change the features of a model. On the one hand, models assign fixed predictions because they use features that can only be changed in specific ways. On the other, we are unable to detect these instances through feature attribution methods because they are designed to explain a prediction, rather than how it can be changed.

Accounting for Actionability Given these challenges, we introduce machinery to capture how features can be changed at the instance level. For example, a change in one feature might necessitate a change in another; this makes strictly independent changes to certain features infeasible.

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*. We can elicit complex constraints from human experts in natural language, and convert them into equations that we can embed into an optimization problem (for examples, see Table 4 in Appendix A). In this way, we can enforce actionability in – for example – algorithms to find recourse actions [see e.g., 36, 57].

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 $x_j = 1 \to 0 \implies x'_j = 0 \to 1$).

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:

$$\mu_j(\boldsymbol{x}_i \mid h, A(\boldsymbol{x}_i)) := \Pr(h(\boldsymbol{x}') = 1 \mid \boldsymbol{x}' = \boldsymbol{x}_i + \boldsymbol{a}, \boldsymbol{a} \in A_j(\boldsymbol{x}_i))$$

The responsive score for a feature j captures the proportion of single-feature actions on feature j that change the prediction of a model h at $\boldsymbol{x}_i$. In what follows, we write $\mu_j(\boldsymbol{x})$ instead of $\mu_j(\boldsymbol{x} \mid h, A(\boldsymbol{x}_i))$ 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 single-feature actions 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 represent an upper bound on responsiveness: "at most $100\mu_j(\boldsymbol{x}_i)\%$ of single-feature actions on feature j attain a desired prediction." Such constraints let us flag undeniable instances of harm. More generally, actionability constraints encode information about how other features are expected to vary when a single feature is changed. For example, if a model has a feature indicating the `job rank` of an individual, we can create actionability constraints that encode the expectation that if job rank increases, so does `income`.

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 $\mu_1(\boldsymbol{x}_i), \ldots, \mu_d(\boldsymbol{x}_i)$ denote the responsiveness scores of $\boldsymbol{x}_i \in \mathcal{X}$ with respect to the action set $A(\boldsymbol{x}_i)$.*

S1 If $\mu_j(\boldsymbol{x}_i) > 0$ for some feature $j \in [d]$, then h can provide recourse to $\boldsymbol{x}_i$ through a single-feature action on j.

S2 If $\mu_j(\boldsymbol{x}_i) = 0$ for all features $j \in [d]$, then either: (a) h assigns a fixed prediction to $\boldsymbol{x}_i$, or (b) h can only provide recourse to $\boldsymbol{x}_i$ through actions that alter two or more features.

Remark 1 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 1 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 (a) – 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 (b) – where a person is assigned predictions that can change through multiple actions – we could provide explanations that highlight subsets of responsive features, include explicit warning against presumptions of feature independence, or proceed in a similar manner to case (a).

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: $R_j(\boldsymbol{x}_i) := \{\boldsymbol{x}_i + \boldsymbol{a}' \mid \boldsymbol{a}' \in A_j(\boldsymbol{x}_i)\}$.

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

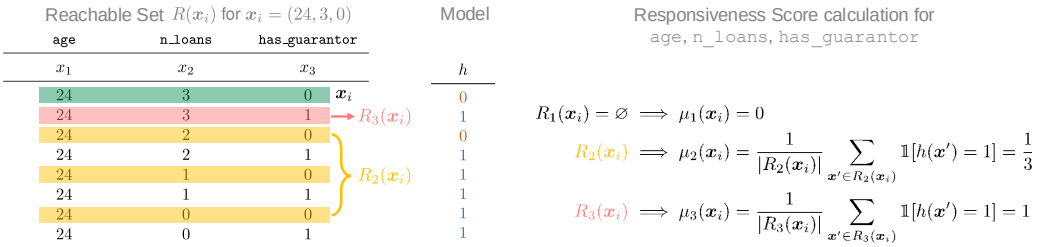

Figure 2: Simple example of how to compute responsiveness scores involving three independent features. `age` is an immutable feature, `n_loans` is a discrete feature taking values from 0 to 3 and `has_guarantor` is a binary feature. The original prediction of 0 is shown in the row highlighted in green. Single-feature actions for `n_loans`, `has_guarantor` are highlighted yellow and red respectively. We can see that we can construct a complete reachable set and directly calculate the responsiveness score for discrete datasets. The responsiveness score for `age` is 0 since it is immutable.

vectors that can be reached through feasible actions. Given reachable sets for each feature $R_j(\boldsymbol{x}_i)$ for $j \in [d]$, we can calculate responsiveness scores (see Definition 4) for a model by querying its predictions (see Fig. 2). This has the benefits that: (1) it can work with any model; (2) we only need to compute the reachable set once; and (3) it can allow us to evaluate other notions of responsiveness.

Enumeration for Discrete Features When dealing with discrete feature spaces, we can obtain a complete reachable set through enumeration following an approach of Kothari et al. [36]. Given a point $\boldsymbol{x}_i$ and its action set $A(\boldsymbol{x}_i)$, this approach returns a set of all reachable points. Given a complete reachable set, we can calculate the responsiveness score of feature j by evaluating the model h on points reachable via single-feature actions on j (see Fig. 2). This approach can certify when an individual cannot change the prediction of a model. Unfortunately, the reachable set grows exponentially with the number of actionable features, which can lead to practical challenges in storage and compute.[2]

Sampling for Continuous Features The enumeration technique described above is infeasible when we wish to evaluate the responsiveness of a continuous feature, or when a feature has downstream effects on continuous features. In such cases, we estimate the responsiveness score for feature j by drawing a uniform sample of reachable points from $R_j(\boldsymbol{x}_i)$. Given a feature without downstream effects – i.e., without downstream features so that $|C_j| = 0$ – we can sample values from $[l_j, u_j]$. When features have downstream effects, we can apply a rejection sampling procedure where we first sample values for all features from $[l_j, u_j]$ and reject values that violate actionability constraints.

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

$$\hat{\mu}_j(\boldsymbol{x}_i) := \frac{1}{N} \sum_{\boldsymbol{x}' \in \hat{R}_j(\boldsymbol{x}_i)} \mathbb{1}[h(\boldsymbol{x}') = 1]$$

When the samples are drawn uniformly from $R_j(\boldsymbol{x}_i)$, the number of responsive predictions $S \sim \text{Bin}(N, \mu_j(\boldsymbol{x}_i))$. We can then quantify the uncertainty associated with our estimate $\hat{\mu}_j(\boldsymbol{x}_i)$:

Proposition 2. *Given a level of significance $\alpha \in (0, 1)$, the $100(1-\alpha)\%$ confidence interval for $\mu_j(\boldsymbol{x}_i)$ is:*

$$\tilde{\mu}_j(\boldsymbol{x}_i) \pm \kappa \sqrt{\frac{1}{\tilde{N}} \tilde{\mu}_j(\boldsymbol{x}_i)(1 - \tilde{\mu}_j(\boldsymbol{x}_i))}$$

[2] Kothari et al. [36] propose to enumerate reachable sets for subsets of features that can be altered independently. In practice, this can allow construction of reachable set for real-world classifcation datasets with discrete features. However, it may require considerable compute and storage. For example, on the `heloc` dataset which contains 31 actionable features and 5,616 distinct points, enumerating all reachable sets requires roughly 1 hour and storing them requires 18.8GB

Here: $\kappa := \Phi^{-1}(1 - \frac{\alpha}{2})$, $\Phi(\cdot)$ *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.*

We can alter two parameters to adjust the uncertainty: the level of significance α and the sample size N. For a fixed N, a larger α will yield a narrower confidence interval at a lower confidence level. Similarly, with a fixed α, we can adjust the N to attain a desired level of certainty for the estimate $\hat{\mu}_j(\boldsymbol{x}_i)$. For example, given $\alpha = 0.05$, $N > 42$ would ensure that the width of the confidence interval surrounding $\hat{\mu}_j(\boldsymbol{x}_i)$ is at most 0.1 when we don't observe any points with the target prediction – i.e. $S = 0$. Note that we use a correction upon the standard binomial confidence interval formulation to improve coverage when $\mu_j(\boldsymbol{x}_i) = 0$ or 1[see 10].

This is a general-purpose approach that can compute responsiveness scores for features that are continuous or discrete. In settings where we are computing the responsiveness score of a discrete feature, we can reap the benefits of responsiveness using a small sample of reachable points. This approach avoids the computation and storage costs of enumeration but sacrifices the ability to identify individuals with fixed predictions with complete certainty.

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 Appendices B and C, and code to reproduce these results at the project 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 (see Appendix B).

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 [21]: *Shapley additive explanation* (SHAP) [40]; and *local interpretable model-agnostic explanations* (LIME) [47].
- *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.

We summarize the viability of promoting recourse using feature-highlighting explanations in Table 2, and the responsiveness of explanations from each method in Table 3. 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, 6].

Our results in Table 2 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.

Results

On Responsiveness Scores Our results in Table 3 show how our approach can support consumers by presenting 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 3 match the values for *% 1-D Rec* in Table 2). 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.

Table 2: Recourse feasibility across datasets and model classes. *% 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 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).

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 [20]	↳ % 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 [14]	↳ % 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 [29]	↳ % n-D Rec	12.0%	6.6%	12.5%

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 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 produced by 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 **??**.

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`.

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 consider a simple post-processing strategy where we only construct explanations using features that are mutable. This reflects a common belief surrounding actionability is that we can find ways to account for it by post-processing [e.g., 30, 43]. Our results show this approach can improve the responsiveness of explanations – as we observe marginal improvements when switching from SHAP and LIME to SHAP-AW and LIME-AW for all models and datasets in Table 3. For example, when we provide explanations for LR model on `heloc`, switching from SHAP to SHAP-AW improves the proportion of explanations that contain at least one responsive feature from 24.4% to 35.3%. As a result, more consumers could achieve recourse based on explanations with at least one responsive feature. Nevertheless, SHAP-AW and LIME-AW explanations still contain unresponsive reasons. In the `heloc` dataset, SHAP-AW and LIME-AW returned explanations where

Table 3: Responsiveness of feature-based explanations for LR and XGB models across all methods and datasets (We defer results for RF to Appendix C.2 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.

		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 [20]	% 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 [14]	% 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 [29]	% 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

every reason is responsive 0.2% of the time under the LR model. In other words, 98% of the explanations given to denied consumers contained at least one unresponsive feature. This occurs because LIME-AW and SHAP-AW still suffer from the same drawbacks as their original counterparts, albeit to a lesser extent. They attribute scores to features that are not responsive when there are other responsive features or have exhausted the list of such features. Overall, these results show that they still fall short in providing responsive explanations.

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.

Limitations The main limitations of our work stem from assumptions about actionability and responsiveness. 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., 12, 17, 35]. A broader limitation is that 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 [13, 33, 58]. 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 robustness to misspecifcation.

Acknowledgements

This work is supported by the National Science Foundation (NSF) under grant IIS-2313105.

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] Adebayo, Julius, Michael Muelly, Ilaria Liccardi, and Been Kim. Debugging tests for model explanations. *arXiv preprint arXiv:2011.05429*, 2020.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

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

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

[11] 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.

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

[13] 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.

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

[15] 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.

[16] 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.

[17] 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.

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

[19] 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.

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

[21] 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.

[22] 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.

[23] 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.

[24] 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.

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

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

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

[28] 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.

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

[30] 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.

[31] 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.

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

[33] 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.

[34] 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.

[35] 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.

[36] 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.

[37] Kulesza, Todd, Margaret Burnett, Weng-Keen Wong, and Simone Stumpf. Principles of explanatory debugging to personalize interactive machine learning. In *Proceedings of the 20th international conference on intelligent user interfaces*, pages 126–137, 2015.

[38] 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.

[39] 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.

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

[41] 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.

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

[43] 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.

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

[45] 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.

[46] 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.

[47] 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.

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

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

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

[51] 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.

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

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

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

[55] Union, European. Regulation (eu) 2024/1689: Artificial intelligence act. https://www.aiact-info.eu/article-86-right-to-explanation-of-individual-decision-making/, 2024. Accessed: 2024-08-01.

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

[57] 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.

[58] 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.

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

[60] 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/.

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

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

[63] 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.

[64] 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.

[65] 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 Example of Actionability Constraints

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_status` $\in \{\texttt{own}, \texttt{rent}, \texttt{other}\}$	$x_k = \texttt{housing\_status=own}$ $x_l = \texttt{housing\_status=rent}$ $x_m = \texttt{housing\_status=other}$	$a_j + x_j \in \{0, 1\}$ 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_account_balance` ≥ 0 else `savings_account_balance` $= 0$	$x_j = \texttt{has\_savings\_account}$ $x_k = \texttt{savings\_account\_balance}$	$a_j + x_j \in \{0, 1\}$ $a_k + x_k \in [0, 10^{12}]$ $a_j + x_j \leq 10^{12}(x_k + a_k)$
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 4: 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., 61].

B Datasets and Actionability Constraints

B.1 `heloc`

B.1.1 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.

B.1.2 Actionability Constraints

Joint Actionability Constraints:

1. DirectionalLinkage: Actions on `NumRevolvingTradesWBalance`$\geq$`2` will induce to actions on ['`NumRevolvingTrades`$\geq$`2`']. Each unit change in `NumRevolvingTradesWBalance`$\geq$`2` leads to:1.00-unit change in `NumRevolvingTrades`$\geq$`2`
2. DirectionalLinkage: Actions on `NumInstallTradesWBalance`$\geq$`2` will induce to actions on ['`NumInstallTrades`$\geq$`2`']. Each unit change in `NumInstallTradesWBalance`$\geq$`2` leads to:1.00-unit change in `NumInstallTrades`$\geq$`2`
3. DirectionalLinkage: Actions on `NumRevolvingTradesWBalance`$\geq$`3` will induce to actions on ['`NumRevolvingTrades`$\geq$`3`']. Each unit change in `NumRevolvingTradesWBalance`$\geq$`3` leads to:1.00-unit change in `NumRevolvingTrades`$\geq$`3`
4. DirectionalLinkage: Actions on `NumInstallTradesWBalance`$\geq$`3` will induce to actions on ['`NumInstallTrades`$\geq$`3`']. Each unit change in `NumInstallTradesWBalance`$\geq$`3` leads to:1.00-unit change in `NumInstallTrades`$\geq$`3`
5. DirectionalLinkage: Actions on `NumRevolvingTradesWBalance`$\geq$`5` will induce to actions on ['`NumRevolvingTrades`$\geq$`5`']. Each unit change in `NumRevolvingTradesWBalance`$\geq$`5` leads to:1.00-unit change in `NumRevolvingTrades`$\geq$`5`

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.

6. DirectionalLinkage: Actions on `NumInstallTradesWBalance`$\geq$`5` will induce to actions on ['`NumInstallTrades`$\geq$`5`']. Each unit change in `NumInstallTradesWBalance`$\geq$`5` leads to:1.00-unit change in `NumInstallTrades`$\geq$`5`
7. DirectionalLinkage: Actions on `NumRevolvingTradesWBalance`$\geq$`7` will induce to actions on ['`NumRevolvingTrades`$\geq$`7`']. Each unit change in `NumRevolvingTradesWBalance`$\geq$`7` leads to:1.00-unit change in `NumRevolvingTrades`$\geq$`7`
8. DirectionalLinkage: Actions on `NumInstallTradesWBalance`$\geq$`7` will induce to actions on ['`NumInstallTrades`$\geq$`7`']. Each unit change in `NumInstallTradesWBalance`$\geq$`7` leads to:1.00-unit change in `NumInstallTrades`$\geq$`7`
9. DirectionalLinkage: Actions on `YearsSinceLastDelqTrade`$\leq$`1` will induce to actions on ['`YearsOfAccountHistory`']. Each unit change in `YearsSinceLastDelqTrade`$\leq$`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
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 `german`

B.2.1 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 a real person with credit. There are $n = 1,000$ instances, each consisting of $d = 20$ features. The features are all either categorical or discrete. The label, class, is a binary indicator of whether somebody is a 'good' ($y_i = 1$) or 'bad' ($y_i = 2$) customer. 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.

B.2.2 Actionability Constraints

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$\geq$1	$\{0,1\}$	0	1	Yes	$+$
CreditAmt$\geq$1000K	$\{0,1\}$	0	1	No	
CreditAmt$\geq$2000K	$\{0,1\}$	0	1	No	
CreditAmt$\geq$5000K	$\{0,1\}$	0	1	No	
CreditAmt$\geq$10000K	$\{0,1\}$	0	1	No	
LoanDuration$\leq$6	$\{0,1\}$	0	1	No	
LoanDuration$\geq$12	$\{0,1\}$	0	1	No	
LoanDuration$\geq$24	$\{0,1\}$	0	1	No	
LoanDuration$\geq$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$\geq$0	$\{0,1\}$	0	1	Yes	$+$
SavingsAcct_exists	$\{0,1\}$	0	1	Yes	$+$
SavingsAcct$\geq$100	$\{0,1\}$	0	1	Yes	$+$

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

Joint Actionability Constraints:

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`$\geq$`1` will induce to actions on ['`Age`']. Each unit change in `YearsEmployed`$\geq$`1` leads to:1.00-unit change in `Age`
3. ThermometerEncoding: Actions on [`CheckingAcctexists`, `CheckingAcct`$\geq$`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`$\geq$`0` is the highest-level-dummy.
4. ThermometerEncoding: Actions on [`SavingsAcctexists`, `SavingsAcct`$\geq$`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`$\geq$`100` is the highest-level-dummy.

B.3 `givemecredit`

B.3.1 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.

B.3.2 Actionability Constraints

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.

Joint Actionability Constraints:

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`$\geq$`10.0` is the lowest-level dummy and `CreditLineUtilization`$\geq$`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.

C Supplementary Experiment Results

C.1 Overview of Model Performance

Dataset	LR Train	LR Test	XGB Train	XGB Test	RF Train	RF Test
`heloc` $n = 5,842$ $d = 43$ ($d_A = 31$) FICO [20]	0.772	0.788	0.859	0.785	0.780	0.790
`german` $n = 1,000$ $d = 36$ ($d_A = 9$) Dua and Graff [14]	0.819	0.760	0.971	0.794	0.828	0.766
`givemecredit` $n = 120,268$ $d = 23$ ($d_A = 13$) Kaggle [29]	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.

C.2 Responsiveness of Explanations for RF Models

Dataset	Metrics	RF: All Features LIME	RF: All Features SHAP	RF: Actionable Features LIME	RF: Actionable Features SHAP	RF: RESP
`heloc` $n = 5,842$ $d = 43$ ($d_A = 31$) FICO [20]	% 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 [14]	% 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 [29]	% 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.

D Additional Plots

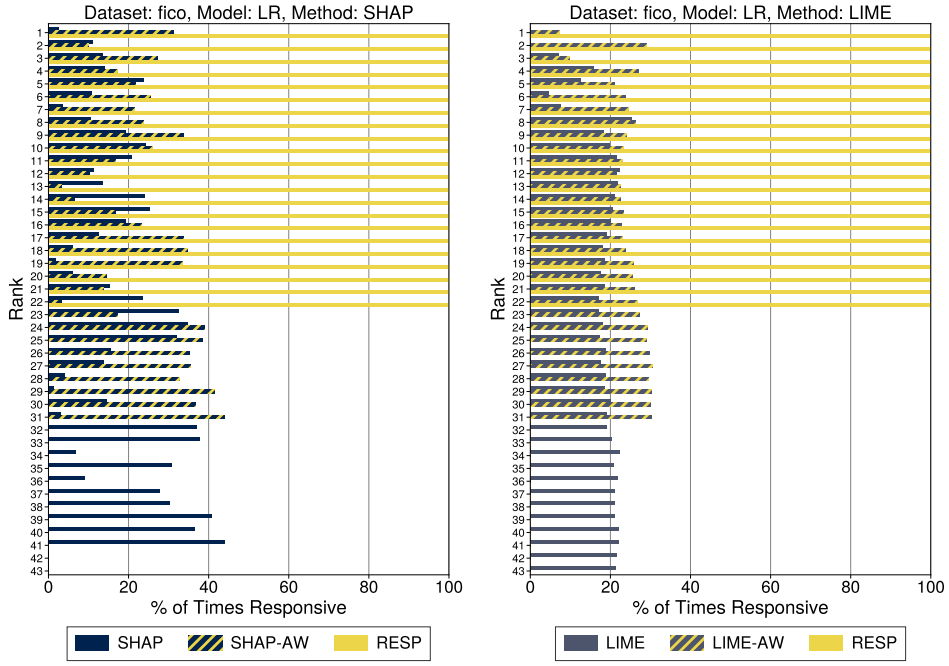

Figure 3: The percent of times where the feature at the shown rank from LIME, LIME-AW, SHAP, SHAP-AW and RESP is responsive – i.e. has at least one single-feature action that leads to recourse – for denied individuals. Only features with a non-zero score under the feature attribution method are shown. Individuals who receive a score of zero do not appear in the chart.

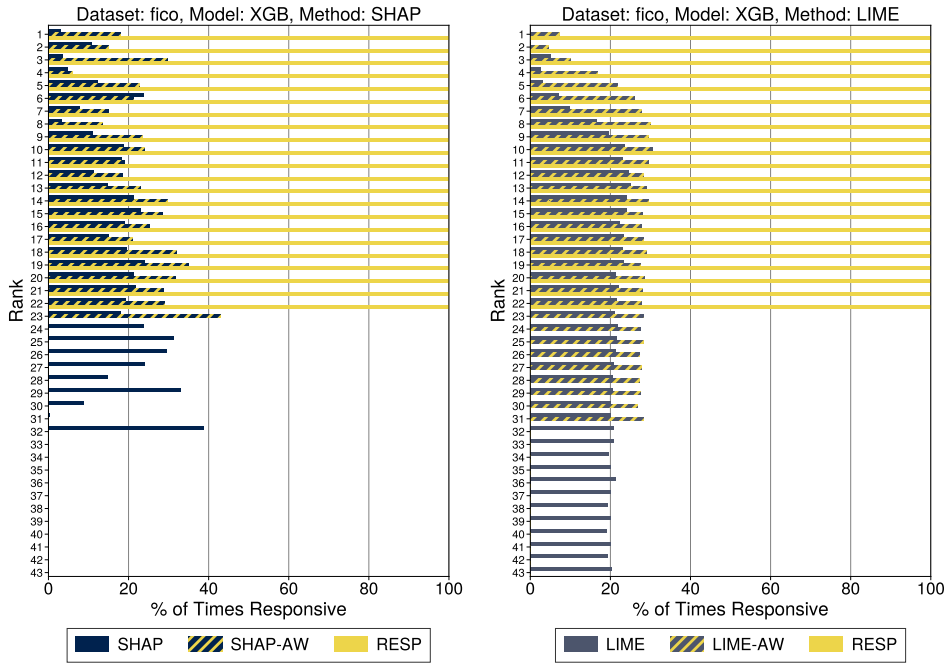

Figure 4: The percent of times where the feature at the shown rank from LIME, LIME-AW, SHAP, SHAP-AW and RESP is responsive – i.e. has at least one single-feature action that leads to recourse – for denied individuals. Only features with a non-zero score under the feature attribution method are shown.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

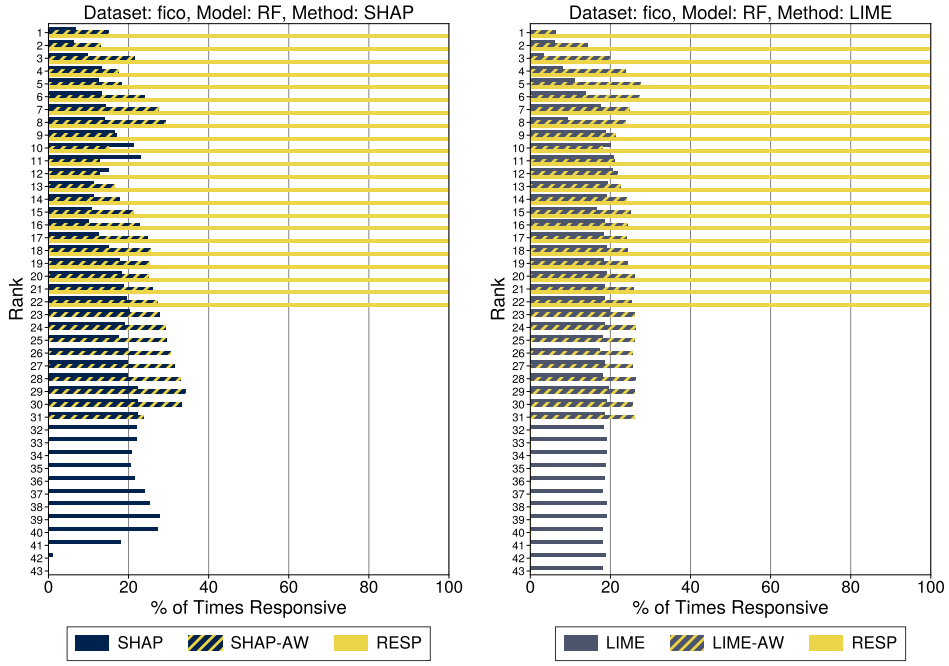

Figure 5: The percent of times where the feature at the shown rank from LIME, LIME-AW, SHAP, SHAP-AW and RESP is responsive – i.e. has at least one single-feature action that leads to recourse – for denied individuals. Only features with a non-zero score under the feature attribution method are shown.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

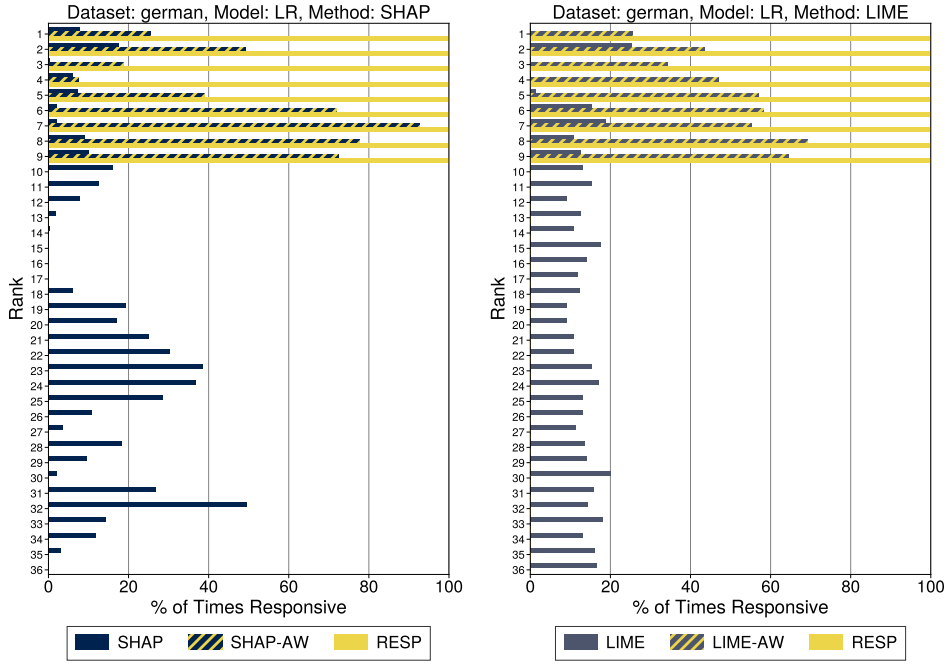

Figure 6: The percent of times where the feature at the shown rank from LIME, LIME-AW, SHAP, SHAP-AW and RESP is responsive – i.e. has at least one single-feature action that leads to recourse – for denied individuals. Only features with a non-zero score under the feature attribution method are shown.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

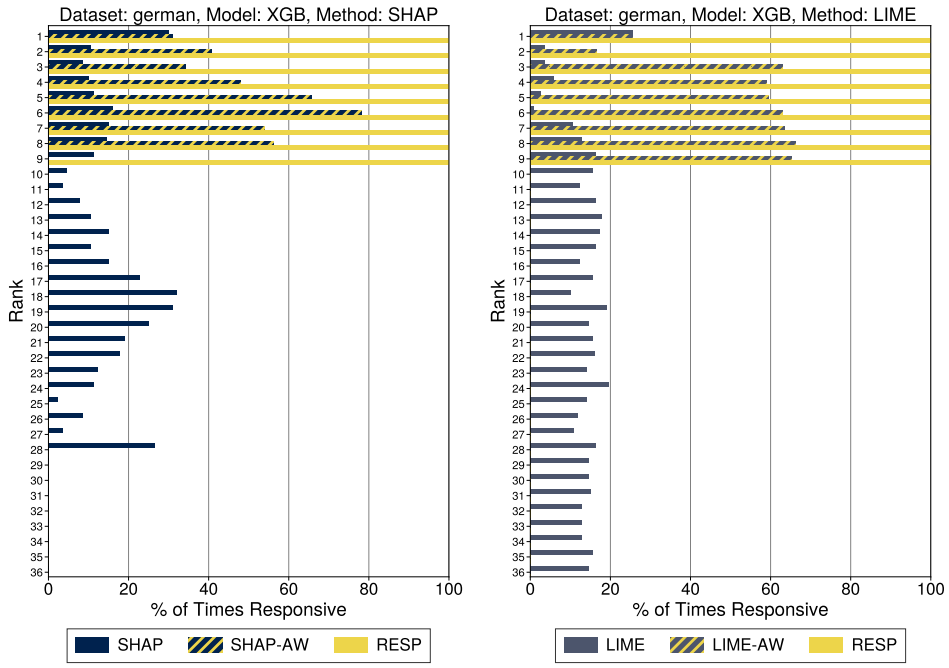

Figure 7: The percent of times where the feature at the shown rank from LIME, LIME-AW, SHAP, SHAP-AW and RESP is responsive – i.e. has at least one single-feature action that leads to recourse – for denied individuals. Only features with a non-zero score under the feature attribution method are shown.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

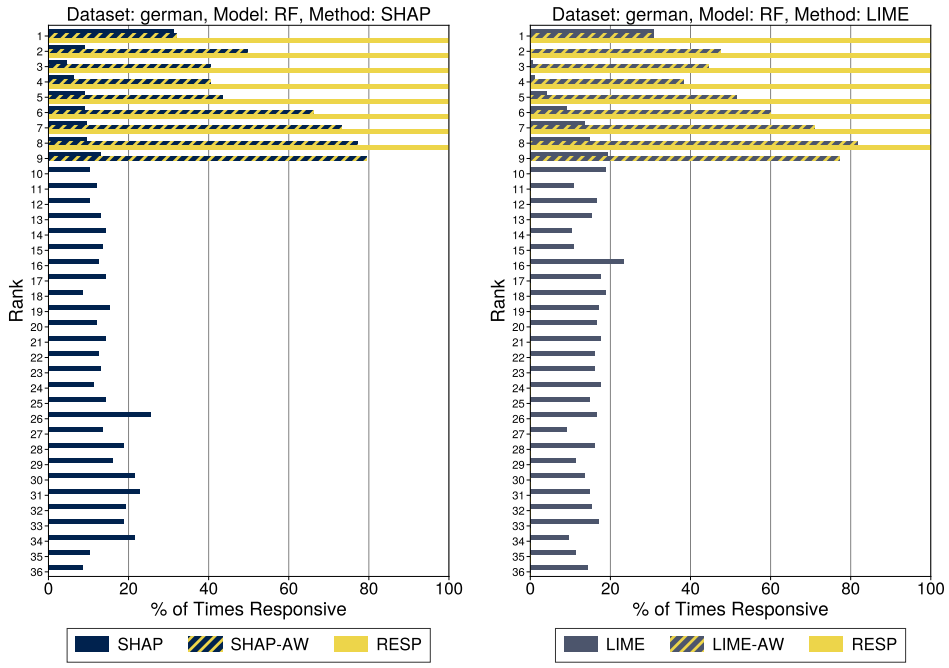

Figure 8: The percent of times where the feature at the shown rank from LIME, LIME-AW, SHAP, SHAP-AW and RESP is responsive – i.e. has at least one single-feature action that leads to recourse – for denied individuals. Only features with a non-zero score under the feature attribution method are shown.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

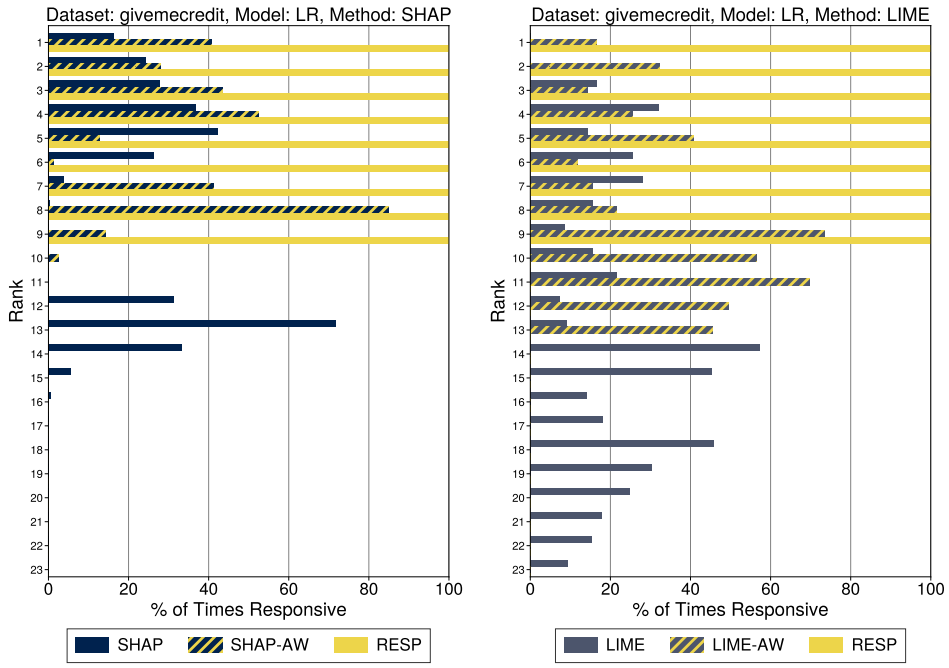

Figure 9: The percent of times where the feature at the shown rank from LIME, LIME-AW, SHAP, SHAP-AW and RESP is responsive – i.e. has at least one single-feature action that leads to recourse – for denied individuals. Only features with a non-zero score under the feature attribution method are shown.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

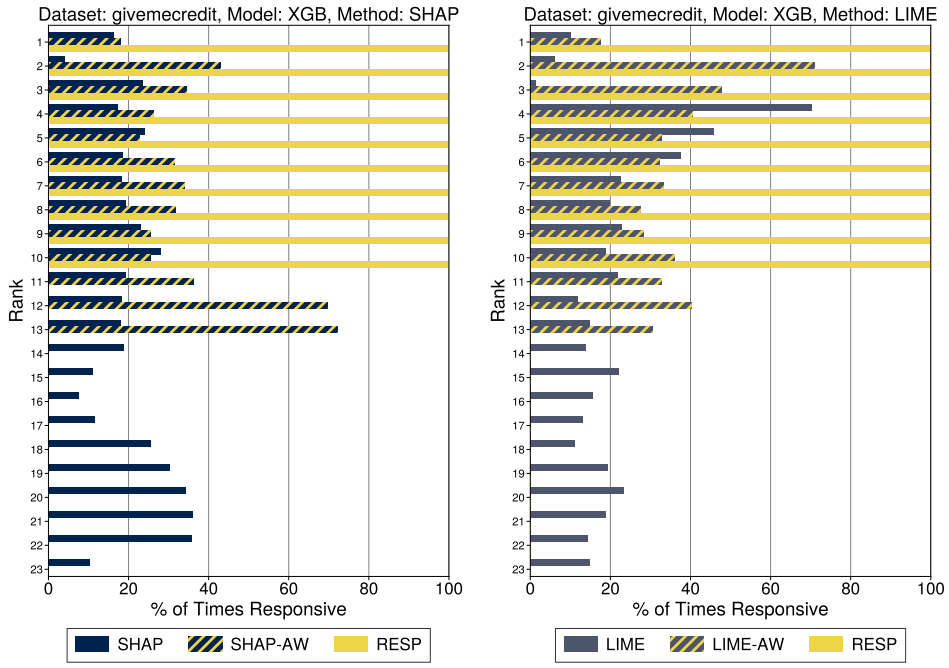

Figure 10: The percent of times where the feature at the shown rank from LIME, LIME-AW, SHAP, SHAP-AW and RESP is responsive – i.e. has at least one single-feature action that leads to recourse – for denied individuals. Only features with a non-zero score under the feature attribution method are shown.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

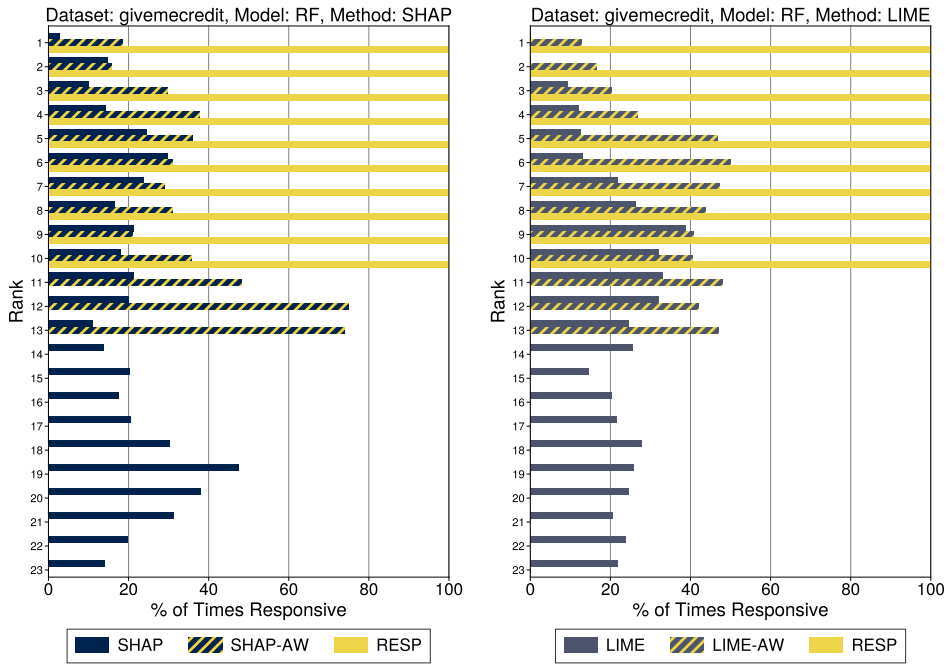

Figure 11: The percent of times where the feature at the shown rank from LIME, LIME-AW, SHAP, SHAP-AW and RESP is responsive – i.e. has at least one single-feature action that leads to recourse – for denied individuals. Only features with a non-zero score under the feature attribution method are shown.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

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction reflect our contributions in Section 3 and our empirical results in Section 4.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations are discussed in Section 5.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The assumptions for the problem statement are provided in Section 2. As for theoretical results, they are either definitions or trivial remarks that do not need written proofs.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We include a project repository that includes code to run the experiment.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     1. If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     2. If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     3. If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     4. We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We include a project repository that includes code and the datasets required to run the experiment.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We outline all training and test details in Section 4 and additional details in **??**.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: Error bars are not included due to running the experiment several times will be computationally expensive.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide computer resource details in **??**.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: We have reviewed the Code of Ethics. Our work does not violate the Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss the societal impact of our work in Section 1 and Section 5.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Our work does not pose risk for misuse of datasets and models.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have cited the original sources of code packages and datasets used in our work.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We include a project repository of our code.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: Our work does not involve human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: Our work does not involve human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.

- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.