Towards User Guided Actionable Recourse

Jayanth Yetukuri
University of California, Santa Cruz
Santa Cruz, California, USA
jayanth.yetukuri@ucsc.edu

Ian Hardy
University of California, Santa Cruz
Santa Cruz, California, USA
ihardy@ucsc.edu

Yang Liu
University of California, Santa Cruz
Santa Cruz, California, USA
yangliu@ucsc.edu

ABSTRACT

Machine Learning's proliferation in critical fields such as healthcare, banking, and criminal justice has motivated the creation of tools which ensure trust and transparency in ML models. One such tool is *Actionable Recourse* (AR) for negatively impacted users. AR describes recommendations of cost-efficient changes to a user's *actionable* features to help them obtain favorable outcomes. Existing approaches for providing recourse optimize for properties such as proximity, sparsity, validity, and distance-based costs. However, an often-overlooked but crucial requirement for actionability is a consideration of *User Preference* to guide the recourse generation process. In this work, we attempt to capture user preferences via soft constraints in three simple forms: *i) scoring continuous features, ii) bounding feature values* and *iii) ranking categorical features*. Finally, we propose a gradient-based approach to identify *User Preferred Actionable Recourse (UP-AR)*. We carried out extensive experiments to verify the effectiveness of our approach.

CCS CONCEPTS

• **Theory of computation** → **Actionable Recourse**; • **Computing methodologies** → *Knowledge representation and reasoning*; • **Human-centered computing** → **Human computer interaction (HCI)**.

KEYWORDS

Actionable recourse, User preference

ACM Reference Format:
Jayanth Yetukuri, Ian Hardy, and Yang Liu. 2023. Towards User Guided Actionable Recourse. In *AAAI/ACM Conference on AI, Ethics, and Society (AIES '23), August 08–10, 2023, Montréal, QC, Canada.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3600211.3604708

1 INTRODUCTION

Actionable Recourse (AR) [30] refers to a list of actions an individual can take to obtain a desired outcome from a fixed Machine Learning (ML) model. Several domains such as lending [28], insurance [26], resource allocation [6, 27] and hiring decisions [1] are required to suggest recourses to ensure the trust of a decision system; in such scenarios, it is critical to ensure the actionability (the viability of taking a suggested action) of recourse, otherwise the suggestions are pointless. Consider an individual named Alice who applies for a loan, and a bank, which uses an ML-based classifier, who denies it. Naturally, Alice asks - *What can I do to get the loan?* The inherent question is what action she must take to obtain the loan in the future. *Counterfactual explanation* introduced in *Wachter* [31] provides a *what-if* scenario to alter the model's decision, but it does not account for actionability. AR aims to provide Alice with a *feasible* action set which is both actionable by Alice and which suggests as low-cost modifications as possible.

While some features (such as age or sex) are inherently inactionable for all individuals, Alice's personalized constraints may also limit her ability to take action on certain suggested recourses (such as a strong reluctance to secure a co-applicant). We call these localized constraints *User Preferences*, synonymous to user-level constraints introduced as *local feasibility* by Mahajan et al. [17]. Figure 1 illustrates the motivation behind UP-AR. Note that how similar individuals can prefer contrasting recourse.

Actionability, as we consider it, is centered explicitly around individual preferences, and similar recourses provided to two individuals (Alice and Bob) with identical feature vectors may not necessarily be equally actionable. Most existing methods of finding actionable recourse are restricted to *omissions* of features from the *actionable feature set* and *box constraints* [18] that bound actions.

In this study, we discuss three forms of user preferences and propose a user-provided score formulation for capturing these different idiosyncrasies. We believe that communicating in terms of preference scores (by say, providing a 1-10 rating on the actionability of specific features) improves the explainability of a recourse generation mechanism, which ultimately improves trust in the underlying model. Such a system could also be easily re-run with different preference scores, allowing for diversifiable recourse. We surveyed 40 individuals and found that an overwhelming 60% majority preferred to provide their preferences on individual features for influencing a recourse mechanism, as opposed to receiving multiple "stock" recourse options or simply receiving a single option. Additional details of our survey are included in Section 7. We provide a hypothetical example of UP-AR's ability to adapt to preferences in Table 1.

Motivated by the above considerations, we capture soft user preferences along with hard constraints and identify recourse based on local desires without affecting the success rate of identifying recourse. For example, consider Alice prefers to have 80% of the recourse "cost" from loan duration and only 20% from the loan amount, meaning she prefers to have recourse with a minor reduction in the loan amount. Such recourse enables Alice to get the benefits of a loan on her terms, and can easily be calculated to Alice's desire. We study the problem of providing *user preferred recourse* by solving a custom optimization for individual user-based preferences. Our contributions include:

AIES '23, August 08–10, 2023, Montréal, QC, Canada

ACM ISBN 979-8-4007-0231-0/23/08.
https://doi.org/10.1145/3600211.3604708

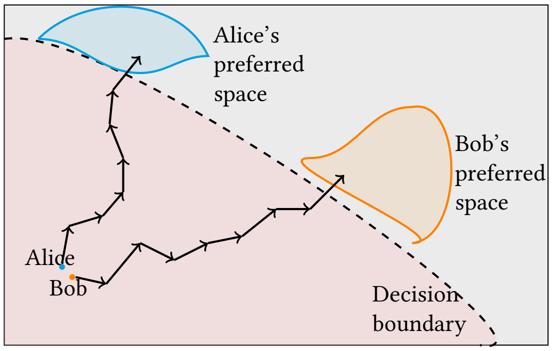


Figure 1: Illustration of UP-AR. Similar individuals Alice and Bob with contrasting preferences can have different regions of desired feature space for a recourse.

Table 1: A hypothetical actionable feature set of adversely affected individuals sharing similar features and corresponding suggested actions by AR and UP-AR. UP-AR provides personalized recourses based on individual user preferences.

Actionable Features	Curr. val.	UP-AR values	
		Alice	Bob
LoanDuration	18	8	17
LoanAmount	$1940	$1840	$1200
HasGuarantor	0	0	1
HasCoapplicant	0	1	0

- We start by enabling Alice to provide three types of user preferences: i) *Scoring*, ii) *Ranking*, and iii) *Bounding*. We embed them into an optimization function to guide the recourse generation mechanism.
- We then present *User Preferred Actionable Recourse (UP-AR)* to identify a recourse tailored to her liking. Our approach highlights a cost correction step to address the *redundancy* induced by our method.
- We consolidate performance metrics with empirical results of UP-AR across multiple datasets and compare them with state-of-art techniques.

1.1 Related Works

Several methods exist to identify counterfactual explanations, such as FACE [22], which uses the shortest path to identify counterfactual explanations from high-density regions, and Growing Spheres (GS) [16] which employs random sampling within increasing hyperspheres for finding counterfactuals. CLUE [3] identifies counterfactuals with low uncertainty in terms of the classifier's entropy within the data distribution. Similarly, manifold-based CCHVAE [21] generates high-density counterfactuals through the use of a latent space model. However, there is often no guarantee that the *what-if* scenarios identified by these methods are attainable.

Existing research focuses on providing feasible recourses, yet comprehensive literature on understanding and incorporating user preferences within the recourse generation mechanism is lacking. It is worth mentioning that instead of understanding user preferences, Mothilal et al. [18] provides a user with diverse recourse options and hopes that the user will benefit from at least one. The importance of diverse recourse recommendations has also been explored in recent works [18, 25, 31], which can be summarized as increasing the chances of actionability as intuitively observed in the domain of unknown user preferences [13]. Karimi et al. [14] and Cheng et al. [5] also resolve uncertainty in a user's cost function by inducing *diversity* in the suggested recourses. Interestingly, only 16 out of the 60 recourse methods explored in the survey by Karimi et al. [13] include diversity as a constraint where diversity is measured in terms of distance metrics. Alternatively, studies like Cui et al. [7], Rawal and Lakkaraju [23], Ustun et al. [30] optimize on a universal cost function. This does not capture individual idiosyncrasies and preferences crucial for actionability.

Efforts of eliciting user preferences include recent work by De Toni et al. [8]. The authors provide interactive human-in-the-loop approach, where a user continuously interacts with the system. However, learning user preferences by asking them to select from one of the *partial interventions* provided is a derivative of providing a diverse set of recourse candidates. In this work, we consider fractional cost as a means to communicate with Alice, where fractional cost of a feature refers to *fraction of cost incurred from a feature i out of the total cost of the required intervention.*

The notion of user preference or user-level constraints was previously studied as *local feasibility* [17]. Since users can not precisely quantify the cost function [23], Yadav et al. [32] diverged from the assumption of a universal cost function and optimizes over the distribution of cost functions. We argue that the inherent problem of feasibility can be solved more accurately by capturing and understanding Alice's recourse preference and adhering to her constraints which can vary between *Hard Rules* such as unable to bring a co-applicant and *Soft Rules* such as hesitation to reduce the amount, which should not be interpreted as unwillingness. This is the first study to capture individual idiosyncrasies in the recourse generation optimization to improve feasibility.

2 PROBLEM FORMULATION

Consider a binary classification problem where each instance represents an individual's feature vector $\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2, \cdot, \mathbf{x}_D]$ and associated binary label $\mathbf{y} \in \{-1, +1\}$. We are given a model $f(\mathbf{x})$ to classify $\mathbf{x}$ into either -1 or $+1$. Let $f(\mathbf{x}) = +1$ be the desirable output of $f(\mathbf{x})$ for Alice. However, Alice was assigned an undesirable label of -1 by f. We consider the problem of suggesting action $\mathbf{r} = [\mathbf{r}_1, \mathbf{r}_2, \cdot, \mathbf{r}_D]$ such that $f(\mathbf{x} + \mathbf{r}) = +1$. Since suggested recourse only requires actions to be taken on *actionable features* denoted by F_A, we have $\mathbf{r}_i \equiv 0 : \forall i \notin F_A$. We further split F_A into *continuous actionable features* F_{con} and *categorical actionable features* F_{cat} based on feature domain. Action $\mathbf{r}$ is obtained by solving the following optimization, where $userCost(\mathbf{r}, \mathbf{x})$ is any predefined cost function of taking an action $\mathbf{r}$ such that:

$$\min_{\mathbf{r}} \ userCost(\mathbf{r}, \mathbf{x}) \tag{1}$$

$$s.t. \ userCost(\mathbf{r}, \mathbf{x}) = \sum_{i \in F_A} userCost(\mathbf{r}_i, \mathbf{x}_i) \tag{2}$$

$$\text{and } f(\mathbf{x} + \mathbf{r}) = +1. \tag{3}$$

2.1 Capturing individual idiosyncrasies

A crucial step for generating recourse is identifying *local feasibility* constraints captured in terms of individual user preferences. In this study, we assume that every user provides their preferences in three forms. Every continuous actionable feature $i \in F_{con}$ is associated with a *preference score* Γ_i obtained from the affected individual. Additional preferences in the form of feature value bounds and ranking for preferential treatment of categorical features are also requested from the user.

User Preference Type I (Scoring continuous features): User preference for continuous features are captured in $\Gamma_i \in [0, 1] : \forall i \in F_{con}$ subject to $\sum_{i \in F_{con}} \Gamma_i = 1$. Such *soft constraints* capture the user's preference without omitting the feature from the actionable feature set. Γ_i refers to the fractional cost of action Alice prefers to incur from a continuous feature i. For example, consider $F_{con} = \{LoanDuration, LoanAmount\}$ with corresponding user-provided scores $\Gamma = \{0.8, 0.2\}$ implying that Alice prefers to incur 80% of fractional feature cost from taking action on *LoanDuration*, while only 20% of fractional cost from taking action on *LoanAmount*. Here, Alice prefers reducing *LoanDuration* to *LoanAmount* and providing recourse in accordance improves actionability.

User Preference Type II (Bounding feature values): Users can also provide constraints on values for individual features in F_A. These constraints are in the form of lower and upper bounds for individual feature values represented by $\underline{\delta_i}$ and $\overline{\delta_i}$ for any feature i respectively. These constraints are used to discretize the steps. For a continuous feature i, action steps can be discretized into pre-specified step sizes of $\Delta_i = \{s : s \in [\underline{\delta_i}, \overline{\delta_i}]\}$. For categorical features, steps are defined as the feasible values a feature can take. For all categorical features we define, $\Delta_i = \{\underline{\delta_i}, \ldots, \overline{\delta_i}\} : \forall i \in F_{cat}$ representing the possible values for categorical feature i.

User Preference Type III (Ranking categorical features): Users are also asked to provide a ranking function $\mathcal{R} : F_{cat} \to \mathbb{Z}^{+1}$ on F_{cat}. Let $\mathcal{R}_i$ refers to the corresponding rank for a categorical feature i. Our framework identifies recourse by updating the candidate action based on the ranking provided. For example, consider $F_{cat} = \{HasCoapplicant, HasGuarantor, CriticalAccountOrLoansElsewhere\}$ for which Alice ranks them by $\{3, 2, 1\}$. The recourse generation system considers suggesting an action on *HasGuarantor* before *HasCoapplicant*. Ranking preferences can be easily guaranteed by a simple override in case of discrepancies while finding a recourse.

2.1.1 *Cognitive simplicity of preference scores.*

The user preferences proposed are highly beneficial for guiding the recourse generation process. Please note that in the absence of these preferences, the recourse procedure falls back to the default values set by a domain expert. Additionally, the users can be first presented with the default preferences, and asked to adjust as per their individual preferences. A simple user interface can help them interact with the system intuitively. For example, adjusting a feature score automatically adjusts the corresponding preference type scores.

2.2 Proposed optimization

We depart from capturing a user's cost of feature action and instead obtain their preferences for each feature. We elicit three forms of preferences detailed in the previous section and iteratively take steps in the action space. We propose the following optimization over the basic predefined steps based on the *user preferences*. Let us denote the inherent hardness of feature action $\mathbf{r}_i$ for feature value $\mathbf{x}_i$ using $cost(\mathbf{r}, \mathbf{x})$ which can be any cost function easily communicable to Alice. Here, $cost\left(\mathbf{r}_i^{(t)}, \mathbf{x}_i\right)$ refers to a "universal" cost of taking an action $\mathbf{r}_i^{(t)}$ for feature value $\mathbf{x}_i$ at step t. Note that this cost function or quantity differs from the $userCost(\cdot, \cdot)$ function specified earlier. This quantity is capturing the inherent difficulty of taking an action.

$$\max_{\mathbf{r}} \sum_{i \in F_A} \frac{\Gamma_i}{cost(\mathbf{r}_i, \mathbf{x}_i)} \tag{Type I}$$

$$s.t. \ f(\mathbf{x} + \mathbf{r}) = +1$$

$$\Gamma_i = 0 : \ \forall i \notin F_A \tag{actionability}$$

$$\Gamma_j = 1 : \ \forall j \in F_{cat}$$

$$\mathbf{r}_i \in \Delta_i : \ i \in F_A \tag{Type II}$$

$$\mathbf{1}\{\mathbf{r}_i > 0\} \geq \mathbf{1}\{\mathbf{r}_j > 0\} : \mathcal{R}_i \geq \mathcal{R}_j \ \forall i, j \in F_{cat} \tag{Type III}$$

The proposed method minimizes the cost of a recourse weighted by Γ_i for all actionable features. We discuss the details of our considerations of cost function in Section 3.1. The order preference of categorical feature actions can be constrained by restrictions while finding a recourse. The next section introduces UP-AR as a stochastic solution to the proposed optimization.

3 USER PREFERRED ACTIONABLE RECOURSE (UP-AR)

Our proposed solution, User Preferred Actionable Recourse (UP-AR), consists of two stages. The first stage generates a candidate recourse by following a connected gradient-based iterative approach. The second stage then improves upon the *redundancy* metric of the generated recourse for better actionability. The details of UP-AR are consolidated in Algorithm 1 and visualized in Figure 2.

3.1 Stage 1: Stochastic gradient-based approach

Poyiadzi et al. [22] identifies a counterfactual by following a high-density connected path from the feature vector $\mathbf{x}$. With a similar idea, we follow a connected path guided by the user's preference to identify a feasible recourse. We propose incrementally updating the candidate action with a predefined step size to solve the optimization. At each step t, a candidate intervention is generated, where any feature i is updated based on a Bernoulli trial with probability $I_i^{(t)}$ derived from user preference scores and the cost of taking a

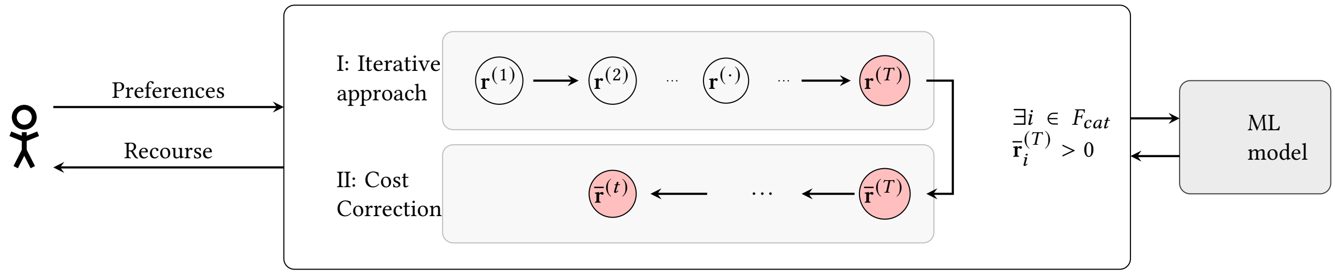


Figure 2: Framework of UP-AR. Successful recourse candidates; $\mathbf{r}^{(\cdot)}$, $\bar{\mathbf{r}}^{(\cdot)}$ are colored in pink.

predefined step $\delta_i^{(t)}$ using the following procedure:

$$I_i^{(t)} \sim Bernoulli\left(\sigma\left(z_i^{(t)}\right)\right) \tag{4}$$

$$\text{where } \sigma\left(z_i^{(t)}\right) = \frac{e^{z_i^{(t)}/\tau}}{\sum_{j \in F_A} e^{z^{(t)}/\tau}},\ z_i^{(t)} = \frac{\Gamma_i}{cost\left(\mathbf{r}_i^{(t)}, \mathbf{x}_i\right)} \tag{5}$$

With precomputed costs for each step, *weighted inverse cost* is computed for each feature, and these values are mapped to a probability distribution using a function like softmax. *Softmax* gives a probabilistic interpretation $P\left(I_i^{(t)} = 1 | z_i^{(t)}\right) = \sigma\left(z_i^{(t)}\right)$ by converting $z_i^{(t)}$ scores into probabilities.

We leverage the idea of *log percentile shift* from AR to determine the cost of action since it is easier to communicate with the users in terms of percentile shifts. Specifically, we follow the idea and formulation in [30] to define the cost:

$$cost\left(\mathbf{r}_i, \mathbf{x}_i\right) = log\left(\frac{1 - Q_i\left(\mathbf{x}_i + \mathbf{r}_i\right)}{1 - Q_i\left(\mathbf{x}_i\right)}\right) \tag{6}$$

were $Q_i\left(\mathbf{x}_i\right)$ representing the *percentile* of feature i with value $\mathbf{x}_i$ is a score below which $Q_i\left(\mathbf{x}_i\right)$ percentage of scores fall in the frequency distribution of feature values in the target population.

We adapt and extend the idea that counterfactual explanations and adversarial examples [29] have a similar goal but with contrasting intention [19]. A popular approach to generating adversarial examples [10] is by using a gradient-based method. We employ the learning of adversarial example generation to determine the direction of feature modification in UP-AR: the Jacobian matrix is used to measure the local sensitivity of outputs with respect to each input feature. Consider that $f : \mathbb{R}^D \rightarrow \mathbb{R}^K$ maps a D-dimensional feature vector to a K-dimensional vector, such that each of the partial derivatives exists. For a given $\mathbf{x} = [\mathbf{x}_1, \ldots, \mathbf{x}_i, \ldots, \mathbf{x}_D]$ and $f(\mathbf{x}) = [f_{[1]}(\mathbf{x}), \ldots, f_{[j]}(\mathbf{x}), \ldots, f_{[K]}(\mathbf{x})]$, the Jacobian matrix of f is defined to be a $D \times K$ matrix denoted by $\mathbf{J}$, where each (j, i) entry is $\mathbf{J}_{j,i} = \frac{\partial f_{[j]}(\mathbf{x})}{\partial \mathbf{x}_i}$. For a neural network (NN) with at least one hidden layer, $\mathbf{J}_{j,i}$ is obtained using the chain rule during backpropagation. For an NN with one hidden layer represented by *weights* $\{w\}$, we have:

$$\mathbf{J}_{j,i} = \frac{\partial f_{[j]}(\mathbf{x})}{\partial \mathbf{x}_i} = \sum_l \frac{\partial f_{[l]}(\mathbf{x})}{\partial a_l} \frac{\partial a_l}{\partial \mathbf{x}_i} \text{ where } a_l = \sum_i w_{li} \mathbf{x}_i \tag{7}$$

Where in Equation 7, a_l is the output (with possible activation) of the hidden layer and w_l is the weight of the node l. Notice line 4 in Algorithm 1 which *updates the candidate* action for a feature i at step t as:

$$\mathbf{r}_i^{(t)} = \mathbf{r}_i^{(t-1)} + Sign\left(\mathbf{J}_{+1,i}^{(t)}\right) \cdot I_i^{(t)} \cdot \delta_i^{(t)} \tag{8}$$

Following the traditional notation of a binary classification problem and with a bit of abuse of notation $-1 \rightarrow 1, +1 \rightarrow +1$, $Sign\left(\mathbf{J}_{+1,i}^{(t)}\right)$ captures the direction of the feature change at step t. This direction is iteratively calculated, and additional constraints such as non-increasing or non-decreasing features can be placed at this stage.

Algorithm 1 User Preferred Actionable Recourse (UP-AR)

Input: Model f, user feature vector $\mathbf{x}$, cost function $cost(\cdot,\cdot)$, step size $\Delta_i : \forall i \in F_A$, maximum steps T, action $\mathbf{r}$ initialized to $\mathbf{r}^{(0)}$, fixed τ, $t = 1$.

1: **while** $t \leq T$ or $f\left(\mathbf{x} + \mathbf{r}^{(t)}\right) \neq +1$ **do**
2: $\quad z_i^{(t)} = \frac{\Gamma_i}{cost\left(\mathbf{r}_i^{(t)}, \mathbf{x}_i\right)} : \forall i$
3: $\quad I_i^{(t)} \sim Bern(\sigma(z_i^{(t)})) : \forall i, \text{where } \sigma(z_i^{(t)}) = \frac{e^{z_i^{(t)}/\tau}}{\sum_{j \in F_A} e^{z^{(t)}/\tau}}$
4: $\quad \mathbf{r}_i^{(t)} = \mathbf{r}_i^{(t-1)} + Sign\left(\mathbf{J}_{+1,i}^{(t)}\right) \cdot I_i^{(t)} \cdot \delta_i^{(t)} : \forall i \in F_A$
5: $\quad t = t + 1$
6: Let $\hat{t}$ be the smallest step such that $f(\mathbf{x} + \mathbf{r}^{(\hat{t})}) = +1$ and initialize $t = \hat{t}$
7: **if** $\exists i \in F_{cat} : \mathbf{r}_i^{(t)} > 0$ **then**
8: $\quad$ **while** $f\left(\mathbf{x} + \bar{\mathbf{r}}^{(t)}\right) = +1$ **do**
9: $\quad\quad \bar{\mathbf{r}}^{(t)} = \mathbf{r}^{(t)}$
10: $\quad\quad \bar{\mathbf{r}}_i^{(t)} = \mathbf{r}_i^{(\hat{t})} : \forall i \in F_{cat}$
11: $\quad\quad t = t - 1$
12: **return** $\bar{\mathbf{r}}^{(t)}$ as action $\mathbf{r}$

3.1.1 Calibrating frequency of categorical actions. We employ *temperature scaling* [11] parameter τ observed in Equation 5 to calibrate UP-AR's recourse generation cost. Updates on categorical features with fixed step sizes are expensive, especially for binary categorical values. Hence, tuning the frequency of categorical suggestions can significantly impact the overall cost of a recourse. τ controls the frequency with which categorical actions are suggested. Additionally, if a user prefers updates on categorical features over continuous features, UP-AR has the flexibility to address this with a smaller τ.

To study the effect of τ on overall cost, we train a Logistic Regression (LR) model on a processed version of *German* [4] dataset and generate recourses for the 155 individuals who were denied credit. The cost gradually decreases with decreasing τ since the marginal probability of suggesting a categorical feature change is diminished and the corresponding experiment is deferred to the Appendix. Hence, without affecting the success rate of recourse generation, the overall cost of generating recourses can be brought down by decreasing τ. In simple terms, with a higher τ, UP-AR frequently suggests recourses with expensive categorical actions. We note that τ can also be informed by a user upon seeing an initial recourse. After the strategic generation of an intervention, we implement a cost correction to improve upon the potential redundancy of actions in a recourse option.

3.2 Stage 2: Redundancy & Cost Correction (CC)

In our experiments, we observe that once an expensive action is recommended for a categorical feature, some of the previous action steps might become redundant. Consider an LR model trained on the processed *german* dataset. Let $F_A = \{LoanDuration, LoanAmount, HasGuarantor\}$ out of all the 26 features, where *HasGuarantor* is a binary feature which represents the user's ability to get a guarantor for the loan. Stage 1 takes several steps over *LoanAmount* and *LoanDuration* before recommending to update *HasGuarantor*. These steps are based on the feature action probability from Equation 5. Since categorical feature updates are expensive and occur with relatively low probability, Stage 1 finds a low-cost recourse by suggesting low-cost steps more frequently in comparison with high-cost steps.

Table 2: Redundancy corrected recourse for a hypothetical individual.

Features to change	Current values	Stage 1 values	Stage 2 values
LoanDuration	18	8	12
LoanAmount	\$1940	\$1040	\$1540
HasGuarantor	0	1	1

Once an update to a categorical feature is recommended, some of the previous low-cost steps may be redundant, which can be rectified by tracing back previous continuous steps. Consider a scenario such that $\exists i \in F_{cat} : \mathbf{r}_i^{(T)} > 0$ for a recourse obtained after T steps in Stage 1. The CC procedure updates all the intermediary recourse candidates to reflect the categorical changes i.e., $\forall i \in F_{cat} : \mathbf{r}_i^{(T)} > 0$, we update $\mathbf{r}_i^{(t)} = \mathbf{r}_i^{(T)} : \forall t \in \{1, 2, \ldots, T-1\}$ to obtain $\bar{\mathbf{r}}^{(t)}$. We then perform a linear retracing procedure to return $\bar{\mathbf{r}}^{(t)}$ such that $f\left(\mathbf{x} + \bar{\mathbf{r}}^{(t)}\right) = +1$ for the smallest t.

4 DISCUSSION AND ANALYSIS

In this section, we analyze the user preference performance of UP-AR. For simplicity, a user understands cost in terms of log percentile shift from her initial feature vector described in Section 3. Let $\hat{\Gamma}_i$ be the observed fractional cost for feature i formally defined in Equation 11. Any cost function can be plugged into UP-AR with no restrictions. A user prefers to have Γ_i fraction of the total desired percentile shift from feature i. Consider $F_A = \{LoanDuration,$

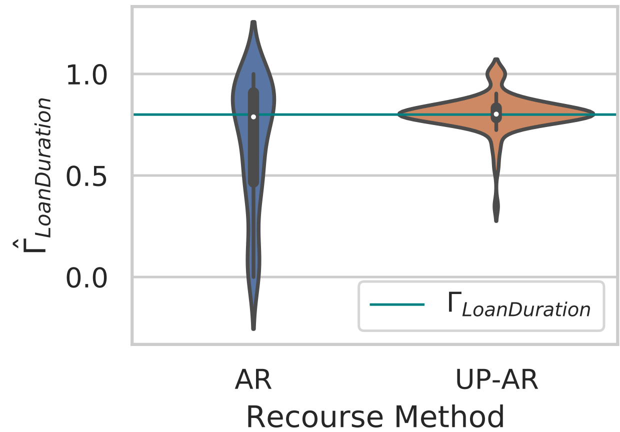


Figure 3: AR and UP-AR's distribution of $\hat{\Gamma}_{LoanDuration}$ for a *Logistic Regression* model trained on *German*.

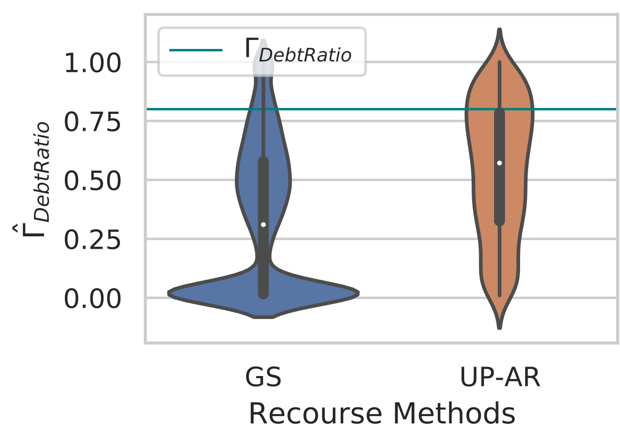


Figure 4: GS and UP-AR's distribution of $\hat{\Gamma}_{DebtRatio}$ for a *Neural Network* model trained on *GMSC*.

$LoanAmount\}$ and let the corresponding user scores provided by all the adversely affected individuals be: $\Gamma = \{0.8, 0.2\}$. Here, "Denied loan applicants prefers reducing *LoanDuration* to *LoanAmount* by 8 : 2." Figure 3 shows the frequency plot of feature cost ratio for feature *LoanDuration* out of total incurred cost from *LoanDuration* and *LoanAmount*. i.e., $y-$axis represents $\hat{\Gamma}_i$. Also, Figure 4 further shows the fractional cost of feature *DebtRatio* for recourses obtained for a NN based model trained on *Give Me Some Credit (GMSC)* dataset. These experiments signify the adaptability of UP-AR to user preferences and provides evidence that distribution of $\hat{\Gamma}_i$ is centered around Γ_i.

Lemma 4.1. *Consider UP-AR identified recourse* $\mathbf{r}$ *for an individual* $\mathbf{x}$. *If* $C_{i,min}^{(T^*)}$ *and* $C_{i,max}^{(T^*)}$ *represent the minimum and maximum cost* *of any step for feature i until T^*, then:*

$$\mathbb{E}\left[cost\left(\mathbf{r}_i, \mathbf{x}_i\right)\right] \leq T^* \sigma \left(\frac{\Gamma_i}{C_{i,min}^{(T^*)}} \right) C_{i,max}^{(T^*)}. \tag{9}$$

Lemma 4.1 implies that the expected cost $\mathbb{E}\left[cost\left(\mathbf{r}_i, \mathbf{x}_i\right)\right]$, specifically for a continuous feature action is positively correlated to the probabilistic interpretation of user preference scores. Hence $\mathbf{r}$ satisfies users critical Type I constraints in expectation. Recall that Type II and III constraints are also applied at each step t. Lemma 4.1 signifies that UP-AR adheres to user preferences and thereby increases the actionability of a suggested recourse.

Corollary 4.2. *For UP-AR with a linear $\sigma(\cdot)$, predefined steps with equal costs and $cost(\mathbf{r}, \mathbf{x}) = \sum_{i \in F_A} cost(\mathbf{r}_i, \mathbf{x}_i)$, total expected cost after T^* steps is:*

$$\mathbb{E}\left[cost\left(\mathbf{r}, \mathbf{x}\right)\right] \leq T^* \sum_{i \in F_A} \sigma\left(\Gamma_i\right). \tag{10}$$

Corollary 4.2 states that with strategic selection of $\sigma(\cdot)$, $\delta_i^{(\cdot)}$ and $cost(\cdot,\cdot)$, UP-AR can also tune the total cost of suggested actions. In the next section, we will compare multiple recourses based on individual user preferences for a randomly selected adversely affected individual.

4.1 Case study of individuals with similar features but disparate preferences

Given an LR model trained on *german* dataset and Alice, Bob and Chris be three adversely affected individuals. $F_A = \{$*LoanDuration*, *LoanAmount*, *HasGuarantor*$\}$ and corresponding user preferences are provided by the users. In Table 3, we consolidate the corresponding recourses generated for the specified disparate sets of preferences.

From Table 3 we emphasize the ability of UP-AR to generate a variety of user-preferred recourses based on their preferences, whereas AR always provides the same low-cost recourse for all the individuals. The customizability of feature actions for individual users can be found in the table. When the Type I score for *LoanAmount* is 0.8, UP-AR prefers decreasing loan amount to loan duration. Hence, the loan amount is much lesser for Chris than for Alice and Bob.

5 EMPIRICAL EVALUATION

In this section, we demonstrate empirically: 1) that UP-AR respects Γ_i-fractional user preferences at the population level, and 2) that UP-AR also performs favorably on traditional evaluate metrics drawn from CARLA [20]. We used the native CARLA catalog for the `Give Me Some Credit (GMSC)` [12], `Adult Income (Adult)` [9] and `Correctional Offender Management Profiling for Alternative Sanctions (COMPAS)` [2] data sets as well as pre-trained models (both the **Neural Network** (NN) and **Logistic Regression** (LR)). NN has three hidden layers of size [18, 9, 3], and the LR is a single input layer leading to a Softmax function. Although AR is proposed for *linear models*, it can be extended to *nonlinear models* by the local linear decision boundary approximation method LIME [24] (referred as AR-LIME).

PERFORMANCE METRICS:. For UP-AR, we evaluate:

(1) *Success Rate (Succ. Rate)*: The percentage of adversely affected individuals for whom recourse was found.
(2) *Average Time Taken (Avg.Tim.)*: The average time (in seconds) to generate recourse for a single individual.
(3) *Constraint Violations (Con. Vio.)*: The average number of non-actionable features modified.
(4) *Redundancy (Red.)*: A metric that tracks superfluous feature changes. For each successful recourse calculated on a univariate basis, features are flipped to their original value. The redundancy for recourse is the number of flips that do not change the model's classification decision.
(5) *Proximity (Pro.)*: The normalized l_2 distance of recourse to its original point.
(6) *Sparsity (Spa.)*: The average number of features modified.

We provide comparative results for UP-AR against state-of-the-art counterfactual/recourse generation techniques such as GS, Wachter, AR(-LIME), CCHAVE and FACE. These methods were selected based on their popularity and their representation of both independence and dependence based methods, as defined in CARLA. In addition to the traditional performance metrics, we also measure *Preference-Root mean squared error (pRMSE)* between the user preference score and the fractional cost of the suggested recourses. We calculate $pRMSE_i$ for a randomly selected continuous valued feature i using:

$$pRMSE_i = \sqrt{\frac{1}{n} \sum_{j=1}^{n} \left(\hat{\Gamma}_i^{(j)} - \Gamma_i^{(j)} \right)^2} \tag{11}$$

$$\text{where } \hat{\Gamma}_i^{(j)} = \frac{cost\left(\mathbf{r}_i, \mathbf{x}_i\right)}{\sum_{k \in F_{con}} cost\left(\mathbf{r}_k, \mathbf{x}_k\right)} \tag{12}$$

Here $\Gamma_i^{(j)}$ and $\hat{\Gamma}_i^{(j)}$ are user provided and observed preference scores of feature i for an individual j. In Table 4, we summarize *pRMSE*, which is the average error across continuous features such that:

$$pRMSE = \frac{1}{|F_{con}|} \sum_{i \in F_{con}} pRMSE_i. \tag{13}$$

DATASETS:. We train an LR model on the processed version of `german` [4] credit dataset from *sklearn's linear_model* module. We replicate Ustun et al. [30]'s model training and recourse generation on `german`. The dataset contains 1000 data points with 26 features for a loan application. The model decides if an applicant's credit request should be approved or not. Consider $F_{con} = \{$*LoanDuration*, *LoanAmount*$\}$, and $F_{cat} = \{$*CriticalAccountOrLoansElsewhere*, *HasGuarantor*, *HasCoapplicant*$\}$. Let the user scores for F_{con} be $\Gamma = \{0.8, 0.2\}$ and ranking for F_{cat} be $\{3, 1, 2\}$ for all the denied individuals. For this experiment, we set $\tau^{-1} = 4$. Out of 155 individuals with denied credit, AR and UP-AR provided recourses to 135 individuals.

Cost Correction: Out of all the denied individuals for whom categorical actions were suggested, an average of ∼ \$400 in *LoanAmount* was recovered by cost correction.

For the following datasets, for traditional metrics, user preferences were set to be uniform for all actionable features to not bias the results to one feature preference over another:

(1) **GMSC:** The data set from the 2011 Kaggle competition is a credit underwriting dataset with 11 features where the

Table 3: Recourses generated by UP-AR for similar individuals with a variety of preferences.

			Alice		Bob		Chris	
Features to change	**Current values**	**AR values**	**User Pref**	**UP-AR values**	**User Pref**	**UP-AR values**	**User Pref**	**UP-AR values**
LoanDuration	30	25	0.8	20	0.8	10	0.2	27
LoanAmount	$8072	$5669	0.2	$7372	0.2	$6472	0.8	$5272
HasGuarantor	0	1	1	1	0	0	1	1

Table 4: Summary of performance evaluation of UP-AR. Top performers are highlighted in green.

		Neural Network							Logistic Regression						
Data.	**Recourse Method**	**Succ. Rate**	**pRMSE**	**Avg Tim.**	**Con. Vio.**	**Red.**	**Pro.**	**Spa.**	**Succ. Rate**	**pRMSE**	**Avg Tim.**	**Con. Vio.**	**Red.**	**Pro.**	**Spa.**
GMSC	GS	0.75	0.16	0.02	0.00	6.95	1.01	8.89	0.62	0.18	0.03	0.00	4.08	1.39	8.99
	Wachter	1.00	0.18	0.02	1.49	6.84	1.08	8.46	1.00	0.17	0.03	1.23	3.51	1.42	7.18
	AR(-LIME)	0.03	0.17	0.45	0.00	0.00	0.17	1.72	0.17	0.17	0.73	0.00	0.00	0.93	1.91
	CCHVAE	1.00	0.18	1.05	2.0	9.99	1.15	10.1	1.00	0.18	1.37	2.00	8.64	2.05	11.0
	FACE	1.00	0.17	8.05	1.57	6.65	1.20	6.69	1.00	0.16	11.9	1.65	7.47	2.30	8.45
	UP-AR	0.94	0.07	0.08	0.00	1.30	0.49	3.22	1.00	0.07	0.12	0.00	1.47	0.68	3.92
Adult	GS	0.84	0.10	0.03	0.00	2.86	1.30	5.09	0.84	0.10	0.04	0.00	1.76	2.05	5.85
	Wachter	0.55	0.10	0.04	1.44	3.05	0.74	4.90	1.00	0.11	0.10	1.68	0.90	1.44	5.81
	AR(-LIME)	0.42	0.10	9.20	0.00	0.00	2.10	2.54	0.76	0.10	7.37	0.00	0.03	2.10	2.31
	CCHVAE	0.84	0.11	0.77	4.47	5.83	3.95	9.40	0.84	0.10	1.08	4.22	6.85	3.96	9.45
	FACE	1.00	0.10	6.78	4.58	7.54	4.11	7.91	1.00	0.10	8.37	4.53	5.91	4.28	7.81
	UP-AR	0.82	0.10	0.76	0.00	0.78	1.77	2.78	0.82	0.05	0.67	0.00	0.55	1.78	2.88
COMPAS	GS	1.00	0.15	0.03	0.00	1.09	0.47	3.35	1.00	0.14	0.04	0.00	0.34	1.12	3.98
	Wachter	1.00	0.14	0.05	1.00	1.61	0.56	4.35	1.00	0.14	0.04	1.00	0.85	1.06	4.83
	AR(-LIME)	0.65	0.13	0.20	0.00	0.00	0.78	0.90	0.52	0.15	0.24	0.00	0.00	1.45	1.57
	CCHVAE	1.00	0.14	5.09	2.27	4.31	1.70	4.91	1.00	0.14	0.02	1.62	2.70	1.74	4.92
	FACE	1.00	0.15	0.37	2.39	3.96	2.35	4.72	1.00	0.15	0.40	2.47	4.38	2.46	4.81
	UP-AR	0.92	0.08	0.04	0.00	0.60	0.63	1.82	1.00	0.10	0.05	0.00	0.81	0.82	2.74

target is the presence of delinquency. Here, we measure what feature changes would lower the likelihood of delinquency. We again used the default protected features (*age* and *number of dependents*). The baseline accuracy for the NN model is 81%, while the baseline accuracy for the LR is 76%.

(2) **Adult Income:** This dataset originates from 1994 census database with 14 attributes. The model decides whether an individual's income is higher than 50,000 USD/year. The baseline accuracy for the NN model is 85%, while the baseline accuracy for the LR is 83%. Our experiment is conducted on a sample of 1000 data points.

(3) **COMPAS:** The data set consists of 7 features describing offenders and a target representing predictions. Here, we measure what feature changes would change an automated recidivism prediction.

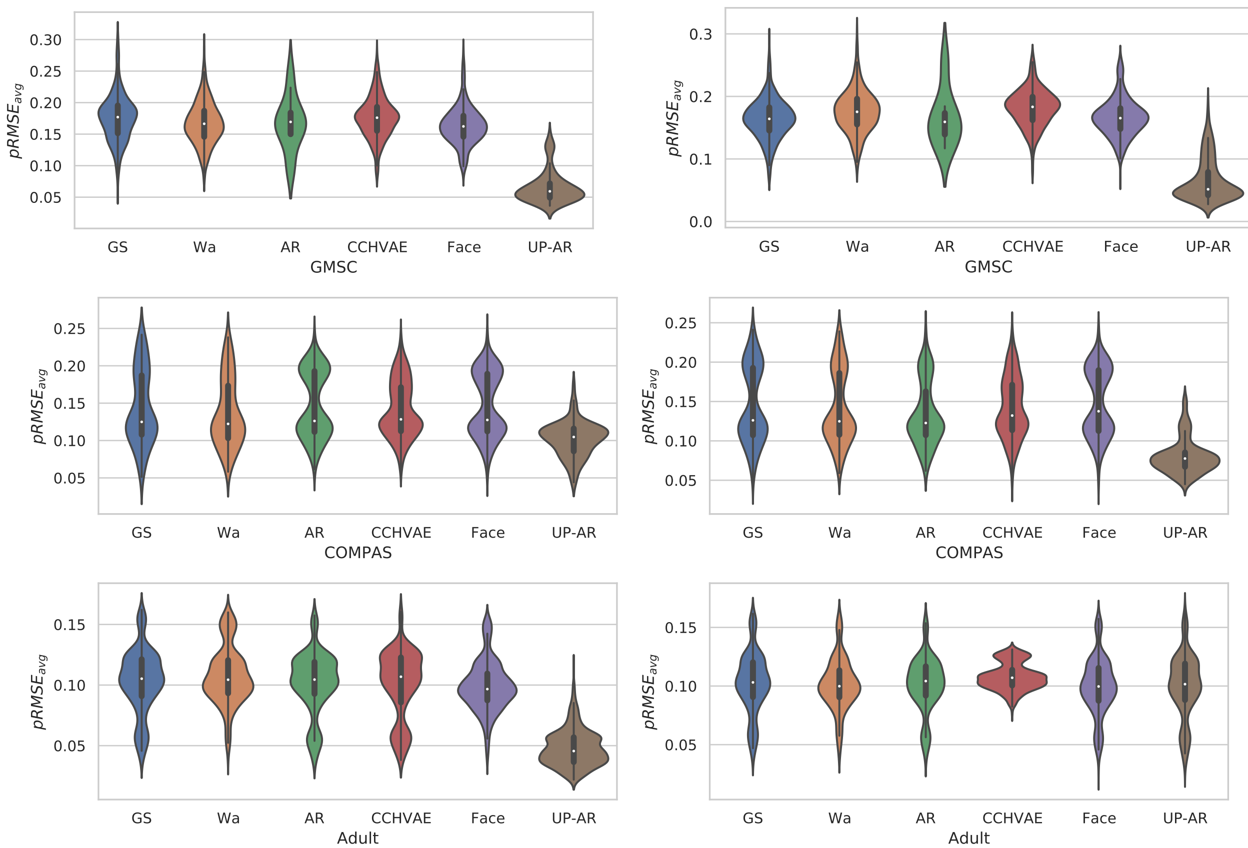
The baseline accuracy for NN is 78%, while baseline accuracy for LR is 71%.

PERFORMANCE ANALYSIS OF UP-AR:. We find UP-AR holistically performs favorably to its counterparts. Critically, it respects feature constraints (which we believe is fundamental to actionable recourse) while maintaining a significantly low redundancy and sparsity. This indicates that it tends to change fewer necessary features. Its speed makes it tractable for real-world use, while its proximity values show that it recovers relatively low-cost recourse. These results highlight the promise of UP-AR as a performative, low-cost option for calculating recourse when user preferences are paramount. UP-AR shows consistent improvements over all the performance metrics. The occasional lower success rate for a NN model is attributed to 0 constraint violations.

*pRMSE***:** We analyze user preference performance in terms of *pRMSE*. From Table 4, we observe that UP-AR's *pRMSE* is consistently better than the state of art recourse methods. The corresponding experimental details and visual representation of the distribution of *pRMSE* is deferred to Appendix 5.1.

5.1 Random user preference study

We performed an experiment with increasing step sizes on *German* dataset. We observed that, with increasing step sizes, $pRMSE_i$ increased from 0.09 to 0.13, whereas it was consistent for AR.



(a) Logistic Regression model

(b) Neural Network model

Figure 5: Distribution of the average *pRMSE* of UP-AR and other recourse methodologies.

In the next experiment, we randomly choose user preference for *LoanDuration* from $[0.4, 0.5, 0.6, 0.7, 0.8]$. The rest of the experimental setup is identical to the setup discussed in Section 4. In this experiment, we observe *pRMSE* with non-universal user preference for adversely affected individuals. Here the average *pRMSE* of both *LoanDuration* and *LoadAmount* for UP-AR is 0.19, whereas for AR it is 0.34.

Further, using the CARLA package, we generated recourses for a set of 1000 individuals and Γ for two continuous features was randomly selected from $[0.3, 0.6, 0.9]$. Figure 5 provides a visual analysis of the distribution of average *pRMSE* using violin plots. The experiments were performed on the 3 datasets discussed in Section 5 for both the LR and NN models. For *GMSC* dataset, F_{con} = {*DebtRatio, MonthlyIncome*} and F_A = {*RevolvingUtilizationOf Un-securedLines, NumberOfTime30-59DaysPastDueNotWorse, DebtRatio, MonthlyIncome, NumberOfOpenCreditLinesAndLoans, NumberOf-Times90DaysLate, NumberRealEstateLoansOrLines, NumberOfTime60-89DaysPastDueNotWorse*}. For *COMPAS* dataset, F_{con} = {*priors-count, length-of-stay*} and F_A = {*two-year-recid, priors-count' length-of-stay*}. For *Adult* dataset, F_{con} = {*education-num, capital-gain*} and F_A = {*education-num, capital-gain, capital-loss, hours-per-week, workclass-Non-Private, workclass-Private, marital-status-Married, marital-status-Non-Married, occupation-Managerial-Specialist, occupation-Other*}.

With these experiments we conclude that UP-AR's $\hat{\Gamma}$ deviation from the user's Γ is consistently lower than the existing recourse generation methodologies. We observe that AR is unaffected by the varying user preference due to the fact that AR and other state-of-the-art recourse methodologies lack the capability of capturing such idiosyncrasies. On the other hand, UP-AR is driven by those preferences and has significantly better *pRMSE* in comparison to AR.

5.2 Cost Correction analysis

In Table 5 we explore the effect of UP-AR's cost correction procedure on the Adult and COMPAS datasets. We do not include the GMSC dataset as it does not include binary features, and therefore does not utilize the cost correction procedure. In Table 5 we show the number of factuals, the percentage of factuals for which recourse was found, the percentage of recourse found which contained at least one binary action, the percent of recourse found which underwent cost correction, the average percentage of steps saved by the cost correction procedure, and the average percent of cost savings, measured as the percent reduction in continuous cost

Table 5: The Frequency and Effect of Cost Correction

Metrics	Adult	COMPAS
Number of Factuals	1000	568
Success Rate	79.3%	99.6%
Percent of Recourse with a Binary Action	71.9%	82.6%
Percent of Recourse with Cost Correction	38.4%	25.5%
Average Percentage of Steps Saved	67.9%	63.5%
Average Percentage of Continuous Cost Saved	83.1%	76.0%

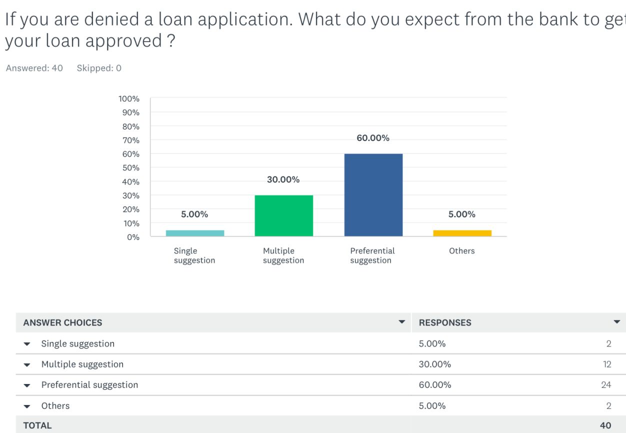


Figure 6: Snapshot of the human acceptance survey.

(l_2 distance) between a factual and its recourse before and after the cost-correction procedure.

6 CONCLUDING REMARKS

In this study, we propose to capture different forms of user preferences and propose an optimization function to generate actionable recourse adhering to such constraints. We further provide an approach to generate a connected [15] recourse guided by the user. We show how UP-AR adheres to soft constraints by evaluating user satisfaction in fractional cost ratio. We emphasize the need to capture various user preferences and communicate with the user in comprehensible form. This work motivates further research on how truthful reporting of preferences can help improve overall user satisfaction.

7 USER ACCEPTANCE SURVEY

We surveyed 40 random students and employees from a mailing list. The goal of this survey is to establish whether people preferred to provide specific preferences over other mechanism. The survey included one question with four options as follows:

If you are denied a loan application. What do you expect from bank to get your loan approved ?

(1) *Single list of suggestions to your profile. Ex: (increase income by 100$ & reduce loan duration by 1 year)*
(2) *A set with multiple lists of suggestions to your profile. Ex: (i) increase income by 100$ and reduce loan duration by 1 year OR ii) increase income by 500$ OR iii) reduce loan duration by 3 year OR iv) bring a co-applicant)*
(3) *Influence bank's suggestions by providing preferential scores for actions you can take. Ex: (preferring to increase loan duration more than loan amount by 8:2, or preferring to bring a guarantor before a co-applicant)*
(4) *Any other form of preferences*

Every individual in the survey was asked to select one of the four choices provided. In this survey, it is identified that majority of 60% of individuals preferred influencing the bank's decision by providing preference scores for individual features, followed by 30% of individuals who wanted multiple recourses from the bank. The remaining 10% of individuals preferred a single recourse or any other form of preference.

ACKNOWLEDGMENTS

This work is partially supported by the National Science Foundation (NSF) under grants IIS-2143895 and IIS-2040800, and CCF-2023495.

REFERENCES

[1] Ifeoma Ajunwa, Sorelle Friedler, Carlos E Scheidegger, and Suresh Venkatasubramanian. 2016. Hiring by algorithm: predicting and preventing disparate impact. *Available at SSRN* (2016).
[2] Julia Angwin, Lauren Kirchner, Jeff Larson, and Surya Mattu. 2016. Machine Bias: There's software used across the country to predict future criminals. And it's biased against blacks. (2016).
[3] Javier Antoran, Umang Bhatt, Tameem Adel, Adrian Weller, and José Miguel Hernández-Lobato. 2021. Getting a {CLUE}: A Method for Explaining Uncertainty Estimates. In *International Conference on Learning Representations*. https://openreview.net/forum?id=XSLF1XFq5h
[4] Kevin Bache and Moshe Lichman. 2013. UCI Machine Learning Repository. http://archive.ics.uci.edu/ml
[5] Furui Cheng, Yao Ming, and Huamin Qu. 2020. Dece: Decision explorer with counterfactual explanations for machine learning models. *IEEE Transactions on Visualization and Computer Graphics* 27, 2 (2020), 1438–1447.
[6] Alexandra Chouldechova, Diana Benavides-Prado, Oleksandr Fialko, and Rhema Vaithianathan. 2018. A case study of algorithm-assisted decision making in child maltreatment hotline screening decisions. In *Conference on Fairness, Accountability and Transparency*. PMLR, 134–148.
[7] Zhicheng Cui, Wenlin Chen, Yujie He, and Yixin Chen. 2015. Optimal action extraction for random forests and boosted trees. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*. 179–188.
[8] Giovanni De Toni, Paolo Viappiani, Bruno Lepri, and Andrea Passerini. 2022. Generating personalized counterfactual interventions for algorithmic recourse by eliciting user preferences. *arXiv preprint arXiv:2205.13743* (2022).
[9] Dheeru Dua and Casey Graff. 2017. UCI Machine Learning Repository. http://archive.ics.uci.edu/ml
[10] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572* (2014).
[11] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *International conference on machine learning*. PMLR, 1321–1330.
[12] Kaggle. 2011. Give Me Some Credit. https://www.kaggle.com/c/GiveMeSomeCredit/

[13] Amir-Hossein Karimi, Gilles Barthe, Bernhard Schölkopf, and Isabel Valera. 2021. A survey of algorithmic recourse: contrastive explanations and consequential recommendations. *ACM Computing Surveys (CSUR)* (2021).

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

[15] Thibault Laugel, Marie-Jeanne Lesot, Christophe Marsala, and Marcin Detyniecki. 2019. Issues with post-hoc counterfactual explanations: a discussion. *ArXiv* abs/1906.04774 (2019).

[16] Thibault Laugel, Marie-Jeanne Lesot, Christophe Marsala, Xavier Renard, and Marcin Detyniecki. 2017. Inverse Classification for Comparison-based Interpretability in Machine Learning. *stat* 1050 (2017), 22.

[17] Divyat Mahajan, Chenhao Tan, and Amit Sharma. 2019. Preserving causal constraints in counterfactual explanations for machine learning classifiers. *arXiv preprint arXiv:1912.03277* (2019).

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

[19] Martin Pawelczyk, Chirag Agarwal, Shalmali Joshi, Sohini Upadhyay, and Himabindu Lakkaraju. 2022. Exploring counterfactual explanations through the lens of adversarial examples: A theoretical and empirical analysis. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 4574–4594.

[20] Martin Pawelczyk, Sascha Bielawski, Johannes van den Heuvel, Tobias Richter, and Gjergji Kasneci. 2021. CARLA: A Python Library to Benchmark Algorithmic Recourse and Counterfactual Explanation Algorithms. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual.*

[21] Martin Pawelczyk, Klaus Broelemann, and Gjergji Kasneci. 2020. Learning model-agnostic counterfactual explanations for tabular data. In *Proceedings of The Web Conference 2020.* 3126–3132.

[22] Rafael Poyiadzi, Kacper Sokol, Raul Santos-Rodriguez, Tijl De Bie, and Peter Flach. 2020. FACE: feasible and actionable counterfactual explanations. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society.* 344–350.

[23] Kaivalya Rawal and Himabindu Lakkaraju. 2020. Beyond individualized recourse: Interpretable and interactive summaries of actionable recourses. *Advances in Neural Information Processing Systems* 33 (2020), 12187–12198.

[24] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why should i trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining.* 1135–1144.

[25] Chris Russell. 2019. Efficient search for diverse coherent explanations. In *Proceedings of the Conference on Fairness, Accountability, and Transparency.* 20–28.

[26] Leslie Scism. 2019. New york insurers can evaluate your social media use - if they can prove why it's needed. [Online; accessed January-2019].

[27] Ravi Shroff. 2017. Predictive Analytics for City Agencies: Lessons from Children's Services. *Big Data* 5, 3 (2017), 189–196. https://doi.org/10.1089/big.2016.0052 arXiv:https://doi.org/10.1089/big.2016.0052 PMID: 28829624.

[28] Naeem Siddiqi. 2012. *Credit risk scorecards: developing and implementing intelligent credit scoring.* Vol. 3. John Wiley & Sons.

[29] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. In *2nd International Conference on Learning Representations, ICLR 2014.*

[30] Berk Ustun, Alexander Spangher, and Yang Liu. 2019. Actionable recourse in linear classification. In *Proceedings of the conference on fairness, accountability, and transparency.* 10–19.

[31] Sandra Wachter, Brent Mittelstadt, and Chris Russell. 2017. Counterfactual explanations without opening the black box: Automated decisions and the GDPR. *Harvard Journal of Law and Technology* 31 (2017), 841.

[32] Prateek Yadav, Peter Hase, and Mohit Bansal. 2021. Low-cost algorithmic recourse for users with uncertain cost functions. *arXiv preprint arXiv:2111.01235* (2021).