

REACT: REcourse Analysis with Counterfactuals and Explanation Tables

by

Anastasiia Avksientieva

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Data Science

Waterloo, Ontario, Canada, 2025

© Anastasiia Avksientieva 2025

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Machine learning models often exhibit not only explicit bias—unequal performance metrics across subgroups—but also implicit bias, where altering a model’s prediction is disproportionately difficult across subgroups. In this work, we investigate two complementary approaches to analyze ways to overturn a model’s decision to achieve a desired label: modifying test input features and unlearning a set of training samples.

The novelty of our solution lies in combining these two methods with data summarization via informative rule mining that highlights biased subgroups. We demonstrate the value of REACT by allowing users to detect a model’s implicit bias and compare the biases of different model versions. The resulting framework is flexible, supporting the definition of practical constraints on feature-level interventions—for example, by limiting changes to modifiable attributes.

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Professor Lukasz Golab, for his research guidance and for continually bringing and sharing ideas to address challenges in explainable AI. I greatly appreciate his mentorship and support in publishing my first paper.

I am also deeply thankful to Professors Parke Godfrey and Jarek Szlichta for their consistent support throughout my studies and for always providing insightful feedback that helped me improve my work.

Dedication

This thesis is dedicated to my father, who has always inspired me to question and explore the peculiarities of our world through mathematics.

Table of Contents

Author's Declaration	ii
Abstract	iii
Acknowledgements	iv
Dedication	v
List of Figures	viii
List of Tables	ix
1 Introduction	1
1.1 Problem Formulation	2
1.2 Contributions	3
1.3 Thesis Structure	4
2 Foundations and Related Research	5
2.1 Feature-Based Recourse	5
2.2 Data-Based Recourse	8
2.3 Summarizing Recourse Paths	11
2.3.1 Informative Rule Mining	14

3	Methodology	16
3.1	Feature Modifications Workflow	17
3.1.1	Redundancy Removal	19
3.1.2	Workflow Example	20
3.2	Data Unlearning Workflow	21
3.2.1	Workflow Example	24
4	Experiments	26
4.1	Implementation Details	26
4.2	Datasets and Model Training	26
4.3	Feature-Based Recourse	27
4.3.1	Recourse Disparities in Salary Predictions	27
4.3.2	Fairness-Driven Model Comparison	29
4.3.3	Fairness in Pathways for Strip Decisions	31
4.4	Comparison with FACTS	32
4.5	Data-Based Recourse	36
5	Conclusion	42
5.1	Future Work	43
	References	45

List of Figures

2.1	Example of a local counterfactual explanation	7
2.2	Datamodeling framework	10
2.3	Data-centric explanation pipeline	13
2.4	Top-5 subsets attributable to statistical disparity in German Credit dataset [42]	14
3.1	The architecture of REACT.	17
3.2	The workflow of summarizing recourse cost of feature modifications.	20
3.3	The workflow of data-based REACT: summarizing recourse support metric.	25
4.1	The feature-based REACT parameter selection.	28
4.2	Subpopulation identified by FACTS with highest unfairness in aggregate effectiveness between female and male subgroups (equivalent to recourse availability in REACT).	33
4.3	Subpopulation identified by FACTS with high unfairness in macro-level choice for recourse	35
4.4	Histogram for the target distribution of data recourse support for adults labeled as low-income by explicitly biased model	37
4.5	Histograms for the target distribution of recourse support for arrested citizens labeled as strip searched (on the left) and as not strip searched (on the right)	40

List of Tables

2.1	An example explanation table	15
4.1	Recourse availability explanation table when changing <i>education level</i> to generate counterfactuals.	28
4.2	Recourse cost, changing <i>workclass</i> and <i>occupation</i>	29
4.3	Comparison of recourse availability when changing <i>hours per week</i> between the first model (explicitly biased, left) and the second model (right).	30
4.4	Recourse choice explanation table when modifying <i>arrest location</i>	31
4.5	Informative subgroups on recourse availability obtained by changing <i>education level</i> and <i>hours-per-week</i>	34
4.6	Informative subgroups on recourse choice obtained by changing <i>education level</i> , <i>hours-per-week</i> and <i>workclass</i>	36
4.7	Summary of data recourse support for low-income adults, first model (explicitly biased).	37
4.8	Summary of data recourse support for low-income adults, second model.	38
4.9	Summarizing a target indicating how frequently a given adult income training sample is being “machine-unlearned”.	39
4.10	Summarizing a target indicating how frequently a given police data training sample is being “machine-unlearned”.	40
4.11	Summary of data recourse support for arrestees predicted to be strip searched. Patterns reveal subgroups requiring significantly more or fewer samples to be unlearned to achieve a positive outcome, such as Black males.	41
4.12	Summary of data recourse support for arrestees predicted to be not strip searched. Top patterns do not contain protected attributes.	41

Chapter 1

Introduction

Before deploying machine learning models in production, it is critical to assess their performance on unseen data. Based on recent interviews with machine learning engineers, it has been observed that practitioners routinely “slice and dice” model predictions across sub-populations to uncover systematic mistakes, identify data points similar to these failures, and use continually updated validation sets designed to capture both distribution shifts and localized failures in overlooked subgroups [41]. Recent approaches such as **Model Slicing** [9], **InfoMoD** [11], and **CAMO** [47] enable such model diagnostics by identifying underperforming subgroups using metrics such as accuracy, F1 score, or ROC-AUC. For instance, such methods might find that a new version of a healthcare model performs better overall than the previous one, but more poorly for younger patients than for older patients. In high-stakes applications, it is critical to identify biases and take corrective actions such as collecting more training data or unlearning subsets that reinforce disparities in model inference.

Recent work has observed that, in addition to *explicit* biases in prediction fairness, models may suffer from *implicit* biases [3, 29]. Explicit bias refers to disparities in a model’s predictions that can be measured using standard fairness metrics, such as differences in accuracy, false positive/negative rates, demographic parity, also referred to as statistical parity (similar rates of favorable outcomes across sensitive groups), or equalized odds (predictions are conditionally independent of the protected attribute given the true label [22]). Implicit bias, on the other hand, can be captured by *recourse* metrics, which quantify the extent to which individuals can either take meaningful actions or require changes to the training data in order to obtain favorable outcomes from a model. For example, a credit approval model might appear explicitly unbiased for men and women but still exhibit implicit disparities: women whose applications are denied may need to add 20 percent

more to their savings than men to be approved, and they may require the removal of more historically biased training examples to flip the outcome.

1.1 Problem Formulation

There are two ways in which a model’s decision may change for a given example to provide recourse: modifying the (features of the) example or modifying the model itself, for example by re-learning it on a different training set. We therefore consider two strategies for identifying implicit biases: evaluating feature-based and data-based recourse. The dual perspective on recourse provides a more complete picture of model bias than either approach alone. *Feature-based recourse* can be measured by a notion of burden, or distance to a *counterfactual example*—a hypothetical instance with modified feature values that flips the model’s decision from an undesirable to a desirable outcome. For example, increasing salary to overturn a loan rejection reflects how far an instance must move in feature space to achieve a favorable prediction. *Data-based recourse* assesses a model’s recourse brittleness by estimating *data counterfactuals* [30, 24]—hypothetical modifications to the training set that would alter the model’s prediction, typically by identifying which and how many training instances must be removed to induce such a change. This formulation draws on the broader idea of *machine unlearning*—removing the influence of specific training data from a model, with researchers exploring efficient approximation techniques to estimate the effect of such removals on model outputs. [6, 4]

Existing work on recourse analysis typically focuses on feature-based recourse, ranking subgroups of a test set by recourse distance—that is, the distance in feature space between test inputs and their corresponding counterfactual examples[17]—or by other related metrics [3, 29]. Since there can be many such subgroups, machine-learning engineers require tools that can summarize recourse analytics to ensure that implicit biases do not go unnoticed. Likewise, end users can benefit from recourse summarization tools to build trust in model outcomes, especially in mission-critical fields such as law enforcement, healthcare and finance. There remains a gap in summarization tools for data-based recourse, where brittleness arises from dependencies on specific training examples. While recent work explores how deletions affect individual predictions [24, 37], tools for detecting and explaining systematic training data effects across groups are still lacking.

1.2 Contributions

To fill these gaps, we present REACT, a tool for *REcourse Analysis with Counterfactuals and Explanation Tables*. An overview of the system architecture is shown in Figure 3.1. Given a test dataset, REACT computes *recourse paths* (*feature or data counterfactuals*) for each example, and summarizes the recourse statistics using the recent work on informative rule mining (*explanation tables*) [18, 47]. We make the following contributions:

1. Summarizing Recourse Diagnostics.

On the conceptual side, we introduce the new problem of summarizing recourse fairness. We propose a modular system architecture that decouples the process of identifying recourse paths from the process of summarizing these paths. We also incorporate the dual problem of the cost or effort required to flip a model’s decision from the desirable to the undesirable class. This can provide an indication of model stability, to complement the implicit bias analysis via recourse distance (as illustrated in Section 4.3.3).

Moreover, to the best of our knowledge, REACT is the first to **summarize data-based recourse**, capturing how easily model predictions can be altered through machine unlearning. This allows us to apply the rule mining techniques to uncover patterns of recourse brittleness.

2. Bridging Counterfactuals and Explanation Tables.

On the technical side, we materialize the above design in REACT, with a focus on binary classifiers. To address the challenge of summarizing recourse diagnostics, REACT combines feature-based and data-based counterfactual explanations with explanation tables, which summarize patterns using informative rule mining. This fusion improves the interpretability and actionability of the summaries. For instance, REACT can identify subgroups where achieving recourse via unlearning is more or less likely compared to the dataset average or uncover subgroups with multiple viable feature-based recourse options (such as either putting more money in a savings account or increasing one’s monthly salary to flip a loan-denied decision to loan-approved).

For efficiency of feature-based recourse, we employ a fast counterfactual mining method that samples from the space of candidate counterfactuals [36]. To achieve scalability in data-based recourse calculations, REACT leverages an efficient *data attribution* approach—methods that estimate how individual training examples influence model predictions—as a proxy for retraining-based recourse.

3. Demonstrating REACT.

We describe the REACT user experience with several classifiers trained on police search and income prediction datasets. Our analysis demonstrates that even equal or fair accuracy rates may still lead to disparities, such as unequal recourse distance, where affected subgroups can be summarized by REACT.

To summarize, we introduce a novel approach to fairness diagnostics. Unlike tools such as InfoMoD [11] that summarize explicit biases in model predictions, REACT investigates an equally critical dimension of implicit bias that may not be evident through model accuracy analyses. In our work, we also provide a detailed comparison with FACTS [29] that proposes various feature-based recourse bias definitions to reveal vulnerable subpopulations—we instead focus on a concise presentation of bias analytics, highlighting “surprising” subgroups whose recourse statistics deviate notably from the average. REACT further advances this space by being the first to summarize the effects of machine unlearning for implicit bias analysis—extending ideas from prior work, which assesses training data influence on a per-sample basis.

1.3 Thesis Structure

The following chapter provides an overview of background concepts and prior work on feature and data attribution, along with methods for generating counterfactual explanations and mining informative patterns. Chapter 3 details our contributions and the algorithms used to achieve the purpose of the thesis. Chapter 4 showcases the application of our conceptual framework to real-world datasets that incorporate protected variables. In Chapter 5, we discuss the constraints of our research and identify possible extensions for further study.

Chapter 2

Foundations and Related Research

This chapter introduces the foundations behind REACT’s dual approach to recourse: modifying input features (feature-based recourse, detailed in Section 2.1) or altering the training data (data-based recourse, detailed in Section 2.2). Since REACT summarizes recourse outcomes across subgroups, we also discuss techniques for explainable aggregation in Section 2.3.

We begin by defining notation. Let $TD = \{z_i = (f_i, l_i)\}_{i=1}^d$ be the training dataset of size d , where each sample consists of a feature vector f_i from a feature space F , and l_i is the corresponding label. For a specific sample z_i , its k -th feature is written as $f_i[k]$. Similarly, the diagnostics dataset is $DD = \{x_j\}_{j=1}^n$. Depending on its purpose, its samples may or may not contain a label. We define A as the learning algorithm that maps TD to a trained model M , and \hat{l}_j as the predicted label for a target sample x_j .

2.1 Feature-Based Recourse

Feature-based recourse relies on modifying the values of the features of a given example in order to achieve algorithmic recourse. Formally, for a given example, a *counterfactual* is a synthetic example, with some feature values perturbed, that leads to a different model prediction. Since there can be many potential counterfactuals, some with many features changed, a *minimal* counterfactual is one with the fewest features modified to flip the model’s prediction.

Since there can be many possible counterfactuals for any given example (any feature can be modified in any way), practical counterfactual generation methods employ some

pruning strategies to reduce this exponential search space. As a result, these methods produce approximate solutions, in the sense that they may miss some counterfactuals or return some that are not minimal [44, 39, 27, 5].

We categorize counterfactual generation methods into black-box approaches, which rely solely on query access to a model’s API (e.g., predictions), and white-box approaches, which additionally require access to the model’s internal parameters. For instance, DiCE [36] and the method by Wachter et al. [45] support both gradient-based and model-agnostic modes, while FACE [39] and MACE [27] are model-agnostic. Gradient-based counterfactual methods frame recourse generation as a continuous optimization problem, making them efficient for differentiable models. Some other model-specific approaches to recourse analysis estimate the exact distance to a decision boundary to assess robustness or the feasibility of recourse [21, 44]. These approaches typically rely on assumptions of linearity or convexity, which restricts their applicability in tools like REACT. In contrast, model-agnostic counterfactual methods avoid such assumptions but suffer from a different shortcoming: they cannot certify infeasibility. As observed in local approximators for non-linear models [44], these methods may fail to identify recourse not because none exists, but due to incomplete search or reliance on heuristics. While problematic in prescriptive settings, this is acceptable in REACT, a global diagnostic tool.

Model-agnostic methods use a variety of strategies without requiring internal access to the model. One approach is to sample from the space of possible counterfactuals, using techniques such as random perturbation or derivative-free optimization. Others enlist the user to define domain constraints, framing the task as a constraint satisfaction problem where the goal is to identify feature combinations that adhere to specified rules [27]. When access to training data is available, methods may also leverage real examples to guide the search toward plausible perturbations, such as the graph-based approach explained later in this section. However, staying within the training data distribution can miss valid counterfactuals, that do not exist in the training set, yet correspond to feasible feature perturbations.

When counterfactual generation is framed as an optimization problem, the goal is to formulate a loss function that balances multiple objectives, the primary one being to identify minimal changes to an input’s features that result in a different model prediction. [35]. One common formulation, introduced by Wachter et al. [45], defines the loss as a combination of two objectives: proximity to the original input—typically measured using L1 or L2 norms—and prediction validity, i.e., how close the model’s output for the counterfactual is to the desired label. For example, in a binary setting, if a loan application is rejected, this formulation seeks a new feature vector—such as increasing income or reducing debt—such that the model predicts approval while keeping changes to the original profile minimal.

Extending this approach, Mothilal et al. developed DiCE [36], which augments the loss function by incorporating *diversity* among the generated counterfactuals—that is, favoring multiple distinct recourse options rather than near-duplicates. To achieve this, DiCE adds a regularization term based on determinantal point processes, a probabilistic model that promotes sets of points that are well-spread in feature space. Moreover, their method promotes *sparsity*—that is, counterfactuals that modify as few features as necessary—through a post-processing step applied to the candidate set.

In addition to its optimization-based approach, DiCE includes random sampling (implemented as DiceRandom), an approach which generates candidate counterfactuals by randomly sampling points near the original input in the feature space and evaluating whether they lead to the desired prediction. It does not optimize the loss function directly, but still applies the same sparsity-promoting post-processing to filter the candidate set. However, depending on the chosen parameters, the candidate set may still include redundant examples—that is, examples where one is subsumed by another, as illustrated in Figure 2.1. In this local counterfactual explanation for a selected individual, the income prediction is flipped by changing two ordinal features: education level and hours worked per week. In the obtained counterfactual set, candidate 1 subsumes candidate 0 by achieving the prediction flip with cheaper changes—that is, smaller feature shifts—while maintaining the same level of sparsity (i.e., the number of modified features is the same). We address this limitation in the Methodology chapter.

	age	workclass	education-num	marital-status	occupation	relationship	race	sex	hours-per-week	native-country	income	
0	0		2		9	2	4	0	2	1	38	0

Diverse Counterfactual set (new outcome: 1)

	age	workclass	education-num	marital-status	occupation	relationship	race	sex	hours-per-week	native-country	income	
0	-	-			15	-	-	-		4	-	1
1	-	-			15	-	-	-		3	-	1

Figure 2.1: Example of a local counterfactual explanation

Another notable counterfactual generation method is FACE by Poyiadzi et al. [39]. FACE builds a graph over training data points by connecting nearby instances using k-nearest neighbors or similar proximity-based rules. It then applies a shortest-path algorithm (e.g., Dijkstra’s) to find a counterfactual that lies along a feasible, high-density path—that is, a sequence of small, realistic steps through regions well supported by the training data. This ensures that the results resemble real data points, making FACE well-suited for settings where out-of-distribution suggestions are undesirable. However, its

reliance on the observed data neighborhoods limits scalability in sparse or high-dimensional feature spaces, and performance may degrade in noisy or low-coverage domains.

Several works build on counterfactual explanations to define *recourse metrics* that quantify the burden required for individuals to achieve favorable outcomes. For example, counterfactual fairness assesses whether an individual’s prediction would remain the same had they belonged to a different demographic group [31]. Actionability—the extent to which suggested changes are feasible and relevant for an individual—is a key consideration in measuring recourse: some methods constrain counterfactual feature changes to reflect domain knowledge and user-specific constraints [26], or formulate recourse directly as an optimization problem under feasibility constraints [44, 28].

Kavouras et al. [29] introduce cost-oblivious fairness metrics, avoiding the need for predefined cost functions that often require domain expertise. Instead, they evaluate the diversity of valid feature changes—e.g., whether individuals across subgroups are offered the same number of effective recourse options. Bell et al. [2] propose metrics that account for disparities in starting conditions and recourse dynamics over time: effort-to-recourse and time-to-recourse, which quantify the cumulative burden and the additional time needed to achieve a positive outcome. However, computing them requires simulating individual trajectories over time steps.

Karimi et al. [28] emphasize that recourse involves not only understanding decision changes through counterfactual explanations but also considering actionable interventions. They use causal reasoning to model how interventions propagate through the system, ensuring that suggested changes are both actionable and realistic. The goal of our work, however, is not to prescribe interventions but to evaluate and summarize recourse through several metrics. We also do not assume that users of our framework have access to a causal model structure. Instead, REACT accommodates real-world constraints by offering flexible configuration of feasibility conditions.

2.2 Data-Based Recourse

Data-based recourse quantifies the impact that removing or altering training samples has on model predictions—by doing so, we adjust the model’s decision boundary to achieve desirable outcomes. This perspective captures model brittleness: if a prediction changes after the removal of only a few examples, the decision is less stable and more dependent on specific training data. Direct retraining after removing training points provides a ground-truth estimate of recourse, but involves exploring an exponential search space of possible

subsets—making it computationally infeasible in practice, similar to the intractability of searching all possible feature perturbations explained in Section 2.1. To overcome this, researchers turn to data attribution methods—they estimate a model’s dependence on specific training points with little or no retraining. Some of these methods can approximate the effects of machine unlearning and are particularly well-suited for estimating data-based recourse.

Data attribution, or instance attribution, has become one of the key approaches for addressing fairness in model development [23]. One widely used technique in this area is influence functions [30], which estimate the impact of removing a single training sample by infinitesimally upweighting it and observing the effect on the loss for a given test point, using gradient-based approximations derived from the loss function’s derivatives. However, high-influence samples do not always yield useful insights [24]; for example, instead of revealing training examples similar to a given example (potentially offering a target for data cleaning), they often highlight redundant ones. Pezeshkpour et al. [38] found that simple similarity-based methods, which compare training and test samples based on their representations in the model’s embedding space (specifically, the penultimate network layer), can be surprisingly effective. In some cases, these methods perform better than computationally expensive gradient-based influence functions – even though there’s no guarantee that similarity reflects influence, this straightforward approach successfully identifies influential training points. Beyond similarity-based techniques, other data attribution methods rely on Shapley values [25, 19], which assign importance scores to training samples by fairly distributing a model’s output based on their contribution to a prediction, or empirical influence, which measures the degree of label memorization by the learning algorithm [13].

Ongoing advances in fast model training with GPU power have allowed Ilyas et al. [24] to develop an efficient method of estimating data influence: a datamodeling framework. More recently, TRAK [37] has proven to be as reliable as datamodels while offering an even faster approach, making it suitable for large-scale models, though limited to differentiable ones.

The core idea behind datamodeling is to learn a simple linear surrogate function that maps training samples to their attribution scores, reflecting their influence on the model’s output for a target example x . Formally, this surrogate function is denoted as $g_\theta(\mathbb{1}_{TD'}) : \{0, 1\}^{|TD'|} \rightarrow \mathbb{R}$, where the binary vector $\mathbb{1}_{TD'}$ indicates which training samples are included in a particular subset TD' . The goal is for g_θ to approximate $M_A(x; TD')$ —an output of a model trained on TD' using learning algorithm A . To achieve this, we need to sample a variety of training sets TD_i using a chosen distribution, as described in [24]. This process, shown in Figure 2.2, consists of two steps:

1. Training m underlying models on masked subsets of the dataset. For each model i , a mask is defined by an indicator function $\mathbb{1}_{TD_i}$ that specifies whether each training sample z belongs to it
2. Training a datamodel specifically for a target example x based on a loss function $\mathcal{L}(\cdot, \cdot)$. The learned parameter vector θ thus provides a per-sample attribution: each component θ_j estimates the contribution of training sample z_j to the model’s prediction on x .

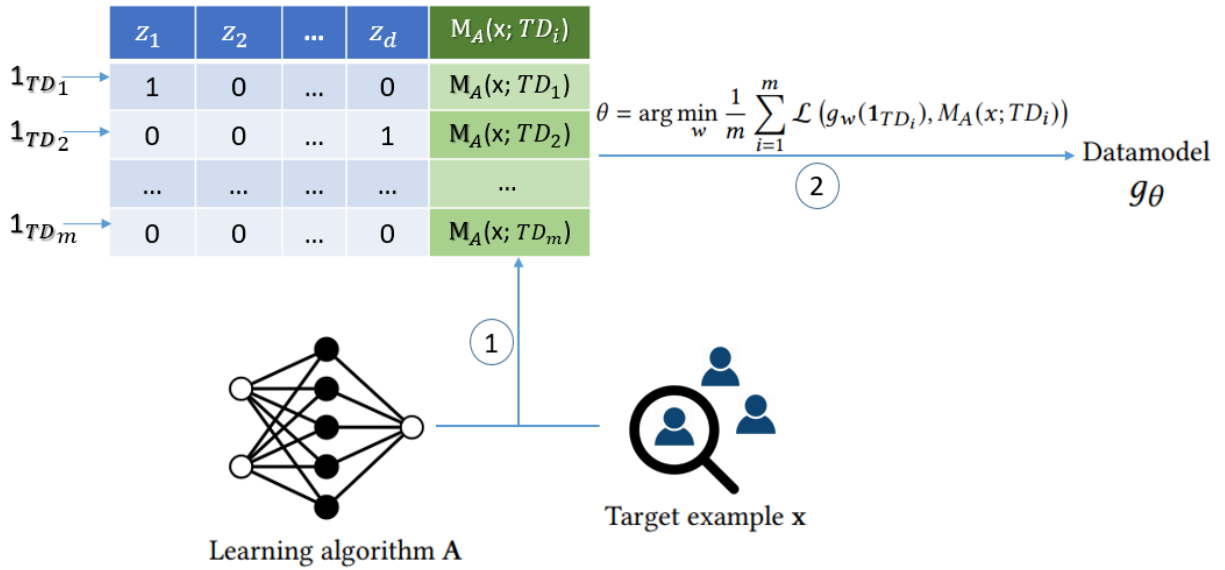


Figure 2.2: Datamodeling framework

The output of the model, used as the target label for datamodel training, can be defined in several ways—for example, as cross-entropy loss, predicted class probability, or a binary correctness indicator. Ilyas et al. use correct-class margin, motivated by empirical results showing that it yields residuals that are approximately normally distributed. It quantifies the confidence of the model’s prediction by measuring the difference between the logit of the correct class and the highest logit among incorrect classes:

$$M_A(x, TD') = \text{logit}_{\text{correct}}(x) - \max_{\text{incorrect } c} \text{logit}_c(x)$$

We refer to this quantity as the margin in subsequent chapters.

To generalize notation from [24], we assume g_θ is any data attribution function that assigns weights to training examples, estimating their contribution to the prediction for x . Then, data counterfactual, or the causal effect of removing a subset $R(x)$ of training examples, can be written as:

$$\mathbb{E}[M_A(x; TD) - M_A(x; TD \setminus R(x))] \approx g_\theta(\mathbf{1}_{TD}) - g_\theta(\mathbf{1}_{TD \setminus R(x)}),$$

Empirically, datamodel-based estimates—where the weights θ are coefficients of a linear function—were proven to align closely with retraining outcomes, even when evaluating the removal of out-of-distribution subsets from the training data.

2.3 Summarizing Recourse Paths

While individual counterfactuals show how outcomes can be changed for specific instances, they do not reveal broader trends. To assess fairness in **feature-based recourse** at the subgroup level, researchers often compute recourse metrics per sample and analyze top-k subsets ranked by these metrics [29]. However, this approach may overlook structure in the data. As an alternative, methods such as decision trees, if-then rules, or rule mining summarize and interpret recourse patterns across groups [10, 33, 40].

For example, GLOBE-CE identifies global bias by learning class-to-class feature translations and analyzing which feature changes are consistently required to achieve recourse; these translations are then expressed as interpretable **if-then** rules. Cornacchia et al. [10] use counterfactual reasoning to detect bias in models trained without sensitive features by modifying non-sensitive attributes to flip the model’s prediction. They then train a sensitive-feature predictor that checks whether the counterfactuals remain in the same group, and the proportion of counterfactuals assigned to a different sensitive group—captured by the Counterfactual Flips metric—indicates potential proxy-based discrimination.

Unlike previous works, FACTS [29] emphasizes analyzing recourse at the subpopulation level. Subpopulations in FACTS are defined by frequent patterns over feature-value pairs mined from individuals receiving an unfavorable prediction (the *affected population*). Specifically, FACTS applies frequent itemset mining to extract predicates p that define meaningful subpopulations G_p . Each subpopulation is then partitioned into *subgroups* $G_{p,v}$, where $v \in \text{dom}(F_k)$ corresponds to a value of a protected attribute F_k (e.g., race or age group). To identify recourse paths, FACTS uses the *FP-growth algorithm* to mine frequent action sets from the *unaffected population*—individuals already receiving favorable outcomes—ensuring that proposed counterfactual changes are realistic and commonly observed. We compare FACTS with REACT in our experiments in Section 4.4.

FACTS also introduces several novel definitions of *micro-* and *macro-level aggregations* of recourse. Micro-level metrics are computed individually for each member of a subpopulation (e.g., whether an individual achieves recourse or how much it costs), and then averaged within the group. Macro-level metrics, in contrast, evaluate the effectiveness of applying the same counterfactual action to an entire subgroup. This reflects a stricter criterion: for example, if we apply a fixed change (e.g., “increase education by one level”) to all individuals in a subgroup, how many of them will actually achieve a positive outcome? For example, their *Equal Choice for Recourse* definition considers a classifier fair if the protected subgroups $G_{p,1}$ and $G_{p,0}$ are offered the same number of sufficiently effective recourse options—as defined by the metric introduced earlier in Section 2.1. An action is sufficiently effective if it succeeds for at least $\phi\%$ of the subgroup. As will be shown in the experiments (Figure 4.3), the top-ranked subpopulation—defined by pattern p , which corresponds to the first line in the figure that starts with “If”—shows that the female subgroup $G_{p,\text{female}}$ has no sufficiently effective recourse options, while the male subgroup does.

In contrast to feature-based recourse, ***data-based recourse*** has not been directly addressed in prior summarization frameworks. However, recent work on global, example-based explanations, such as those using machine unlearning, has aimed to capture training-data-driven causes of explicit bias. Figure 2.3 illustrates the two core components required to generate data-centric explanations. Instance attribution methods assign importance scores (weights) to training samples, similar to how feature attribution methods assign importance to input features. While local interpretations may identify similar training examples for a given test input or estimate data counterfactuals, global interpretations focus on patterns—such as recourse-related ones—derived from the training or diagnostics dataset.

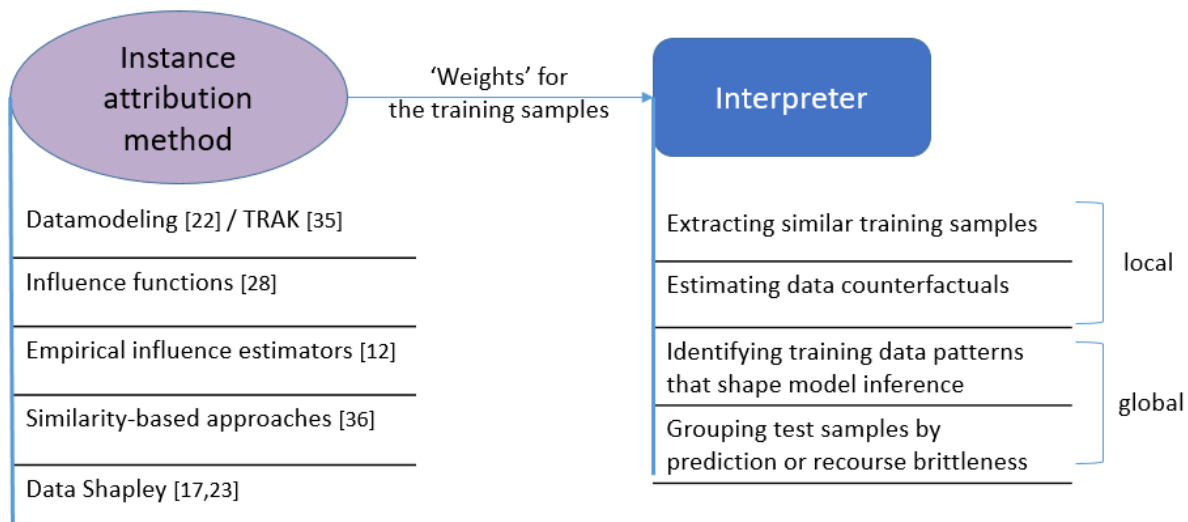


Figure 2.3: Data-centric explanation pipeline

One global approach on detecting explicit bias through instance attribution is FUME, proposed by Surve et al. [42] which identifies training data subsets responsible for fairness violations in non-parametric models, such as random forests. The method quantifies a subset’s contribution to group bias—measured using metrics like statistical parity and equalized odds—by estimating how bias is reduced when the subset is removed. To avoid retraining models for every candidate subset, FUME leverages machine unlearning specifically designed for random forests. To make this process tractable, they employ a lattice-based subset search inspired by frequent itemset mining and apply pruning rules - these rules remove irrelevant, overly complex, or low-support subsets. Figure 2.4 shows an example of interpretable subsets of training data that drive model bias in terms of statistical parity on the German Credit dataset, a benchmark dataset containing loan approval records with demographic and financial features.

Patterns	Support	Parity Reduction
Status of checking account = < 0 DM, Number of people liable = High	5.00%	97.79%
Savings = 100 ≤ ... < 500 DM, Job = Skilled employee / official	7.13%	95.58%
Installment plans = Bank, Debtors = None	12.00%	93.38%
Status of checking account = No checking account, Property = Unknown / no property	5.25%	91.17%
Housing = Rent, Status and sex = Female divorced/separated/married	10.00%	89.91%

Figure 2.4: Top-5 subsets attributable to statistical disparity in German Credit dataset [42]

2.3.1 Informative Rule Mining

To achieve an interpretable summary in REACT, we employ the rule mining method introduced by Gebaly et al. [18]. They apply a greedy, information-theoretic strategy to select a small number of overlapping patterns that together explain the distribution of a target variable, forming an explanation table. The method is designed to satisfy three key criteria: interpretability, by identifying overlapping patterns in data that highlight macro-level trends; informativeness, by enabling accurate micro-level reconstruction of the target attribute; and efficiency, through sampling-based heuristics that scale to large datasets. This makes it well-suited for summarizing recourse metrics, treating them as target variables in the explanation process.

An example output is shown in Table 2.1. We defer details about how the recourse cost target is constructed, as these will be covered in the next chapter on counterfactual generation and recourse metrics. Each rule, or pattern, in the explanation table corresponds to a subset of the train or test data—an informative subgroup.

To extract patterns that deviate from expected trends, the explanation table identifies the k most informative subgroups with respect to the user-specified target variable—in this case, recourse cost. In each iteration (constrained by k , the user-defined maximum number of patterns), the algorithm greedily selects the pattern that yields the greatest reduction in the Kullback-Leibler (KL)—a measure of how one probability distribution differs from another—between the true distribution of the target variable and its maximum-entropy approximation based on the selected patterns.

	age	race	sex	native-country	marital-status	recourse_cost	support
0	*	*	*	*	*	0.342	831
1	*	*	Female	*	*	0.411	127
2	>50 years	White	Male	*	*	0.351	210
3	*	*	*	US	Widowed	0.510	26

Table 2.1: An example explanation table

The result is a disjunction of patterns, where each pattern is a conjunction of attribute-value conditions. A star (*) indicates that an attribute can take any value. The last column, support, indicates the number of test samples matching each pattern. The first rule in Table 2.1 captures the overall average recourse cost (0.34). The next rule is chosen to provide the most additional information about the distribution of recourse cost: the pattern “*, *, female, *, *” indicates that females form a subgroup with a distinctly different (higher) recourse cost.

Chapter 3

Methodology

As illustrated in Figure 3.1, REACT examines recourse fairness via two modules:

- a *recourse finder* (Step ①); and
- a *recourse summarizer* (Step ②).

In Step ①, REACT generates counterfactuals for every example in DD . The input to REACT consists of the dataset DD with labels \hat{l} produced by M . The user selects the counterfactual label. For recourse burden analysis, the desirable class label is selected, and counterfactuals are generated to assess the cost of flipping the model’s decision to the desirable label. However, the user can also select the undesirable label as the counterfactual goal, to measure the cost of turning positive examples to negative ones. Without loss of generality, let $\hat{l} = 1$ be the desired counterfactual label in the remainder of this chapter. Depending on the type of analysis, the produced counterfactuals are either based on feature perturbations, or data counterfactuals. The formulation and evaluation of these two types are described separately in Sections 3.1 and 3.2, with illustrative workflows shown in Figures 3.2 and 3.3.

The output of Step ①—a scalar recourse metric, computed for each sample from either TD or DD —becomes the input to Step ②, the recourse summarizer. Step ② requires three parameters: the set of features to summarize, the maximum number of rules, or patterns, to include in each explanation table, and the minimum support required for a pattern to be added. The number of explanation tables returned corresponds to how many the recourse metrics were selected by the user.

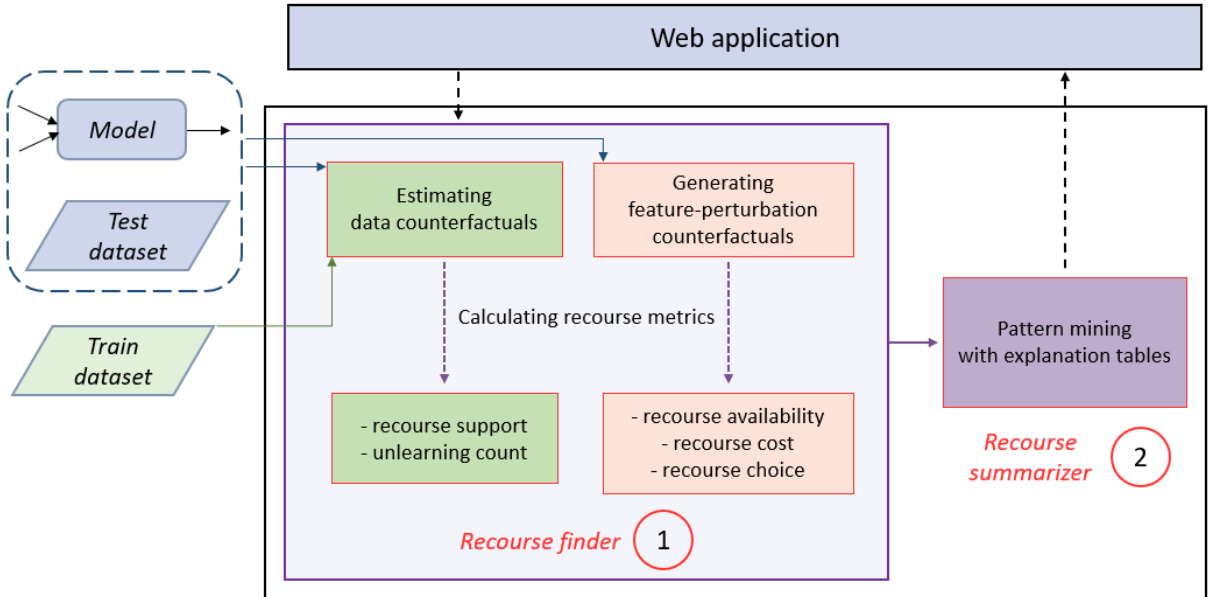


Figure 3.1: The architecture of REACT.

3.1 Feature Modifications Workflow

We divide feature space F into three disjoint subsets, F_C , the features that can be perturbed to generate counterfactuals—typically selected by users based on domain expertise to reflect features that can be feasibly modified; F_S , the features that will be used to construct subgroups of the test set whose recourse statistics will be compared; and the remaining features, F_R . In total, $F = F_C \cup F_S \cup F_R$, with the first two feature subsets selected by the user in the REACT web interface.

In Step ①, feature-based REACT generates counterfactuals for every example in T with $l = 0$, by perturbing the values of F_C in a way that changes M 's prediction to $\hat{l} = 1$. In our implementation, we use a model-agnostic version of DiCE [36], which performs a randomized search to generate a candidate set. Due to the stochastic nature of the sampling-based approximation, this approach may occasionally underestimate the number of viable recourse options or overestimate the burden (e.g., recourse cost) for individual instances. Nonetheless, the REACT back end remains compatible with any counterfactual generation technique, and users may substitute alternative methods—such as those

discussed in Chapter 2—if more precise or deterministic behavior is required.

Additionally, we filter the produced counterfactuals by removing the redundant ones. For example, “increase salary by 10%” and “increase salary by 10% and decrease credit card debt by 20%” are two redundant explanations, but “increase salary by 10%” and “increase salary by 5% and reduce debt by 5%” are not redundant, because the second involves a trade-off between smaller changes in multiple factors, offering a different path to recourse. The process is described later in this section.

Per individual sample, REACT computes the following metrics, that we have selected as the most insightful for our summarization purposes among those used in prior work on recourse analysis [17, 29]:

1. *Recourse Availability* equals one if there exists a counterfactual and zero otherwise (if no changes to F_C can produce a new hypothetical example that the model will predict as $\hat{l} = 1$).
2. *Recourse Cost* is the distance between the original example and its *nearest* counterfactual - we provide the formulation below.
3. *Recourse Choice* is the number of counterfactuals produced, representing the number of recourse options; e.g., if either increasing salary or decreasing debt leads to a rejected loan application being approved, then the recourse choice is two. Clearly, zero recourse availability implies zero recourse choice.

At the end of Step ①, every example is labeled with its recourse availability and choice. However, we calculate recourse cost only for samples with a non-empty set of counterfactuals. Otherwise, assigning an “infinite cost” would result in a long-tail distribution in Step ②, distorting the average values. The exclusion is not problematic, as individuals without recourse options are accounted for by the other two metrics.

The ambiguity in defining the recourse cost arises from the fact that the cost of modifying a feature depends on its type. Given a counterfactual x^{cf} of a sample x , we use an adaptation of Gower’s distance [20], suitable for mixed-type set of features F_C :

$$Recourse_cost(x_i, x_i^{cf}) = \frac{1}{|F_C|} \sum_{f[k] \in F_C} cost(f_i[k], f_i^{cf}[k])$$

The value of $cost(f[k], f^{cf}[k])$ equals a Manhattan distance, range- or rank-normalized, if k-th feature $f[k]$ is numerical or ordinal respectively. If $f[k]$ is categorical, the cost is 0

if the original and counterfactual values are the same, and 1 otherwise—that is, a binary matching coefficient.

The number of features to perturb affects the applicability of each metric. For instance, if we choose only one categorical feature to perturb, the cost would provide no additional information beyond recourse availability; the recourse choice, on the other hand, is not meaningful when only one numerical attribute is altered. Although perturbing more features allows for a larger number of potential changes that fuel the insights, it can also lead to a less interpretable summary. Additionally, some attributes may be more “expensive” to adjust than others—inequities can arise when individuals from disadvantaged subgroups are disproportionately encouraged to modify such features. Although Gower’s distance and similar measures can be adjusted with feature-specific weights to reflect real-world constraints, these coefficients must be chosen carefully, often requiring input from domain experts. Importantly, the weights reflecting the cost of changing a feature value are distinct from the concept of feature importance.

3.1.1 Redundancy Removal

This step impacts the outcome of recourse choice computation. Specifically, we perform pairwise comparisons of counterfactuals: given x^{cf_1} and x^{cf_2} , we remove x^{cf_1} if both the following conditions hold true:

1. All numerical and ordinal features of x^{cf_1} are greater than or equal to those in x^{cf_2} , in case the user intends to flip sample x to a favorable outcome and all features have monotonic direct relationship(*) with their respective costs of modification. In case of flipping to unfavorable outcome, all such features must be lower or equal than those in x^{cf_2} .
2. Categorical features of x^{cf_1} and x^{cf_2} are either identical or features of x^{cf_2} remain unchanged from the original sample.

(*) In redundancy exclusions for the current implementation of REACT, we assume that any numerical or ordinal feature has a monotonic direct relationship with the cost of its modification. In the other case, we would need the user to provide additional parameters and handle them in both redundancy removal and recourse cost calculation. For instance, for some numerical attributes—such as body weight, which may appear in healthcare domain—both increasing and decreasing its value may impose a burden on the individual or group.

3.1.2 Workflow Example

To illustrate the workflow, consider the test dataset for loan approval classification shown in Figure 3.2. The feature set F consists of Sex, Age, Ethnicity and Income. The Approved label is one if the loan was approved and zero otherwise. Suppose the user selects two sets: $F_C = \text{Income}$ and F_S be the remaining features. That is, if the individuals can modify Income to turn declined examples into approved ones via counterfactuals, what recourse patterns emerge within subgroups of the test set identified by Sex, Age and Ethnicity?

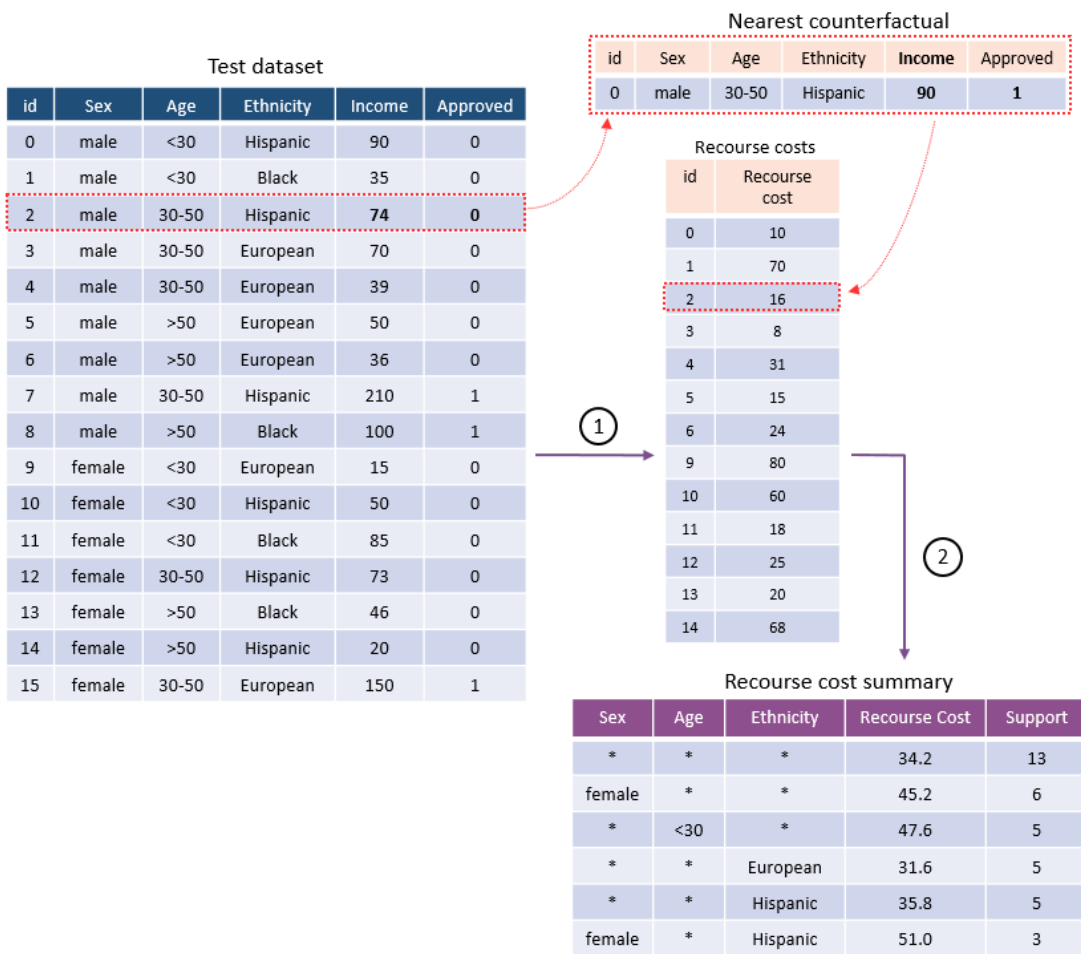


Figure 3.2: The workflow of summarizing recourse cost of feature modifications.

Consider the test example with id=2 and Income=74. The nearest counterfactual for

this example is shown in the top-right corner of Figure 3.2, with Income increased to 90, for a recourse cost of 16. Note that there can be multiple non-redundant counterfactuals if there are multiple features in F_C . Also note that this example did not normalize Income for simplicity, but REACT would do this before computing recourse cost.

Next, suppose we are interested in recourse cost analysis. We show a table with a single column of recourse cost computed for every example in the test set with the undesirable label of zero (and recourse availability one), of which there are 13.

Finally, in the bottom-right corner of Figure 3.2, we show the explanation table for recourse cost target, with $k = 6$ and minimum support of 20%. The first three columns are the features in F_S : Sex, Age and Ethnicity. The next column is the average recourse cost for each subgroup (defined below) reported in the explanation table. The last column is the support of (i.e., the number of examples with the undesirable label in) each subgroup.

The first row of the explanation table, all stars, corresponds to all examples with the undesirable label, where the average recourse cost is 34.2. The next rule states that for Sex=female, recourse cost is higher at 45.2. The next rule states that for age \geq 30, recourse cost is even higher at 47.6, and so on. Overall, the explanation table indicates that young individuals as well as females, especially Hispanic females, incur a higher than average cost to overturn a loan denial. On the other hand, the recourse cost for European individuals is lower than average. This gives a summary of the implicit bias of the model, drawing attention to subgroups with surprising or unusual recourse statistics.

3.2 Data Unlearning Workflow

As introduced in Section 2.2, data-based recourse captures how removing specific training points can alter model predictions. We choose datamodels as our data attribution method, a machine unlearning proxy that allows us to estimate data counterfactuals, due to their strong empirical alignment with full retraining outcomes, even on out-of-distribution subsets. Although any model-agnostic method can be used in our workflow, datamodels strike a practical balance between attribution quality and computational efficiency. In this section, we define two formalizations of data-based recourse metrics and then describe how to extract patterns from their distributions using explanation tables.

Our goal is to identify the minimal subset of TD that must be removed to flip the model’s prediction on a given sample x . This builds upon the concept of data support introduced by Ilyas et al. [24], which measures the brittleness of model predictions under data removal. In their formulation, data support refers to the smallest subset $R \subset TD$

such that models trained on $TD \setminus R$ misclassify x on average. In contrast, we refine this definition to focus not just on misclassification, but on the smallest subset that actively flips the model’s decision for x —which aligns more directly with our notion of data-based recourse.

To achieve this, we adapt data support estimation algorithm [24] to Algorithm 1, which identifies training subsets whose removal leads to a change in the predicted label. The number of such samples—i.e., the cardinality of the minimal subset required for a flip—is what we refer to as recourse support. Motivated by the reasons discussed in Section 2.2, we choose the margin as the model output function M_A . First, we initialize a buffer B to store margin estimates (line 2) and compute an initial margin m of a chosen sample x (line 3), which corresponds to an output of the model trained on full dataset. Then, for a range of candidate subset sizes $k \in K$ (line 4), the algorithm identifies a subset R_k of size k whose removal is most likely to change the prediction. We determine a subset of k training examples based on the sign of the margin:

- If $m > 0$ (correct prediction), we remove training samples corresponding to the *highest positive datamodel weights* θ to weaken support for the correct class.
- If $m < 0$ (incorrect prediction), we remove samples corresponding to the *most negative weights* θ to reduce opposition to the correct class.

This adaptation provides a closed-form solution (avoids exhaustive subset search by using a selection strategy based on datamodel weights) to lines 6 and 8 in Algorithm 1, ensuring an efficient estimation of the minimal training subset necessary for achieving recourse.

Algorithm 1 Data Attribution-Guided Estimation of Recourse Support for Prediction Flip

```
1: procedure RECURSE_SUPPORT( $x, TD, g_\theta, K$ )
2:    $B \leftarrow []$ 
3:   Compute initial margin  $m = M_A(x, TD)$ 
4:   for  $k \in K$  do
5:     if  $m > 0$  then ▷ Initially correct prediction
6:        $R_k \leftarrow \arg \min_{|R|=k} g_\theta(TD \setminus R)$ 
7:     else ▷ Initially incorrect prediction
8:        $R_k \leftarrow \arg \max_{|R|=k} g_\theta(TD \setminus R)$ 
9:     end if
10:    Estimate  $\mathbb{E}[M_A(x; TD \setminus R_k)]$ 
11:    Append  $(k, \mathbb{E}[M_A(x; TD \setminus R_k)])$  to  $B$ 
12:  end for
13:  Compute piecewise-linear interpolation  $h(\cdot)$  from  $B$ 
14:   $\hat{k} \leftarrow k$  for which  $h(k) = 0$ 
15:  return TOP-K( $\theta, \hat{k} \times 1.1$ ) ▷ Conservative estimate
16: end procedure
```

In line 10, we estimate $\mathbb{E}[M_A(x; TD \setminus R_k)]$ by retraining the model, and the result is stored (line 11). We could also use a simpler heuristic with no retraining, estimating recourse support by finding the smallest k such that the sum of the top- k datamodel weights exceeds the average margin of x , without a need to retrain. While computationally efficient, this approach does not guarantee minimality in identifying the subset that flips the prediction. The number of such retraining steps performed corresponds to the size of the set K ; for example, if $K = \{1, 5, 10, 100, \dots\}$, the model will be retrained on datasets with the top-1, top-5, top-10, and so on training points (as ranked by θ) removed, respectively.

After iterating through all values of k , a piecewise-linear interpolation is fitted over the buffer to approximate when the model’s margin crosses zero (lines 13-14), meaning the prediction on the sample x was flipped. Finally, the algorithm returns a conservative estimate of the top training samples to unlearn, scaled slightly beyond the estimated flip threshold (line 15).

To analyze aggregated data counterfactuals using explanation tables, we define two methods for computing the target value, assuming users have access to both the training dataset (TD) and the labeled diagnostics dataset (DD). In this context, we define the grouping set as the input dataset over which the explanation table is constructed—i.e., the

set of data points to be grouped into rules. Either dataset can serve as the input for an explanation table, and we introduce two target definitions for data-centric REACT:

1. *DD as the grouping set.* Each test sample $x_i \in DD$ is associated with a recourse support value—a target representing the minimal number of training samples that must be removed to alter its prediction, estimated via Algorithm 1.

A higher recourse support of a subgroup suggests that its features are strongly integrated into the model’s decision-making, making predictions more resistant to changes in training data. This can reflect meaningful generalization. However, if a protected subgroup exhibits high recourse support, it may indicate a need to investigate potential bias in the dataset. Conversely, low recourse support may suggest over-reliance on a few specific training points, which could signal spurious correlations, data poisoning or train-test leakage - a lack of robustness in the model’s learned patterns.

2. *TD as the grouping set.* The target—unlearning count—reflects how frequently a given training sample $z_j \in TD$ is selected for removal when Algorithm 1 is applied across all samples in DD . To compute this, we maintain a dictionary with training IDs j as keys, incrementing the count whenever z_j is included in the estimated minimal subset required for unlearning. A higher average frequency indicates that the subset plays a critical role in shaping model inference.

3.2.1 Workflow Example

Figure 3.3 illustrates a worked example of our data-based recourse pipeline, applied to a synthetic dataset. In this scenario, we focus on the first target definition—the number of training examples that must be removed to flip a model’s prediction. For each test individual predicted as not approved, we use Algorithm 1, yielding a scalar recourse support. For example, for the test instance with id=12, the algorithm estimates that unlearning just 5 training samples would be sufficient to flip the prediction.

We choose the first three protected features as an input to Step ②. Following the same rule mining process described earlier, the aggregation shows which patterns are more or less prone to brittleness: for example, pattern “*male, *, European*” has a higher-than-average value compared to the overall population indicating that individuals in this group require more training examples to be removed in order to change their outcome.

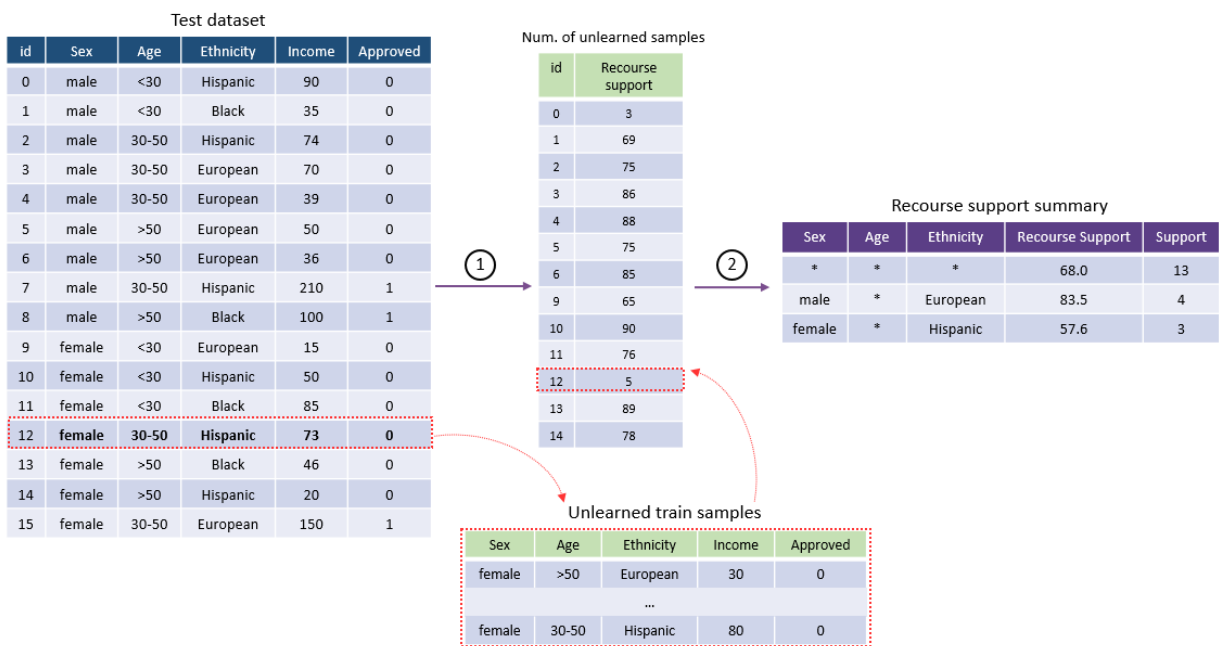


Figure 3.3: The workflow of data-based REACT: summarizing recourse support metric.

Chapter 4

Experiments

This chapter presents experiments conducted with the REACT tool on two real-world datasets. The tool is intended for users such as model engineers and AI auditors.

4.1 Implementation Details

We implemented most of REACT in Python, with an interactive web interface built using Streamlit (see Figure 4.1 for the front end). The explanation table generation component is written in C++ for improved performance. We extended it to support a numerical target attribute by adapting the original algorithm and optimizing the iterative process for scalability on datasets with a large number of records, following techniques described in SIRUM [14].

4.2 Datasets and Model Training

The first dataset is the the Adult Income - a census dataset with the binary label indicating whether an individual earns more than \$50,000 per year. Among 10 attributes, sex, race, and native country are protected features; some studies also classify age, marital and relationship status as protected. The dataset contains 45,222 instances in total, split into 31,655 for training (70%) and 13,567 for testing (30%). The split was stratified by the ground truth label.

The second dataset is the Toronto Police Strip Search, containing demographic attributes of arrested individuals, features describing the circumstances of their arrest, and a label indicating whether they were strip-searched. It was released as part of the police service’s efforts to identify and reduce potential systemic racism in policing practices [43]. In our analysis, we consider race, sex, and age group as protected attributes. The dataset consists of 65,078 instances, with 52,062 used for training (80%) and 13,016 for testing (20%), also using a stratified split by the outcome label.

For both datasets, we trained gradient-boosted decision tree classifiers using the XGBoost library. Because the datasets are imbalanced, we employed a pipeline that first applies SMOTE [8] to increase the minority class by creating synthetic samples, and then applies a random undersampling technique to reduce the majority class.

4.3 Feature-Based Recourse

In this subchapter, we describe three user investigation scenarios with feature modifications-based REACT.

4.3.1 Recourse Disparities in Salary Predictions

We start with the Adult Census Income dataset, which is widely used to evaluate algorithmic fairness. We begin by analyzing the first XGBoost classifier trained on this dataset without applying any techniques to balance accuracy across protected groups. Previous work on explicit bias analysis shows a poor classification performance (explicit bias) for females and individuals from non-White racial groups [1].

Suppose the user aims to assess the difficulty of flipping to a positive prediction (annual income over \$50,000), selecting education level as F_C , with 16 distinct values from Preschool to Doctorate, to perturb in Step ①, *recourse availability* as a target metric, and F_S as shown in Figure 4.1. REACT outputs the explanation table shown in Table 4.1. Note that not all the features listed in Step ② may be displayed in the tables—REACT runs a post-processing step removing features that do not participate in any rules. First, the average recourse availability across 9,966 samples is 36%. The summary then shows that 79% of married individuals achieve recourse, but only 33% of males without a family do (Rules 1, 6). Users can try other grouping attributes: running REACT with relationship and sex in F_S reveals an even lower recourse availability of 20% for females with a “not-in-family” status.

REACT: Recourse Analysis with Counterfactuals and Explanation Tables

Select metrics - the targets for explanation tables:

Recourse availability Recourse choice Recourse cost

Select a model

adult_classifier_... x

STEP 1: Select features to perturb in counterfactual generation Label of returned counterfactuals

education-num x 1

STEP 2: Select features for summarization

age x workclass x relationship x race x sex x native-country x

Number of returned subgroups for tables Min support parameter for tables

6 1%

Run diagnostics

Figure 4.1: The feature-based REACT parameter selection.

	age	relationship	race	sex	native-country	recourse_availability	support
0	*	*	*	*	*	36%	9,966
1	*	Married	*	*	*	79%	2,862
2	(41, 50]	*	*	*	*	72%	1,457
3	(34, 41]	*	White	*	US	57%	1,221
4	>50 years	*	*	*	*	54%	1,619
5	(26, 34]	*	White	Male	US	49%	1,174
6	*	Not-in-family	*	Male	*	33%	1,717

Table 4.1: Recourse availability explanation table when changing *education level* to generate counterfactuals.

We continue the recourse diagnostics by selecting two more sets F_C : [workclass, occupation] and hours-per-week, binned into four categories ranging from part-time to overtime. For workclass and occupation perturbations, we choose *recourse cost* and only the protected attributes to summarize by. In this very example, the cost of modifying these categorical features is either one or two per individual, corresponding to changing workclass and/or occupation to flip the prediction. The generated explanation table, shown in Table 4.2, reveals implicit bias patterns: subgroups of Mexican nationals and Black adults (Rules 1 and 3) with the average costs of 1.67 and 1.37, both exceeding the overall average of 1.31. For hours-per-week, including all the features in F_S , *recourse availability* shows that the model is more likely to predict higher income for males over 34 years if their number of working hours were to increase.

	marital-status	relationship	race	sex	native-country	recourse_cost	support
0	*	*	*	*	*	1.31	3,017
1	*	*	*	*	Mexico	1.67	113
2	*	Not-in-family	*	*	*	1.53	421
3	*	*	Black	*	*	1.37	176
4	Married-civ-spouse	*	White	Male	*	1.29	1,970

Table 4.2: Recourse cost, changing *workclass* and *occupation*.

4.3.2 Fairness-Driven Model Comparison

To illustrate how users can evaluate the impact of model updates on recourse fairness, we extend our analysis of the Adult dataset by introducing an additional XGBoost model with a modified training pipeline. This pipeline equalizes accuracy with respect to sex as one of the protected subgroups (explicit bias). Using the Fairlearn library [46], we remove any correlation of input features with sex as a preprocessing step. Next, we apply GridSearch proposed by Agarwal et al. [1] to select model parameters in a way that balances accuracy (measured by the F1 score) and fairness (evaluated using statistical parity).

In this use case, we select only a few features for summarization in Step ②, including the sex attribute—which is our primary focus—to avoid obtaining overly granular patterns that might reveal disparities related to other protected features.

Suppose the user wants to counterfactually modify hours-per-week to determine whether any implicit bias exists in the modified model. REACT shows that men’s recourse availability is 23.7% higher compared to the overall average (7.3% compared to 5.9%, see Table 4.3). This represents a reduction in the advantage for men, as compared to the initial explicitly biased model, where it was 50% higher (12% compared to 8%). The user observes the same trend when generating counterfactuals for two additional feature sets, [workclass, occupation] and [education].

	sex	race	recourse_availability	support
0	*	*	8%	9,966
1	*	Asian-Pac-Islander	12%	281
2	Male	Black	10%	525
3	Male	*	12%	5,980
4	*	White	9%	8,348
5	Male	White	12%	5,153

	sex	race	recourse_availability	support
0	*	*	5.9%	10,952
1	Male	*	7.3%	6,955
2	*	Asian-Pac-Islander	11.8%	297
3	Female	White	3.9%	3,206
4	Male	Black	8.5%	568
5	Male	Asian-Pac-Islander	15.5%	181
6	*	White	5.9%	9,259

Table 4.3: Comparison of recourse availability when changing *hours per week* between the first model (explicitly biased, left) and the second model (right).

In this use case, REACT shows that eliminating explicit bias in terms of demographic parity did not fully equalize recourse (implicit bias), revealing persistent gender disparity despite some improvement. This highlights the need for tools such as REACT to assess model unfairness, which can help with building new models that can balance all three factors: overall accuracy, fairness in accuracy, and fairness in recourse. One actionable insight enabled by REACT is selecting the best-performing model version by fairness in recourse.

4.3.3 Fairness in Pathways for Strip Decisions

We now turn to analysing the gradient-boosted tree trained on Police dataset. Assume the user decides, unlike in the previous two use cases, to analyze the likelihood of the model changing an arrestee’s classification to an *undesirable outcome* (labeled as 1, indicating a strip search, with the “label of returned counterfactuals” also set to 1) and selects *recourse choice* as the metric. For the counterfactual features in Step ①, they choose occurrence category (the type of incident leading to the arrest) or location of the arrest, aggregated at the Division level. In this setup, if a protected pattern appears with a higher average number of recourse choices, it reflects greater vulnerability—a wider range of feature changes under which the model would assign the undesirable label.

As shown in Table 4.4, REACT reveals that two protected subgroups—Black males (Rule 9) and young White arrestees aged 25 to 34 years (Rule 10)—would be predicted as strip-searched given more options where they could have been arrested, instead of their actual arrest location (averaging 3.02 and 4.03 recourse options, respectively, compared to 2.60 for the overall population). For the occurrence category, the model is more likely to flip its prediction to the undesirable outcome for males than for females, during the second arrest quarter in particular, when more alternative reasons for arrest are considered.

	arrest_quarter	race	sex	age	occurrence_category	combative_actions	assaulted_o	recourse_choice	support
0	*	*	*	*	*	*	*	2.60	10,573
1	*	*	*	*	Warrant	*	*	8.34	555
2	1	*	*	*	*	False	*	4.21	2,363
3	3	*	*	*	*	*	False	3.38	2,599
4	2	*	*	*	*	*	False	3.02	2,473
5	*	*	*	*	Robbery & Theft	*	*	4.50	1,438
6	*	*	Male	*	Assault & Other crime	*	*	4.88	710
7	*	*	*	*	FTA/FTC (Compliance	*	*	3.62	1,138
8	*	White	*	*	*	*	*	3.13	4,350
9	*	Black	Male	*	*	*	*	3.02	2,170
10	*	White	*	Aged 25 to 34 years	*	*	*	4.03	1,212
11	*	*	*	*	Drug Related	*	*	11.00	210

Table 4.4: Recourse choice explanation table when modifying *arrest location*.

4.4 Comparison with FACTS

The following outlines key differences between the FACTS and feature-based REACT approaches:

1. Main Parameters

In FACTS, the user defines an affected population - a subset based on a chosen sensitive attribute and a label corresponding to an undesirable outcome predicted by a model. Unlike FACTS, REACT does not require the sensitive attribute to be defined in advance, although one or more such attributes may be incorporated into F_C , F_S , or both, REACT additionally supports counterfactual analysis targeting flips to an undesirable prediction, which inverts the interpretation of aggregated metrics.

2. Counterfactual Generation Step

In FACTS, generating feature modifications (actions) by applying the frequent itemset mining on the unaffected population improves computational efficiency and increases the relevance of suggested actions. Therefore, constraining the analysis to a single protected attribute—though it can be considered a limitation—allows FACTS to compute counterfactuals significantly faster than DiCE used in REACT.

3. Summarization Step

FACTS ranks subpopulations by unfairness scores, computed as disparities in a chosen metric across subgroups, while REACT employs rule mining, detailed in Section 2.3.1, to identify patterns of recourse that diverge most significantly from the population average.

FACTS also allows users to define *domain-specific weights* for a feature set, which influence recourse cost-based formulas. However, this does not resolve the issue of imbalanced cost across different feature types—a limitation shared by REACT. Furthermore, although the authors of FACTS propose a cost function similar to the Gower distance used in our framework, their formulation is even more imbalanced: ordinal feature changes are not scaled proportionally to their range, treating all level transitions as equal.

Overall, REACT is designed to offer a concise summary of recourse rules for a fixed set of allowed feature modifications, making it well-suited for model engineers seeking actionable insights. In contrast, FACTS provides a broader diagnostic audit over the entire feature space and action set for a fixed protected attribute, making it more appropriate for detailed fairness investigations by model auditors. The frameworks are complementary—each summarizes group-level disparities through distinct, structured, feature-based patterns.

First, consider an example of a recourse report produced by FACTS on the (explicitly biased) Adult Income classifier we analyzed in previous section, and juxtapose it with results from REACT. In our comparison experiment, we focus on sex as a sensitive attribute. Figure 4.2 shows a subpopulation with the highest disparity in Equal Effectiveness—a metric defined by the FACTS authors to capture whether the same proportion of individuals across protected subgroups can achieve recourse—closely aligning with our notion of aggregated recourse availability. Notably, this subpopulation comprises White individuals from the ‘Private’ workclass whose native country is the United States - FACTS reports bias against the female subgroup within this population. A similar implicit bias pattern can be derived using REACT, in case the user sets F_C :[*education-num*, *hours-per-week*], as shown in Table 4.5, specifically in Rules 5 and 10.

```

If age = (26, 34], education-num = Some-college, hours-per-week = (39, 40], native-country = US, race = White, workclass = Private:
  Protected Subgroup 'Male', 1.74% covered out of 5980
    Make education-num = Bachelors with effectiveness 39.42% and counterfactual cost = 3.
    Make education-num = Bachelors, hours-per-week = (40, 55] with effectiveness 55.77% and counterfactual cost = 4.
    Make age = (34, 41] with effectiveness 55.77% and counterfactual cost = 10.
    Make age = (34, 41], hours-per-week = (40, 55] with effectiveness 55.77% and counterfactual cost = 11.
    Make age = (34, 41], education-num = Bachelors with effectiveness 55.77% and counterfactual cost = 13.
    Make age = (34, 41], education-num = Bachelors, hours-per-week = (40, 55] with effectiveness 60.58% and counterfactual cost = 14.
    Make age = (41, 50] with effectiveness 60.58% and counterfactual cost = 20.
    Make age = (41, 50], hours-per-week = (40, 55] with effectiveness 61.54% and counterfactual cost = 21.
    Make age = (41, 50], education-num = Bachelors with effectiveness 61.54% and counterfactual cost = 23.
    Make age = (41, 50], education-num = Bachelors, hours-per-week = (40, 55] with effectiveness 64.42% and counterfactual cost = 24.
    Make age = (41, 50], education-num = Masters, hours-per-week = (40, 55] with effectiveness 90.38% and counterfactual cost = 25.
    Make age = >50 years with effectiveness 90.38% and counterfactual cost = 30.
    Make age = >50 years, education-num = Bachelors with effectiveness 90.38% and counterfactual cost = 33.
    Make age = >50 years, education-num = Bachelors, hours-per-week = (40, 55] with effectiveness 90.38% and counterfactual cost = 34.
    Aggregate cost of the above recourses = 0.90
  Protected Subgroup 'Female', 1.58% covered out of 3986
    Make education-num = Bachelors with effectiveness 6.35% and counterfactual cost = 3.
    Make education-num = Bachelors, hours-per-week = (40, 55] with effectiveness 9.52% and counterfactual cost = 4.
    Make age = (34, 41] with effectiveness 9.52% and counterfactual cost = 10.
    Make age = (34, 41], hours-per-week = (40, 55] with effectiveness 9.52% and counterfactual cost = 11.
    Make age = (34, 41], education-num = Bachelors with effectiveness 9.52% and counterfactual cost = 13.
    Make age = (34, 41], education-num = Bachelors, hours-per-week = (40, 55] with effectiveness 11.11% and counterfactual cost = 14.
    Make age = (41, 50] with effectiveness 11.11% and counterfactual cost = 20.
  ...
    Make age = >50 years, education-num = Bachelors, hours-per-week = (40, 55] with effectiveness 19.05% and counterfactual cost = 34.
    Aggregate cost of the above recourses = 0.19
  Bias against Female due to Equal Effectiveness. Unfairness score = 0.713.

```

Figure 4.2: Subpopulation identified by FACTS with highest unfairness in aggregate effectiveness between female and male subgroups (equivalent to recourse availability in REACT).

	age	workclass	occupation	race	sex	native-country	recourse_availability	support
0	*	*	*	*	*	*	57%	9,966
1	(41, 50]	*	*	*	*	*	97%	1,457
2	(34, 41]	*	*	*	*	*	89%	1,608
3	>50 years	*	*	*	*	*	88%	1,619
4	(26, 34]	*	*	*	Male	*	54%	1,547
5	*	*	*	White	Male	US	65%	4,704
6	*	*	Exec-managerial	*	*	*	80%	791
7	(26, 34]	*	*	White	*	US	49%	1,744
8	*	*	Craft-repair	*	*	*	76%	1,341
9	*	*	Machine-op-inspct	*	Male	*	72%	632
10	(26, 34]	Private	*	*	Female	*	31%	661

Table 4.5: Informative subgroups on recourse availability obtained by changing *education level* and *hours-per-week*

For recourse cost calculation in FACTS, we assigned the following weights: {"race": 10, "sex": 10, "age": 10, "native-country": 10, "marital-status": 1, "relationship": 1, "occupation": 1, "workclass": 1, "hours-per-week": 1, "education-num": 1}. In REACT, all features are equally weighted, which is acceptable as the first four attributes listed above are not perturbed (excluded from the set of features permitted for constructing F_C). As seen on the Figure 4.2, for the surfaced subpopulation, FACTS provides information about each recourse option along with its cost.

We now run FACTS using the Equal Choice for Recourse metric, which adopts a macro-level interpretation of fairness, and set minimal effectiveness threshold for a recourse option to $\phi = 20\%$. Figure 4.3 presents one of the top-15 subpopulations identified using this criterion, showing that female subgroup has no effective recourse options and achieves a high unfairness score. While most subpopulations primarily involved counterfactual changes to age, several—such as the one shown—featured changes to [*education-num*, *hours-per-week*, *workclass*]. Notably, the corresponding subpopulations often shared overlapping features,

which limited the readability of the resulting summary.

```

If hours-per-week = (39, 40], native-country = US, occupation = Machine-op-inspct, workclass = Private:
  Protected Subgroup 'Male', 6.10% covered out of 5980
    Make hours-per-week = (40, 55], occupation = Exec-managerial with effectiveness 43.29%.
    Make occupation = Craft-repair with effectiveness 20.82%.
    Make occupation = Exec-managerial with effectiveness 31.51%.
    Make hours-per-week = (40, 55], occupation = Prof-specialty with effectiveness 40.55%.
    Make occupation = Prof-specialty with effectiveness 30.68%.
    Make hours-per-week = (40, 55], occupation = Sales with effectiveness 30.96%.
    Make hours-per-week = (40, 55], occupation = Craft-repair with effectiveness 29.59%.
    Make occupation = Sales with effectiveness 28.49%.
    Make hours-per-week = >55 hrs, occupation = Exec-managerial with effectiveness 43.84%.
    Make occupation = Tech-support with effectiveness 28.49%.
    Make hours-per-week = >55 hrs, occupation = Prof-specialty with effectiveness 29.32%.
    Make occupation = Prof-specialty, workclass = Local-gov with effectiveness 28.22%.
    Make hours-per-week = >55 hrs, occupation = Sales with effectiveness 30.68%.
    Make hours-per-week = (40, 55], occupation = Exec-managerial, workclass = Self-emp-inc with effectiveness 51.23%.
    Make occupation = Protective-serv, workclass = Local-gov with effectiveness 31.23%.
    Make hours-per-week = (40, 55], occupation = Prof-specialty, workclass = Local-gov with effectiveness 28.77%.
    Make hours-per-week = (40, 55], occupation = Prof-specialty, workclass = Self-emp-not-inc with effectiveness 40.82%.
    Make hours-per-week = (40, 55], occupation = Sales, workclass = Self-emp-not-inc with effectiveness 30.68%.
    Make occupation = Adm-clerical, workclass = Federal-gov with effectiveness 30.41%.
    Make occupation = Exec-managerial, workclass = Self-emp-inc with effectiveness 41.92%.
    Make hours-per-week = >55 hrs, occupation = Exec-managerial, workclass = Self-emp-inc with effectiveness 49.59%.
    Make hours-per-week = (40, 55], occupation = Adm-clerical with effectiveness 22.47%.
  Aggregate cost of the above recourses = -22.00
  Protected Subgroup 'Female', 3.36% covered out of 3986
    No recourses for this subgroup!
  Aggregate cost of the above recourses = 0.00
  Bias against Female due to Equal Choice for Recourse(Macro) (threshold = 0.2). Unfairness score = 22.0.

```

Figure 4.3: Subpopulation identified by FACTS with high unfairness in macro-level choice for recourse

When we set these features as F_C in REACT and apply the recourse choice metric, racial and ethnic bias is surfaced through separate rules in the explanation table (Table 4.6). In contrast to FACTS, REACT employs a micro-level interpretation, computing recourse options at the individual level prior to aggregation. The macro perspective provided by FACTS is particularly useful for filtering out actions with low subgroup-level effectiveness, thereby strengthening the fairness assessment by excluding actions unlikely to generalize within subgroups.

	marital-status	relationship	race	sex	native-country	recourse_choice	support
0	*	*	*	*	*	4.00	9,966
1	*	Married	*	*	*	12.28	2,862
2	*	*	White	Male	*	6.05	5,153
3	*	Not-in-family	*	Male	*	1.23	1,717
4	Married-civ-spouse	*	White	*	US	12.46	2,313
5	*	Unmarried	*	Male	*	1.98	333

Table 4.6: Informative subgroups on recourse choice obtained by changing *education level*, *hours-per-week* and *workclass*

4.5 Data-Based Recourse

For the Adult Income explicitly biased model, the histogram on Figure 4.4 shows a distribution of minimal number of the minimum number of training samples needed to unlearn to achieve recourse to a higher income. For around 11% of test samples, it is enough to unlearn less than 100 train samples. For individuals for whom recourse can be achieved without unlearning the entire TD , median support value is 24,723 samples (the train set size is 31,655 samples). The first table (Table 4.7) shows patterns summarizing this distribution of support. On average, 23 thousand data points have to be unlearned to change a model’s decision boundary in favor of a desirable outcome - that’s around 2/3 of the training data. The patterns show that females require more data to unlearn on average. Among males, however, the youngest adults form a subgroup with significantly less brittle predictions. As well as feature-based experiments, data-centric recourse aggregations revealed potential gender bias.

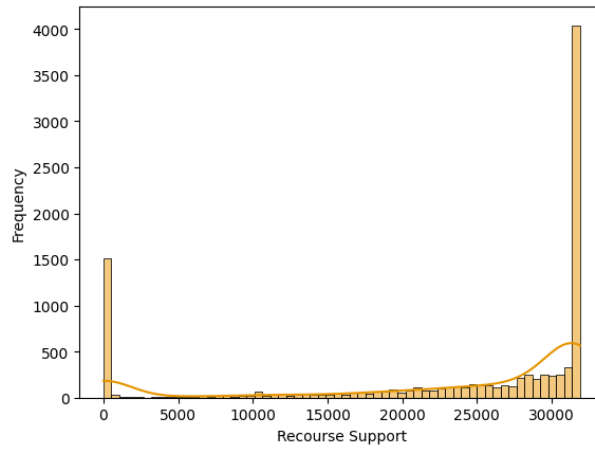


Figure 4.4: Histogram for the target distribution of data recourse support for adults labeled as low-income by explicitly biased model

	age	workclass	education-num	marital-status	occupation	relationship	race	sex	hours-per-week	native-country	recourse_support	support
0	*	*	*	*	*	*	*	*	*	*	23311	9966
1	*	*	*	Never-married	*	*	*	*	*	*	28816	4370
2	*	*	*	*	*	*	*	Female	*	*	27201	3986
3	<26 years	*	*	*	*	*	*	Male	*	*	29754	1663
4	*	*	*	Divorced	*	*	*	*	*	*	25714	1752
5	*	Private	*	*	*	*	*	*	(39, 40]	*	24481	3926
6	*	*	*	*	Other-service	*	*	*	*	*	28652	1444
7	*	*	*	*	*	*	*	*	<25 hrs	*	28442	1389
8	*	*	*	*	*	*	*	*	(25, 39]	*	26891	1416
9	*	*	HS-grad	*	*	Not-in-family	*	*	*	*	28635	1024

Table 4.7: Summary of data recourse support for low-income adults, first model (explicitly biased).

The second Adult Income model has similar patterns (Table 4.8), however, Rule 6 indicates a more fine-grained pattern that highlights female adults specifically from Private workclass. Again, this suggests that traditional bias elimination techniques may fall short in fully addressing underlying biases.

	age	workclass	education-num	marital-status	occupation	relationship	race	sex	hours-per-week	native-country	recourse_support	support
0	*	*	*	*	*	*	*	*	*	*	22549	10952
1	*	*	*	Never-married	*	*	*	*	*	*	29216	4397
2	*	*	*	Divorced	*	*	*	*	*	*	25993	1774
3	<26 years	*	*	*	*	*	*	*	*	*	30562	2943
4	*	*	*	*	Other-service	*	*	*	*	*	29244	1442
5	(26, 34]	*	*	*	*	*	*	*	(39, 40]	*	25206	1315
6	*	Private	*	*	*	*	*	Female	*	*	27401	3189
7	*	*	*	Separated	*	*	*	*	*	*	27392	402
8	*	*	*	*	*	*	*	*	(25, 39]	*	27601	1425
9	*	*	HS-grad	*	*	Not-in-family	*	*	*	*	29418	1026

Table 4.8: Summary of data recourse support for low-income adults, second model.

Interestingly, unlike feature-centric *REACT*, the data counterfactuals perspective with *DD* as the grouping set did not surface any racial bias in either of the two models. This suggests that this perspective alone may not be sufficient for diagnosing implicit bias. To address this, we turn to the second target definition—using *TD* as the underlying dataset—to examine which training samples are most frequently unlearned. Notably, race-related patterns do appear in the explanation table in Table 4.9, where the target—unlearning count—is assigned to each training sample. However, the average unlearning frequency across surfaced subgroups does not differ significantly from the overall average, complicating its interpretation as a signal of systemic bias.

	age	workclass	education-num	marital-status	occupation	relationship	race	sex	hours-per-week	native-country	unlearning_count	support
0	*	*	*	*	*	*	*	*	*	*	4171	31655
1	*	*	*	*	Sales	*	*	*	*	*	4245	3776
2	*	Private	HS-grad	*	*	Married	*	Male	*	*	4252	3023
3	*	*	*	*	*	*	*	Female	*	*	4186	10305
4	(34, 41]	Private	*	*	*	*	*	*	*	*	4224	4260
5	(34, 41]	Private	*	*	*	*	*	Male	*	*	4245	3003
6	*	*	*	Divorced	*	*	White	*	*	US	4214	3677
7	*	*	*	Divorced	*	*	*	*	*	*	4204	4456
8	*	*	*	Divorced	*	*	White	*	*	*	4211	3833
9	(41, 50]	*	*	*	*	*	*	Female	*	*	4257	1779
10	*	Private	Bachelors	*	*	*	*	*	*	*	4218	3604
11	*	Private	Bachelors	*	*	*	White	*	*	*	4220	3217
12	*	*	*	*	*	Married	*	*	>55 hrs	*	4265	1760
13	*	*	*	*	*	*	*	*	(40, 55]	*	4198	6910
14	*	*	*	*	*	*	White	*	(40, 55]	*	4198	6380
15	*	*	*	*	*	*	*	*	(40, 55]	US	4196	6477

Table 4.9: Summarizing a target indicating how frequently a given adult income training sample is being “machine-unlearned”.

For the Toronto Police model, the second target definition results in aggregation shown in Table 4.10. All of the top 15 subgroups include “sex = Male”, and three of them also contain “race = White” in combination with other patterns, indicating a reliance on these protected attributes when unlearning data to flip predictions on the diagnostics dataset.

Table 4.11 and Table 4.12 show the aggregation of recourse support for flipping predictions for arrestees who were labeled strip-searched and those who were not, respectively. We provide the corresponding histograms in Figure 4.5. We observe that the first subset—comprising individuals predicted by the model as strip searched, the undesirable outcome—is associated with a significantly greater brittleness than the other subset, lower recourse support, and two protected attributes (race and sex) included in the informative patterns. This can be connected to the higher rate of misclassifications observed within this subset, and highlights importance to run explanation tables separately on subsets labeled with $\hat{l} = 1$ and $\hat{l} = 0$.

	arrest_quarter	race	sex	age	location	occurrence_category	concealed_items	combative_actions	resisted_actions	mental_inst	assaulted_o	cooperative	unlearning_count	support
0	*	*	*	*	*	*	*	*	*	*	*	*	5,075	52,062
1	*	*	Male	Aged 25 to 34 years	*	*	*	*	False	*	False	*	5,106	12,655
2	*	*	Male	Aged 25 to 34 years	*	*	False	*	False	*	False	*	5,106	12,600
3	3	*	Male	*	*	*	*	False	False	False	False	*	5,114	9,859
4	*	*	Male	Aged 25 to 34 years	*	*	False	*	False	False	False	*	5,107	12,183
5	*	*	Male	Aged 25 to 34 years	*	*	*	*	False	False	False	*	5,106	12,226
6	3	*	Male	*	*	*	*	False	False	False	*	*	5,113	9,875
7	*	White	Male	*	*	*	False	False	False	False	*	*	5,099	15,530
8	*	*	Male	Aged 25 to 34 years	*	*	*	*	False	*	*	*	5,104	12,726
9	*	White	Male	*	*	*	False	False	False	False	False	*	5,099	15,504
10	3	*	Male	*	*	*	False	False	False	False	False	*	5,113	9,826
11	*	White	Male	*	*	*	False	False	False	*	*	*	5,098	15,879

Table 4.10: Summarizing a target indicating how frequently a given police data training sample is being “machine-unlearned”.

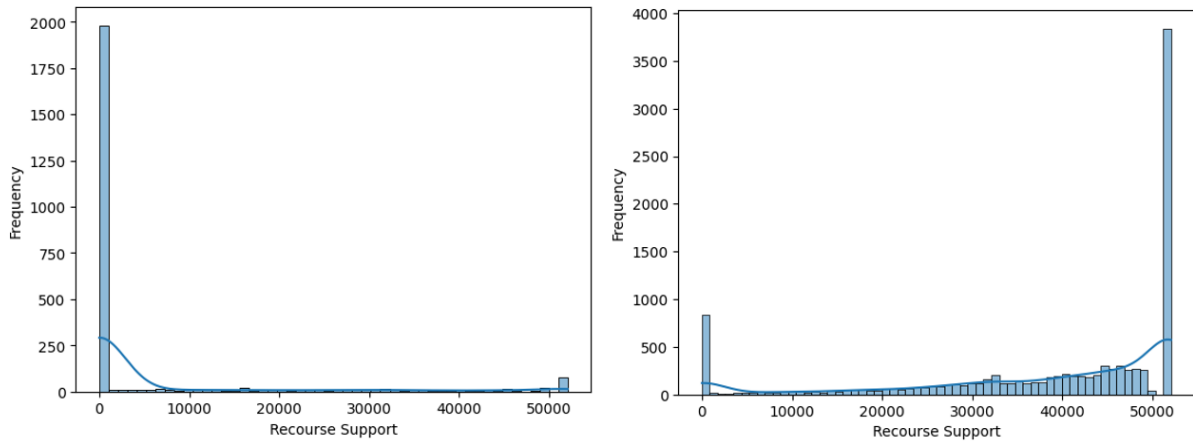


Figure 4.5: Histograms for the target distribution of recourse support for arrested citizens labeled as strip searched (on the left) and as not strip searched (on the right)

	arrest_quarter	race	sex	age	location	occurrence_category	concealed_items	combative_actions	resisted_actions	mental_inst	assaulted_o	cooperative	recourse_support	support
0	*	*	*	*	*	*	*	*	*	*	*	*	5,594	2,443
1	*	*	*	*	51	*	*	*	*	*	*	*	11,076	483
2	*	*	Male	*	*	Weapons & Homicide	*	*	*	*	*	*	19,060	120
3	*	*	*	*	*	Drug Related	*	*	*	*	*	*	9,082	384
4	*	*	*	*	*	Assault & Other crimes	False	*	*	*	*	False	9,213	287
5	*	*	*	*	*	Warrant	False	*	*	*	*	*	7,075	300
6	*	*	*	*	*	*	*	True	*	*	*	*	11,079	204
7	*	White	*	*	*	*	False	False	False	*	*	True	5,973	465
8	*	Black	Male	*	*	*	*	*	*	*	*	*	8,068	694
9	*	*	*	*	*	Break & Enter	False	*	False	*	*	*	5,409	185
10	*	*	*	*	*	*	*	*	*	True	*	*	10,329	194
11	3	*	Male	*	*	*	*	*	False	*	*	*	7,744	550

Table 4.11: Summary of data recourse support for arrestees predicted to be strip searched. Patterns reveal subgroups requiring significantly more or fewer samples to be unlearned to achieve a positive outcome, such as Black males.

	arrest_quarter	race	sex	age	location	occurrence_category	concealed_items	combative_actions	resisted_actions	mental_inst	assaulted_o	cooperative	recourse_support	support
0	*	*	*	*	*	*	*	*	*	*	*	*	38,844	10,573
1	*	*	*	*	XX	*	False	False	*	False	*	*	43,216	5,412
2	4	*	*	*	*	*	False	*	*	False	False	*	45,271	2,945
3	*	*	*	*	*	Assault	False	False	False	*	*	*	49,210	1,372
4	*	*	*	*	*	Police Category (Admit	*	*	*	*	*	*	49,579	832
5	*	*	*	*	XX	*	False	*	*	*	False	True	45,292	2,539
6	*	*	*	*	*	Impaired	*	*	*	*	*	*	49,776	272
7	*	*	*	*	*	Other Statute & Other	*	*	*	*	*	*	44,402	445
8	*	*	*	*	XX	Robbery & Theft	False	False	False	False	*	*	43,668	1,033
9	*	*	*	*	*	Mischief	*	*	*	*	*	*	48,416	232

Table 4.12: Summary of data recourse support for arrestees predicted to be not strip searched. Top patterns do not contain protected attributes.

Chapter 5

Conclusion

REACT introduces a novel combination of feature-based and data-based counterfactual explanations with subgroup-level aggregation to highlight implicit bias patterns in model behavior. For feature-based recourse, we used a model-agnostic variant of DiCE with a custom post-processing step to remove redundant counterfactuals. For data-based recourse, we employed a data attribution-guided unlearning algorithm which estimates the minimal subset of training points whose removal flips a prediction—avoiding the cost of repeated retraining. This setup allowed us to compute and summarize several recourse metrics, including cost, availability, choice, support, and unlearning counts, surfacing subgroups with disproportionately high barriers to recourse. Unlike other frameworks, REACT does not require specifying protected features in advance; instead, it identifies disparities across any attributes selected for summarization, allowing practitioners to discover which subpopulations are most affected without presupposing where bias may lie.

In experiments on the Adult Income and Toronto Police datasets, our approach surfaced protected groups—such as ethnic and racial minorities—that faced fewer recourse options or required more training examples to be unlearned. At the same time, it revealed advantaged subgroups—such as gender majorities—that experienced lower recourse burden, indicating implicit bias even in the model where explicit fairness metrics appeared balanced. The obtained insights are actionable: they can inform data collection, selective unlearning of problematic training examples, or adjustments to model decision thresholds to reduce disparities for the most affected subgroups.

5.1 Future Work

Future research could focus on improving the efficiency and scalability of REACT. Its modular architecture allows users to easily substitute alternative counterfactual generators, enabling exploration of methods that are both faster and more reliable than the random-sampling-based DiCE—while maintaining model-agnostic applicability and addressing the limitation of potentially failing to find counterfactuals even when valid ones exist.

Another promising direction is to extend REACT for more effective comparison between model versions in terms of recourse fairness. Instead of relying on side-by-side explanation tables (as we did in Section 4.3.2), one could adopt an approach inspired by CAMO [47], which summarizes performance differences across subgroups between two or more models. While CAMO does not consider recourse disparities, a similar idea can be applied in REACT by defining a target metric that quantifies, for each individual, the difference in recourse between model versions. Aggregating these per-individual disparities using rule mining would yield a single explanation table that highlights subgroups where the models differ most in terms of implicit bias. This can support informed model upgrades in production by identifying whether newer versions reduce or exacerbate recourse disparities.

We also suggest extending REACT’s data-based recourse analysis by defining new target metrics that capture the trade-off between the number of training examples that must be unlearned to achieve a label flip and the resulting loss in model accuracy. When using the diagnostics dataset as the grouping set, this would allow us to surface subgroups whose recourse requires a disproportionately large reduction in overall model performance, providing a more nuanced view of model brittleness.

Adapting REACT to other *data modalities* such as images and text presents an important but nontrivial direction. As an example, in surveillance settings, image-based models may exhibit similar recourse biases to those we observed in the tabular Toronto Police dataset—where young Black males were disproportionately associated with recourse options leading to the undesirable outcome (Section 4.3.3). For example, we counterfactually modify the attire of a person in surveillance footage—such as adding or removing a hoodie. If the model is then more likely to flag individuals from certain protected subgroups as potential suspects, this would indicate that it exhibits implicit bias in how it interprets visual features. For text data, recourse disparities may arise when counterfactual removal of certain tokens—such as those in resumes or college admission essays—results in more favorable outcomes for individuals whose group membership is indicated by other tokens correlated with gender, ethnicity, or race, reflecting that the model weighs linguistic inputs differently across demographic groups.

The primary challenge lies in the feasibility of generating meaningful feature-based counterfactuals for unstructured data. Image counterfactuals present significant computational and interpretability challenges compared to tabular data. For instance, Chang et al. [7] highlighted the reality gap in counterfactual generation using generative models—while visually plausible, such counterfactuals may not correspond to feasible real-world modifications or lie on the true data manifold. For natural language processing tasks (NLP) such as text classification, measuring recourse distance may help reveal if the model has learned some concept artifacts - spurious correlations between input tokens and output that don't reflect true causal relationships between unstructured features and classes. Although no prior work has specifically addressed recourse in NLP, several efforts have been made to develop and automate the generation of meaningful text counterfactual explanations. For example, Fern et al. [15] proposed a method combining latent space optimization and beam search guided by Shapley values to generate sparse, meaningful edits that flip the model's prediction. Lemberger et al. [32] and Madaan et al. [34] proposed methods that also effectively intervene in the latent representation space. The former uses a linear projection to separate the sensitive attribute information from the rest of the representation. The representation space is decomposed into two components: x_{\perp} , the part of the representation orthogonal to the sensitive attribute, containing general information about the text, and the aligned component x_{\parallel} .

To apply REACT in these domains, one would also need to extract interpretable concepts from unstructured data—for instance, object categories from images or topic/concept tags from text—to serve as inputs to the informative rule mining algorithm that requires structured features.

References

- [1] Alekh Agarwal, Alina Beygelzimer, Miroslav Dudík, John Langford, and Hanna M. Wallach. A reductions approach to fair classification. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*, volume 80, pages 60–69. PMLR, 2018.
- [2] Andrew Bell, Joao Fonseca, Carlo Abrate, Francesco Bonchi, and Julia Stoyanovich. Fairness in algorithmic recourse through the lens of substantive equality of opportunity, 2024.
- [3] Andrew Bell, João Fonseca, and Julia Stoyanovich. The game of recourse: Simulating algorithmic recourse over time to improve its reliability and fairness. In *Companion of the 2024 International Conference on Management of Data, SIGMOD/PODS 2024*, page 464–467, 2024.
- [4] Lucas Bourtole, Varun Chandrasekaran, Christopher A. Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 141–159, 2021.
- [5] Lucius E. J. Bynum, Joshua R. Loftus, and Julia Stoyanovich. A new paradigm for counterfactual reasoning in fairness and recourse. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI 2024, Jeju, South Korea, August 3-9, 2024*, pages 7092–7100. ijcai.org, 2024.
- [6] Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. In *2015 IEEE Symposium on Security and Privacy*, pages 463–480, 2015.
- [7] Chun-Hao Chang, Elliot Creager, Anna Goldenberg, and David Duvenaud. Explaining image classifiers by counterfactual generation. In *7th International Conference*

on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019. OpenReview.net, 2019.

- [8] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.
- [9] Yeounoh Chung, Tim Kraska, Neoklis Polyzotis, Ki Hyun Tae, and Steven Euijong Whang. Automated data slicing for model validation: A big data - AI integration approach. *IEEE Trans. Knowl. Data Eng.*, 32(12):2284–2296, 2020.
- [10] Giandomenico Cornacchia, Vito Walter Anelli, Giovanni Maria Biancofiore, Fedelucio Narducci, Claudio Pomo, Azzurra Ragone, and Eugenio Di Sciascio. Auditing fairness under unawareness through counterfactual reasoning. *Inf. Process. Manag.*, 60(2):103224, 2023.
- [11] Armin Esmaelizadeh, Sunil Cotterill, Liam Hebert, Lukasz Golab, and Kazem Taghva. Infomod: information-theoretic machine learning model diagnostics. *Distributed Parallel Databases*, 43(1):6, 2025.
- [12] S. Eyuboglu, M. Varma, K. K. Saab, J.-B. Delbrouck, C. Lee-Messer, J. Dunnmon, J. Zou, and C. Ré. Domino: Discovering systematic errors with cross-modal embeddings. In *ICLR*, 2022.
- [13] Vitaly Feldman and Chiyuan Zhang. What neural networks memorize and why: Discovering the long tail via influence estimation. In Hugo Larochelle, Marc’Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.
- [14] Guoyao Feng, Lukasz Golab, and Divesh Srivastava. Scalable informative rule mining. In *33rd IEEE International Conference on Data Engineering, ICDE 2017, San Diego, CA, USA, April 19-22, 2017*, pages 437–448. IEEE Computer Society, 2017.
- [15] Xiaoli Fern and Quintin Pope. Text counterfactuals via latent optimization and Shapley-guided search. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5578–5593, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

- [16] Hidde Fokkema, Damien Garreau, and Tim van Erven. The risks of recourse in binary classification. In Sanjoy Dasgupta, Stephan Mandt, and Yingzhen Li, editors, *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pages 550–558. PMLR, 02–04 May 2024.
- [17] Christos Fragkathoulas, Vasiliki Papanikou, Danae Pla Karidi, and Evaggelia Pitoura. On explaining unfairness: An overview. In *40th International Conference on Data Engineering, ICDE 2024 - Workshops*, pages 226–236. IEEE, 2024.
- [18] Kareem El Gebaly, Guoyao Feng, Lukasz Golab, Flip Korn, and Divesh Srivastava. Explanation tables. *IEEE Data Eng. Bull.*, 41(3):43–51, 2018.
- [19] Amirata Ghorbani and James Zou. Data shapley: Equitable valuation of data for machine learning. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2242–2251. PMLR, 09–15 Jun 2019.
- [20] John C Gower. A general coefficient of similarity and some of its properties. *Biometrics*, pages 857–871, 1971.
- [21] Vivek Gupta, Pegah Nokhiz, Chitradeep Dutta Roy, and Suresh Venkatasubramanian. Equalizing recourse across groups, 2019.
- [22] Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. *Advances in neural information processing systems*, 29, 2016.
- [23] Peter Henderson, Xuechen Li, Dan Jurafsky, Tatsunori Hashimoto, Mark A. Lemley, and Percy Liang. Foundation models and fair use. *J. Mach. Learn. Res.*, 24:400:1–400:79, 2023.
- [24] Andrew Ilyas, Sung Min Park, Logan Engstrom, Guillaume Leclerc, and Aleksander Madry. Datamodels: Understanding predictions with data and data with predictions. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato, editors, *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 9525–9587. PMLR, 2022.
- [25] Ruoxi Jia, David Dao, Boxin Wang, Frances Ann Hubis, Nick Hynes, Nezihe Merve Gürel, Bo Li, Ce Zhang, Dawn Song, and Costas J. Spanos. Towards efficient data valuation based on the shapley value. In Kamalika Chaudhuri and Masashi Sugiyama,

- editors, *The 22nd International Conference on Artificial Intelligence and Statistics, AISTATS 2019, 16-18 April 2019, Naha, Okinawa, Japan*, volume 89 of *Proceedings of Machine Learning Research*, pages 1167–1176. PMLR, 2019.
- [26] Shalmali Joshi, Oluwasanmi Koyejo, Warut Vijitbenjaronk, Been Kim, and Joydeep Ghosh. Towards realistic individual recourse and actionable explanations in black-box decision making systems. *CoRR*, abs/1907.09615, 2019.
- [27] Amir-Hossein Karimi, 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.
- [28] Amir-Hossein Karimi, Bernhard Schölkopf, and Isabel Valera. Algorithmic recourse: from counterfactual explanations to interventions. In Madeleine Clare Elish, William Isaac, and Richard S. Zemel, editors, *FACCT ’21: 2021 ACM Conference on Fairness, Accountability, and Transparency, Virtual Event / Toronto, Canada, March 3-10, 2021*, pages 353–362. ACM, 2021.
- [29] Loukas Kavouras, Konstantinos Tsopelas, Giorgos Giannopoulos, Dimitris Sacharidis, Eleni Psaroudaki, Nikolaos Theologitis, Dimitrios Rontogiannis, Dimitris Fotakis, and Ioannis Z. Emiris. Fairness aware counterfactuals for subgroups. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023*, 2023.
- [30] Pang Wei Koh, Kai-Siang Ang, Hubert H. K. Teo, and Percy Liang. On the accuracy of influence functions for measuring group effects. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d’Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 5255–5265, 2019.
- [31] M. J. Kusner, J. Loftus, C. Russell, and R. Silva. Counterfactual fairness. In *NeurIPS*, volume 30, 2017.
- [32] Pirmin Lemberger and Antoine Saillenfest. Explaining text classifiers with counterfactual representations. In Ulle Endriss, Francisco S. Melo, Kerstin Bach, Alberto José Bugarín Diz, Jose Maria Alonso-Moral, Senén Barro, and Fredrik Heintz, editors, *ECAI 2024 - 27th European Conference on Artificial Intelligence, 19-24 October 2024, Santiago de Compostela, Spain - Including 13th Conference on Prestigious Applications of Intelligent Systems (PAIS 2024)*, volume 392 of *Frontiers in Artificial Intelligence and Applications*, pages 890–897. IOS Press, 2024.

- [33] Dan Ley, Saumitra Mishra, and Daniele Magazzeni. GLOBE-CE: A translation based approach for global counterfactual explanations. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 19315–19342. PMLR, 2023.
- [34] Nishtha Madaan, Inkit Padhi, Naveen Panwar, and Diptikalyan Saha. Generate your counterfactuals: Towards controlled counterfactual generation for text. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 13516–13524. AAAI Press, 2021.
- [35] Christoph Molnar. *Interpretable Machine Learning*. 3 edition, 2025.
- [36] Ramaravind Kommiya Mothilal, Amit Sharma, and Chenhao Tan. Explaining machine learning classifiers through diverse counterfactual explanations. In *FAT* ’20: Conference on Fairness, Accountability, and Transparency, 2020*, pages 607–617. ACM, 2020.
- [37] Sung Min Park, Kristian Georgiev, Andrew Ilyas, Guillaume Leclerc, and Aleksander Madry. TRAK: attributing model behavior at scale. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 27074–27113. PMLR, 2023.
- [38] Pouya Pezeshkpour, Sarthak Jain, Byron C. Wallace, and Sameer Singh. An empirical comparison of instance attribution methods for NLP. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tür, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 967–975. Association for Computational Linguistics, 2021.
- [39] Rafael Poyiadzi, Kacper Sokol, Raul Santos-Rodriguez, Tijl De Bie, and Peter Flach. Face: feasible and actionable counterfactual explanations. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 344–350, 2020.

- [40] Kaivalya Rawal and Himabindu Lakkaraju. Beyond individualized recourse: Interpretable and interactive summaries of actionable recourses. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 12187–12198. Curran Associates, Inc., 2020.
- [41] Shreya Shankar, Rolando Garcia, Joseph M Hellerstein, and Aditya G Parameswaran. ” we have no idea how models will behave in production until production”: How engineers operationalize machine learning. *Proceedings of the ACM on Human-Computer Interaction*, 8(CSCW1):1–34, 2024.
- [42] Tanmay Surve and Romila Pradhan. Explaining fairness violations using machine learning. In Alkis Simitsis, Bettina Kemme, Anna Queralt, Oscar Romero, and Petar Jovanovic, editors, *Proceedings 28th International Conference on Extending Database Technology, EDBT 2025, Barcelona, Spain, March 25-28, 2025*, pages 623–635. Open-Proceedings.org, 2025.
- [43] Toronto Police Service. Race based data: Open data documentation, November 2022. Accessed April 2025.
- [44] Berk Ustun, Alexander Spangher, and Yang Liu. Actionable recourse in linear classification. In *Proceedings of the Conference on Fairness, Accountability, and Transparency, FAT* ’19*. ACM, January 2019.
- [45] Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harv. JL & Tech.*, 31:841, 2017.
- [46] Hilde J. P. Weerts, Miroslav Dudík, Richard Edgar, Adrin Jalali, Roman Lutz, and Michael Madaio. Fairlearn: Assessing and improving fairness of AI systems. *J. Mach. Learn. Res.*, 24:257:1–257:8, 2023.
- [47] Andy Yu, Parke Godfrey, Lukasz Golab, Divesh Srivastava, and Jaroslaw Szlichta. CAMO: explaining consensus across models. In *40th IEEE International Conference on Data Engineering, ICDE 2024*, pages 5493–5496. IEEE, 2024.