REACT: REcourse Analysis with Counterfactuals and Explanation Tables

Anastasiia Avksientieva
University of Waterloo
Waterloo, Canada
aavksientieva@uwaterloo.ca

Parke Godfrey
York University
Toronto, Canada
godfrey@yorku.ca

Lukasz Golab
University of Waterloo
Waterloo, Canada
lgolab@uwaterloo.ca

Divesh Srivastava
AT&T Chief Data Office
New Jersey, USA
divesh@research.att.com

Jarek Szlichta
York University
Toronto, Canada
szlichta@yorku.ca

ABSTRACT

Machine learning models may suffer from explicit bias (poor performance on some test examples) and implicit bias (hard to modify some test examples to change the model's prediction from an undesirable to a desirable label). We demonstrate REACT: a system designed to summarize implicit bias patterns. Our solution combines counterfactual analysis (modifying model inputs to alter predictions) with data summarization (finding biased subgroups). In our demonstration, participants will use REACT to analyze and understand implicit biases of various models.

1 INTRODUCTION

Before deploying machine learning models in production, it is critical to assess their performance on unseen data. Recent approaches such as Model Slicing [3], InfoMoD [4] and CAMO [10] enable such model diagnostics, for example, by identifying under-performing subgroups. Such tools 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 such as medicine, it is critical to identify biases and take corrective actions such as collecting more training data.

Recent work has observed that, in addition to *explicit* biases in prediction fairness, models may suffer from *implicit* biases [2]. Implicit bias can be captured by *recourse distance*; this is the distance between an example and its *counterfactual*, which is a modified example whose features are perturbed to flip the model's decision from an undesirable to a desirable outcome. For example, a credit card approval model that has the same accuracy on men and women might still be unfair in terms of recourse distance or burden: say that men whose applications were denied would only need to add 10 percent on average to their savings to be approved, whereas women whose applications were denied would need to add 20 percent on average to their savings *and* reduce their existing debt by 25 percent.

Existing work on recourse analysis sorts subgroups of a test set by recourse distance [5] or other related metrics [2, 7]. 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.

To address this problem, we present REACT[1], a tool for *REcourse Analysis with Counterfactuals and Explanation Tables.* Given a test dataset, REACT computes *recourse paths* (*counterfactuals*) for each example, and summarizes the recourse statistics using our recent work on informative rule mining (*explanation tables*) [6, 10]. We make the following contributions:

- **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 3.3).
- **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 counterfactual explanations with explanation tables. This fusion enables to capture both individual-level feature perturbations and broader patterns within the data, improving the interpretability and actionability of the summaries. For instance, REACT can identify subgroups where achieving recourse is more or less likely compared to the dataset average or uncover subgroups with multiple viable 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-accepted).
- **Demonstrating REACT.** We describe the REACT user experiencewith several classifiers and test datasets ranging from police search to income prediction. Our analysis demonstrates that even equal or fair accuracy rates may still lead to disparities, such as unequal recourse distance, which can be effectively summarized by REACT.

To summarize, we introduce a novel approach to fairness diagnostics. Unlike tools such as InfoMoD [4] 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. Compared to FACTS [7], which proposes various recourse bias definitions and frameworks, REACT

[1] REACT is available at http://lg-research-2.uwaterloo.ca:8053

Series ISSN: 2367-2005

10.48786/edbt.2025.98

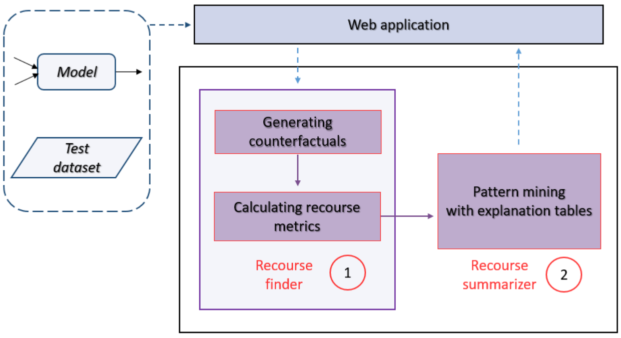

Web application
Model
Test dataset
Generating counterfactuals
Calculating recourse metrics
Recourse finder
1
Pattern mining with explanation tables
Recourse summarizer
2

Figure 1: The architecture of REACT.

instead focuses on a concise presentation of bias analytics, highlighting "surprising" subgroups whose recourse statistics are significantly different from the average.

2 SYSTEM DESCRIPTION

2.1 System Design

Figure 1 illustrates the design of REACT, implemented as an interactive web application using Python and Streamlit (see Figure 3 for the front end). The input to REACT consists of a model M and a test dataset T with features f and a binary label l. We divide f into three disjoint subsets, f_C, the features that can be perturbed to generate counterfactuals, 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. Additionally, 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 $l = 1$ be the desired counterfactual label in the remainder of this section.

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 T with $l = 0$, by perturbing the values of f_C in a way that changes M's prediction to $l = 1$. We use DiCE [8] in our implementation due to its efficiency, but the REACT back end is compatible with any counterfactual generator, as long as the produced counterfactuals are not redundant, i.e., that no counterfactual is subsumed by another. 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.

Next, for every example in T with $l = 0$ and based on the generated counterfactuals, 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 [5, 7]:

(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 $l = 1$).
(2) *Recourse Cost* is the distance between the original example and its *nearest* counterfactual, using range- or rank-normalized Manhattan distance for numeric and ordinal features, respectively, and binary distance (one if the feature has changed, zero otherwise) for categorical features.
(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 in T with $l = 0$ is labelled with its recourse availability and choice. Examples with availability one are additionally labelled with their recourse cost.

The output of Step ① becomes the input to Step ②, the recourse summarizer. Here, the user selects the metric of interest: availability, cost, or choice. For each selected metric, REACT produces a corresponding summary: a set of k rules, with k being a user-set parameter, each referring to a subgroup of T defined by the values of the features in f_S. For example, a rule over sex and age might be "sex = male and age = 18-to-45". In addition to f_S and k, the user selects a support threshold specifying the minimum number of examples that each rule must cover.

REACT uses *explanation tables* [6, 10], an information-theoretic rule mining method. It identifies the k most informative rules with respect to the distribution of some dependent variable, which in our case is either the recourse availability, recourse cost or recourse choice. We describe explanation tables in more detail below.

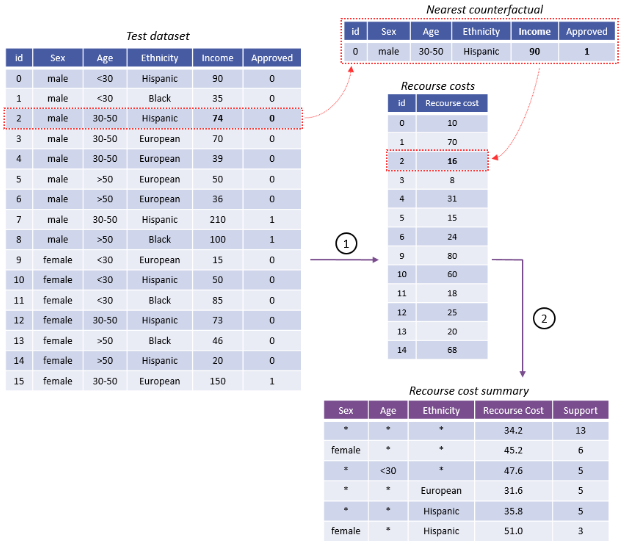
2.2 Workflow Overview

To illustrate the workflow, consider the test set for loan approval classification shown in Figure 2 on the left. The feature set f consists of Sex, Age, Ethnicity and Income. The Approved label is one if the loan was approved and zero otherwise. Let f_C = Income and f_S be the remaining features. That is, if we can modify Income to turn declined examples into approved ones via counterfactuals, what are the recourse patterns within subgroups of the test set identified by Sex, Age and Ethnicity?

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 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. In the middle-right corner of Figure 2, we show a table with 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 2, we show the explanation table for recourse cost, 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



Test dataset

id	Sex	Age	Ethnicity	Income	Approved
0	male	<30	Hispanic	90	0
1	male	<30	Black	35	0
2	male	30-50	Hispanic	**74**	**0**
3	male	30-50	European	70	0
4	male	30-50	European	39	0
5	male	>50	European	50	0
6	male	>50	European	36	0
7	male	30-50	Hispanic	210	1
8	male	>50	Black	100	1
9	female	<30	European	15	0
10	female	<30	Hispanic	50	0
11	female	<30	Black	85	0
12	female	30-50	Hispanic	73	0
13	female	>50	Black	46	0
14	female	>50	Hispanic	20	0
15	female	30-50	European	150	1

Nearest counterfactual

id	Sex	Age	Ethnicity	Income	Approved
0	male	30-50	Hispanic	**90**	**1**

Recourse costs

id	Recourse cost
0	10
1	70
2	**16**
3	8
4	31
5	15
6	24
9	80
10	60
11	18
12	25
13	20
14	68

Recourse cost summary

Sex	Age	Ethnicity	Recourse Cost	Support
*	*	*	34.2	13
female	*	*	45.2	6
*	<30	*	47.6	5
*	*	European	31.6	5
*	*	Hispanic	35.8	5
female	*	Hispanic	51.0	3

Figure 2: Workflow example.

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.

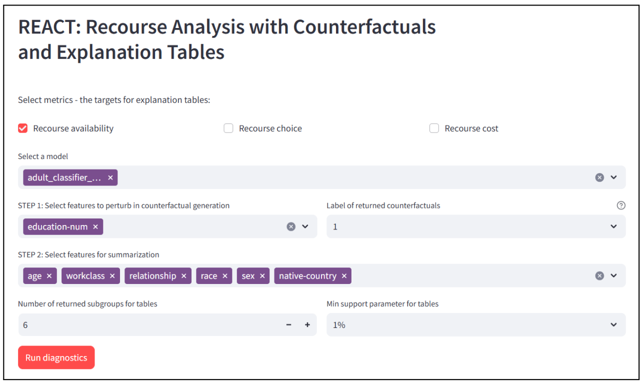
Each row of an explanation table is a rule that describes a subgroup. A subgroup is a conjunction of "attribute=value" conditions, with a star representing all possible values. The first row, all stars, corresponds to all examples with the undesirable label, where the average recourse cost is 34.2. The next rule states that for Sex=female, recourse cost is higher at 45.2. The next rule states that for age<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.

To extract unusual or surprising patterns, an explanation table identifies the k most informative subgroups with respect to some dependent variable, which is the recourse cost in this example. The first rule states that the average recourse cost in the test dataset is 34.2. The next rule is chosen to provide the most additional information about the distribution of recourse cost. In other words, the chosen rule has the most unusual or surprising distribution compared to the rules generated so far. Here, "female, *, *" is the most informative rule since recourse costs in this subgroup are significantly different from the average of 34.2. This process repeats k times, each time selecting the rule with the most information about the recourse cost distribution.

3 DEMONSTRATION PLAN

Participants will use REACT to produce recourse summaries for two preloaded test sets: Adult (a census dataset with demographic attributes and incomes) and Toronto (a police dataset with demographic attributes of arrested individuals and a label indicating whether they were strip-searched). For each test set, REACT includes several models to choose from.

Figure 3 shows the REACT front end. At the top, users select the recourse metrics of interest, followed by a model-testset pair.



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_... ×

STEP 1: Select features to perturb in counterfactual generation

education-num ×

Label of returned counterfactuals

1

STEP 2: Select features for summarization

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

Number of returned subgroups for tables

6

Min support parameter for tables

1%

Figure 3: The REACT front end.

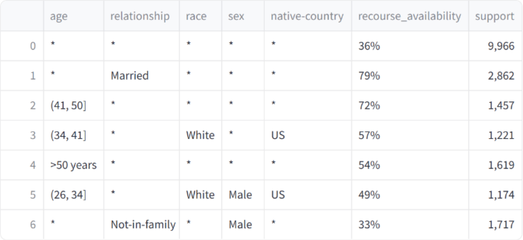
In the box labelled STEP 1, users specify f_C, the features to perturb in counterfactual generation, followed by the label of the returned counterfactuals (in the figure, the user wants counterfactuals to flip test examples to a positive label). In the box labelled STEP 2, users enter f_S, the subgroup features for explanation tables. At the bottom, users select k, the number of returned subgroups, and the minimum support. In the remainder of this section, we describe three starting points for user interaction with REACT.

3.1 Recourse Disparities in Salary Predictions

We start with the Adult Census Income dataset, which is widely used to evaluate algorithmic fairness. The features correspond to demographic information and the binary label indicates whether an individual earns more than $50,000 per year. Let us select the XGBoost classifier trained on this dataset. Among the ten features, sex and race are protected features; previous work on explicit bias analysis shows poor classification performance (explicit bias) for females and individuals from non-White racial groups. [1].

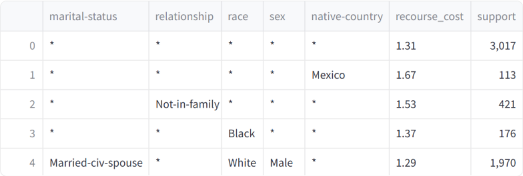
Suppose the user aims to assess the difficulty of flipping to a positive prediction (annual income over $50,000), selecting education level, with 16 distinct values from Preschool to Doctorate, to perturb in Step ①, *recourse availability* as a target metric, and f_S as shown in Figure 3. REACT outputs the explanation table shown in Figure 4. 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.

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. 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 Figure 5, 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*



	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

Figure 4: Recourse availability explanation table when changing *education level* to generate counterfactuals.



	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

Figure 5: Recourse cost, changing *workclass* and *occupation.*

shows that the model is more likely to predict higher income for males over 34 years if their hours were to increase.

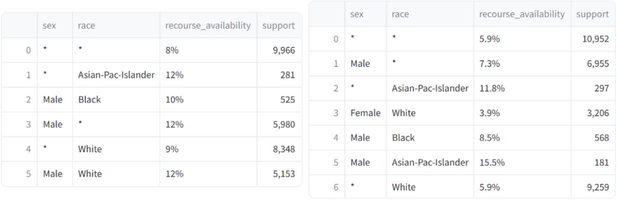
3.2 Fairness-Driven Model Comparison

To illustrate how participants 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 [9], we remove any correlation of input features with sex as a preprocessing step. Next, we apply GridSearch [1] to select model parameters in a way that balances accuracy (measured by the F1 score) and fairness (evaluated using statistical parity).

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 Figure 6). 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]. 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 overall accuracy, fairness in accuracy, and fairness in recourse.

3.3 Fairness in Pathways for Strip Decisions

The arrest and strip search dataset collected by Toronto police has a positive label if an arrestee was subject to removal of some or all clothing and a visual inspection of the body. 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 "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.



	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

Figure 6: Comparison of recourse availability when changing *hours per week* between the first model (explicitly biased, left) and the second model (right).

REACT shows that two protected subgroups—Black males and young White adults aged 25 to 34 years—would be predicted as strip-searched given more options of where they could have been arrested at, instead of their actual arrest location (averaging 3.02 and 4.03 recourse choices, 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.

REFERENCES

[1] Alekh Agarwal, Alina Beygelzimer, Miroslav Dudík, John Langford, and Hanna M. Wallach. 2018. A Reductions Approach to Fair Classification. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*, Vol. 80. PMLR, 60–69. http://proceedings.mlr.press/v80/agarwal18a.html

[2] Andrew Bell, João Fonseca, and Julia Stoyanovich. 2024. 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*. 464–467. https://doi.org/10.1145/3626246.3654742

[3] Yeounoh Chung, Tim Kraska, Neoklis Polyzotis, Ki Hyun Tae, and Steven Euijong Whang. 2020. Automated Data Slicing for Model Validation: A Big Data - AI Integration Approach. *IEEE Trans. Knowl. Data Eng.* 32, 12 (2020), 2284–2296. https://doi.org/10.1109/TKDE.2019.2916074

[4] Armin Esmaelizadeh, Sunil Cotterill, Liam Hebert, Lukasz Golab, and Kazem Taghva. 2025. Infomod: information-theoretic machine learning model diagnostics. *Distributed Parallel Databases* 43, 1 (2025), 6. https://doi.org/10.1007/S10619-024-07450-8

[5] Christos Fragkathoulas, Vasiliki Papanikou, Danae Pla Karidi, and Evaggelia Pitoura. 2024. On Explaining Unfairness: An Overview. In *40th International Conference on Data Engineering, ICDE 2024 - Workshops*. IEEE, 226–236. https://doi.org/10.1109/ICDEW61823.2024.00035

[6] Kareem El Gebaly, Guoyao Feng, Lukasz Golab, Flip Korn, and Divesh Srivastava. 2018. Explanation Tables. *IEEE Data Eng. Bull.* 41, 3 (2018), 43–51. http://sites.computer.org/debull/A18sept/p43.pdf

[7] Loukas Kavouras, Konstantinos Tsopelas, Giorgos Giannopoulos, Dimitris Sacharidis, Eleni Psaroudaki, Nikolaos Theologitis, Dimitrios Rontogiannis, Dimitris Fotakis, and Ioannis Z. Emiris. 2023. Fairness Aware Counterfactuals for Subgroups. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023*. http://papers.nips.cc/paper_files/paper/2023/hash/b60161e93f3e0e4207081a3b4ef5e8d8-Abstract-Conference.html

[8] Ramaravind Kommiya Mothilal, Amit Sharma, and Chenhao Tan. 2020. Explaining machine learning classifiers through diverse counterfactual explanations. In *FAT* '20: Conference on Fairness, Accountability, and Transparency, 2020*. ACM, 607–617. https://doi.org/10.1145/3351095.3372850

[9] Hilde J. P. Weerts, Miroslav Dudík, Richard Edgar, Adrin Jalali, Roman Lutz, and Michael Madaio. 2023. Fairlearn: Assessing and Improving Fairness of AI Systems. *J. Mach. Learn. Res.* 24 (2023), 257:1–257:8. https://jmlr.org/papers/v24/23-0389.html

[10] Andy Yu, Parke Godfrey, Lukasz Golab, Divesh Srivastava, and Jaroslaw Szlichta. 2024. CAMO: Explaining Consensus Across MOdels. In *40th IEEE International Conference on Data Engineering, ICDE 2024*. IEEE, 5493–5496. https://doi.org/10.1109/ICDE60146.2024.00436