FAIRLABEL: Correcting Bias in Labels

1st Srinivasan H Sengamedu
People Experience and Technology
Amazon
Seattle, USA
sengamed@amazon.com

2nd Hien Pham
People Experience and Technology
Amazon
Seattle, USA
hienpham@amazon.com

Abstract—There are several algorithms for measuring fairness of *ML models*. A fundamental assumption in these approaches is that the ground truth is fair or unbiased. In real-world datasets, however, the ground truth often contains data that is a result of historical and societal biases and discrimination. Models trained on these datasets will inherit and propagate the biases to the model outputs. We propose FAIRLABEL, an algorithm which detects and corrects biases in labels. The goal of FAIRLABELis to reduce the Disparate Impact (DI) across groups while maintaining high accuracy in predictions. We propose metrics to measure the quality of bias correction and validate FAIRLABEL on synthetic datasets and show that the label correction is correct 86.7% of the time vs. 71.9% for a baseline model. We also apply FAIRLABEL on benchmark datasets such as UCI Adult, German Credit Risk, and Compas datasets and show that the Disparate Impact Ratio increases by as much as 54.2%.

I. INTRODUCTION

With ML models playing a fundamental role in many decisions such as job applications, loan applications, and criminal justice decisions, Algorithmic Fairness has emerged as an important aspect of ML modeling. Typically, societal bias and historical discrimination manifest both in adverse decisions against the minority group and favorable decisions for the majority group. In this paper, we refer to the majority group as receiving favorable treatment and the minority group as receiving unfavorable treatment. The biases in decisions, whether conscious or unconscious, lead to biased data. ML models trained on these data, if steps are not taken to mitigate the inherent bias contained within them, can lead to biased model outputs and decisions. This leads to further propagation of bias.

In Algorithmic Fairness, there are several metrics such as *Disparate Impact Ratio (DIR)* that are defined to quantify the fairness of ML models. The U.S. Equal Employment Opportunity Commission has established a guideline termed the *Four-Fifth's or 80% Rule* which states that the selection rate for the minority class should not be less than four-fifths (80%) of that of the majority group. See [1] and [2]. Real world datasets such as UCI Adult dataset has a DIR ratio of male/female (salaries above $50K USD per annum) of 0.353 which is well below the acceptable threshold of 0.8 for disparate impact.

Manually correcting ground truth labels in real-world datasets is an impossible task as this means re-assessing historical decisions (such as loan and job applications) for which detailed data and information are no longer available. Instead, we propose an algorithm, FAIRLABEL, to correct biased labels directly. We propose a data generation framework to validate FAIRLABELin which we inject bias into the synthetic data and measure the algorithm's ability to find and correct the bias. We then, we show the performance of FAIRLABEL on ML benchmark datasets such as UCI Adult, German Credit Risk, and Compas.

The contributions of the paper are as follows.

1) We propose FAIRLABEL, an algorithm to identify and correct biases in labelled data.
2) We propose a framework to generate biased synthetic data to validate FAIRLABEL. We also propose relevant metrics for the task.
3) We demonstrate the performance of FAIRLABEL on UCI Adult, German Credit Risk, and Compas datasets and report improvements in DIR. The improvements range from 13.4% to 54.2%.

The paper is organized as follows. Section II discusses related work in the area of algorithmic fairness. Section III describes FAIRLABEL algorithm. Section IV outlines the synthetic data generation framework which introduces noise and bias to mimic real-world data and Section V defines relevant metrics for the debiasing task. Section VI contains details of synthetic and real-world datasets used and the performance of FAIRLABEL in terms of the debiasing metrics as well as Disparate Impact. Section VII concludes the paper.

II. RELATED WORK

There is rich research work on Algorithmic Fairness. A recent review on Bias and Fairness [3] lists the sources of bias as well as techniques to improve fairness. The sources of bias are often listed as biased features, selection bias, and label imbalance. Algorithmic bias is defined as *bias is not present in the data and is added purely by the algorithm*. Hence the problem of label bias has not received been addressed widely. For another, more recent survey, see [4]. The book [5] provides a broad as well as deep coverage of the topic.

The standard framework for achieving algorithmic fairness is either using pre-processing, mid-processing, or post-processing to achieve target fairness metrics. As mentioned earlier, all these frameworks *assume that the data is unbiased* and the unfairness springs mainly from feature representations or ML models. [6] discusses prevention of discrimination in

data mining. Pre-processing approaches include messaging [7], preferential sampling [8] [9], disparate impact removal [10] to remove biases from the data. Papers like [10] change features but *keep labels intact* while removing data biases. FAIRLABEL is complementary in the sense that it removes label bias.

There is relatively less research focus on biases in data. Biases in data often means selection or curation bias and biases in features in terms of missing features or missing values. One challenge in data collection is data skews [11] [12] [13] [14]. Analysis often happens on slices of data and patterns in individual slices and aggregates can be different. This is called Simpson Paradox [15]. [16] [17] [18] address Simpson Paradox related issues. [19] propose having labels to categorize data quality. Previous work has leveraged different loss function [20] instead of correcting data bias.

Real-world data often has biased labels. Getting the data relabeled is usually not an option. At the same time, it is essential to use historical data to build models. However, the issue of bias in ground truth labels and debiasing them has received relatively less research focus. Synthetic data generation has been used to mitigate the problem [21] [22]. However, these methods care more about privacy than bias.

Counterfactual Analysis is a popular causal analysis approach. Research on Counterfactual Fairness [23] mentions the issues of biased ground truth. The approach taken in Counterfactual Fairness is changing *feature values* to achieve fairness. The approach is based on structural models with latent variables. The challenge with this approach is the is the validation. [24] also uses causal approaches.

FairSMOTE [25] is an algorithm to detect biased labels. FairSMOTE is based on 'situation testing': flip the sensitive attribute and check if the label has changed. The paper does not propose metrics to define the effectiveness of the approach. It is interesting to note that only the training data is debiased and not the test data. The reason is that the model solving the task is also used for debiasing. We decouple the two tasks and use a separate model for label debiasing.

The challenge with these approaches is that they do not consider the fact that *bias is asymmetric or directional and unbalanced.* In other words, one demography is penalized and the other demography is advantaged (*asymmetry*) and one of the demography is often under-represented (*unbalanced*). We need principled approaches to deal with these issues.

Appendix A lists metrics used in Algorithmic Fairness. We primarily use Disparate Index Ratio from an application perspective. For the newly defined task of label debiasing, we define metrics in Section V.

III. FAIRLABEL

The intuition behind FAIRLABEL is simple. In the minority class, a biased decision occurs when the decision maker makes a negative decision despite the person having the requisite qualifications. The decision maker in this case would make the opposite decision if the person belonged to the majority class. The decision maker, whether conscious or subconscious, is acting in a biased manner against the minority class where

the probability of a favorable outcome for the minority group is less than that of the majority group. Expressed mathematically this is:

$$P(y = 1|p = minority) < P(y = 1|p = majority)$$

Thus, ground truth labels in certain applications become biased. Measuring the performance of a machine learning model against biased labels without any bias mitigation poses problems of perpetuating bias downstream.

The first goal of FAIRLABELis to debias ground truth labels. This is an inherently ill-posed problem because (1) the real unbiased ground truth is not known and (2) asking human experts to relabel is either infeasible (e.g., prohibitive cost and insufficient data to make decisions) or not advisable (i.e., biases may still be present) or both. The recommended approach in the literature is Situation Assessment (SA). SA can correct the labels of both majority and minority groups. We propose a variant of this based on the fundamental observation that we first want to correct labels only of the minority group in only one direction ($0 \rightarrow 1$). This is inline with the real-world biases acting against minority group. This process also ensures the positive biases received by the majority group are reflected in the correction for minority group. We call this process FAIRMIN because the minority group's data is corrected. The process is as follows:

1) **Split data into majority and minority groups:** Create training and validation data consisting only of the majority group.
2) **Train a classifier on majority group:** Build a model using only the majority group's data. This can be an ensemble of classifiers or any single classification algorithm. This ensures the model only learns the patterns from the majority group.
3) **Run inference on minority group:** Get predictions for minority class using the trained model in the previous step.
4) **Flip labels of minority group:** For a given data point where the ground truth label is 0 and the model predicts 1, flip the label, i.e. the label transforms '$0 \rightarrow 1$' for all instances where prediction=1. We can can use a hold-out set to determine threshold for '$0 \rightarrow 1$' label change so that DIR is close to 1.
5) **Concatenate majority and minority datasets:** After flipping labels in the minority dataset, combine the majority dataset with the modified/debiased minority dataset. The resulting dataset can be considered to be debiased.

In several scenarios, not only does the bias manifest against the minority group, the data also has bias in favor of the majority group. To remove such bias, we complement the above process and debias the majority group dataset. Similar to FAIRMIN, we split the dataset by majority/minority groups, but train a classifier only on the minority dataset and flip the labels of the majority group from 1 to 0 if the prediction=0.

We call this process FAIRMAJ and the detailed steps are as follows:

1) **Split data into majority and minority groups:** Create training and validation data consisting only of the minority group.
2) **Train a classifier on minority group:** Build a model using only the minority group's data. This can be an ensemble of classifiers or any single classification algorithm. This ensures the model only learns the patterns from the minority group.
3) **Run inference on majority group:** Get predictions for majority class using the trained model in the previous step.
4) **Flip labels of majority group:** For a given data point where the ground truth label is 1 and the model predicts 0, flip the label, i.e. the label transforms '1 → 0' for all instances where prediction=0. We can can use a hold-out set to determine threshold for '1 → 0' label change so that DIR is close to 1.
5) **Concatenate majority and minority datasets:** After flipping labels in the majority dataset, combine the minority dataset with the modified/debiased majority dataset. The resulting dataset can be considered to be debiased.

For FAIRLABEL, we first run FAIRMIN and then, optionally, run FAIRMAJ. Depending on the level of bias inherent in the data, whether it is present in the minority decisions or majority decisions, we can choose to run FAIRMIN or FAIRMAJ, or both.

IV. SYNTHETIC DATASET FOR CLASSIFICATION

We validate the label flipping using synthetically generated data. By injecting bias into synthetic data, we have the ability to track where the bias was added, an attribute of the synthetic data that is not possible in real-world datasets. Consider a classification problem which has some protected attributes like gender and race. To generate biased synthetic data for this, we need the following.

- **Independent Variables:** These are the aspects of data such as numerical, categorical, and textual features. The variables necessarily include protected attributes
- **Data Model:** This is the underlying data generation process. The data generation depends on all the independent variables excluding protected attributes. Inclusion of protected attributes is optional. We can use a range of techniques like linear regression, mixture models, decision trees, etc. to generate clean and unbiased data.
- **Noise:** Real-world data is noisy. Both features and labels can be noisy. We use a noise model to introduce noise in non-protected attributes and run the model to generate noisy labels. If the data model is simple, this may not introduce noise in labels. So also introduce noise in labels. *The noise is independent of protected attributes.*
- **Bias:** *Bias can be considered systematic label noise which depends on protected attributes.* The noise is systematic

in the sense that it is unidirectional: either 0→1 (for majority) or 1→0 (for minority) based on protected attributes. The severity of bias is given by bias probability.

A. Synthetic Data Generation

Real-world datasets have both bias and noise. Noise can affect both features and labels. *Noise is independent of the protected class.* We quantify noise as ϵ. Bias can be considered noise except that *bias is based on the protected attribute and is unidirectional*. See Figure 1 for Synthetic data generation as well as its use in FAIRLABEL evaluation.

B. Linear Synthetic Data Generation Example

Clean Data: As an example, let us consider 10 unprotected numerical features (x) and 1 protected categorical feature (z) for a linear classification problem.

$$y = f(a^T x + b)$$

where a are the coefficients, b is the intercept, and $f(\cdot)$ is the logistic function. The model is specified by 11 random coefficients for x and b. We now randomly generate N data points based on random x. The protected attribute z is also randomly generated (with certain distribution) but not fed into the model.

Noisy Data: We now introduce noise to the data.

$$y' = f(a^T x' + b + \epsilon)$$

where x' and ϵ are noisy versions.

Biased Data: We now fix the protected attribute value (v), bias direction (d) and bias severity (p).

1) Loop over records with protected attribute value, $z = v$.
   a) Choose a record with probability p.
   b) Change the label in the direction d.
2) Store the changes as metadata.

V. METRICS FOR DEBIASING TASK

Assume that the ground truth is known. Assume negative bias based on the unprotected attribute. In this case, FAIRLABEL recommends $0 \rightarrow 1$ flips for the minority class. Some of the flips are correct and TPR is the fraction of correct flips. TNR is the fraction of missed flips. FPR and TNR are not applicable. We rename these metrics as Correct Flip Rate (CFR) and Missed Flip Rate (MFR). We report CFR and MFR for the majority class too. In case of the majority class, the metrics are reported for $1 \rightarrow 0$ flips.

It is not possible to measure CFR and MFR in real-world datasets. Instead, we consider standard observational metrics related to bias and fairness. Definitions are as follows:

Demographic parity or statistical parity: it suggests that a predictor is unbiased if the prediction $\hat{y}$ is independent of the protected attribute p so that:

$$Pr(\hat{y}|p) = Pr(\hat{y})$$

Fig. 1. Synthetic data generation and validation of FAIRLABEL

Fig. 2. Evaluation of DI metrics

Disparate Impact Ratio (DIR): The ratio of the demographic parity:

$$\frac{Pr(\hat{y} = 1 | p = minority)}{Pr(\hat{y} = 1 | p = majority)}$$

Disparate Impact Difference (DID): The difference in demographic parity:

$$Pr(\hat{y} = 1 | p = minority) - Pr(\hat{y} = 1 | p = majority)$$

VI. EXPERIMENTAL RESULTS

A. Synthetic Dataset

We ran several experiments comparing FAIRLABEL to a Naive ML model for debasing using three different synthetic datasets listed below. The baseline Naive approach is described in Section VI-B.

- **Linear:** Linear dataset created by logistic function as described in section IV-B.
- **Clusters around n-hypercubes:** This initially creates clusters of points normally distributed (std=1) about vertices of an n-informative-dimensional hypercube and assigns an equal number of clusters to each class. It introduces interdependence between these features and adds various types of further noise to the data. We use 8-dimensional hypercube with an edge length of 0.5.
- **Gaussian Quantiles:** This classification dataset is constructed by taking a multi-dimensional standard normal distribution and defining classes separated by nested concentric multi-dimensional spheres such that roughly equal numbers of samples are in each class (quantiles of the distribution).

Additional details of the data generation are listed in Appendix B. For each of the synthetic datasets, we generated 100,000 samples with 10 features and a binary class for the label. In addition, we experimented with Logistic Regression, Random Forest, and Gradient Boosted Tree algorithms for the underlying ML ensemble model, with an 80/20 random split between train and test sets. We vary the bias injection rate in the synthetic data to understand how the amount of bias inherent in the data affects FAIRLABEL.

B. Baseline for Debiasing

The baseline for the debiasing does not take directionality of bias into consideration. The debiasing is based on training an ML model on the full data and flipping labels of the minority class from 0 to 1 based on classifier prediction. We call this the Naive approach. Figure 1 shows how we perform the baselining.

C. Benchmark Datasets

We use three benchmark datasets: UCI Adult, German Credit Risk, and Compas.

- **UCI Adult:** The dataset has 14 attributes and task is to predict whether income exceeds $50K/yr based on census data. Also known as "Census Income" dataset. The protected attributes in the dataset are age, gender, and race. We use gender as the protected attribute.
- **German Credit Risk:** The dataset contains 20 attributes and the task is to classify people described by a set of attributes as good or bad credit risks. The protected attribute is sex.

- **Compas:** The dataset has 10 attributes and the target is prediction of two-year recidivism. The protected attributes are race and sex. For our purpose, we used race binarized to 'Caucasian' (majority) and 'African-American' (minority).

Table I summarizes the data sets.

D. Results

1) Results on Synthetic Datasets: We compare the Correct Flip Rate (CFR), Missed Flip Rate (MFR) of datasets D2 and D3, and F1-score (of Models M1 and M2) across multiple synthetic datasets where the debiased datasets produced by FAIRLABEL and the Naive models (D2 and D3) are compared against the dataset with known bias (D1). Table II summarizes the results of FAIRLABEL across these metrics with the bias injection rate held constant at 0.2. Results show that FAIRLABEL corrects more bias than the Naive model. Also, FAIRLABEL does not sacrifice accuracy in order to capture more bias flips as it outperforms the Naive model on F1-score. This is true across the entire range of synthetic datasets. Bias acts as noise to an ML model and FAIRLABEL is not adversely affected by bias in the minority class because the FAIRLABEL is only trained on the majority class.

FAIRLABEL incorrectly flips labels at a slightly higher rate than the Naive approach. This pattern holds true across datasets and model types. i.e there is a penalty for flipping labels in a unidirectional manner. In real world-applications, this means some people in the minority class will be "incorrectly" boosted.

In general, FAIRLABEL is more robust to the amount of bias present in the data in terms of CFR and F1-score vs. the Naive models across different levels of bias. For the data in Figures 3, 4, and 5, we add bias to the synthetic datasets at various proportions to see how the bias proportion affects the CFR, MFR, and F1-score, respectively.

The US Equal Employment Opportunity Commission states that bias is acceptable when DIR is between 0.8 and 1.0. Checking DIR in our models trained on synthetic datasets, we see that models trained on debiased datasets produced by FAIRLABEL does better than those trained on the original (biased) datasets, regardless of the amount of bias in the data, with DIR close to 1 and DID difference to be close to 0. See Figure 6.

2) Results on Benchmark Datasets: The advantage or running analyses on synthetic datasets is the ability to measure CFR and MFR because we know with certainty which labels are biased. Real-world datasets, however, do not have this advantage. Instead we look at aggregate-level metrics, such as as demographic parity and disparate impact.

We ran FAIRLABEL on the datasets Adult, German Credit Risk, and Compas. Figure 2 shows the evaluation setup. We then ran the AI Fairness Package (aif360) and checked Disparate Impact Ratio both on models trained on the debiased dataset created by FAIRLABEL and on models trained on the original datasets. The results are in Table III. Across each dataset, the DIR improved from +0.356 (UCI) to +0.542

(Compas), showing that FAIRLABEL has the ability to reduce disparity between groups.

VII. CONCLUSIONS

In this paper, we have considered the fundamental issue of bias in ground truth labels and have proposed an intuitive approach to address the issue: bias is directional and affects subsets of data differently. The algorithm to codify the intuition has to parts: FAIRMIN and FAIRMAJ. FAIRMIN addresses the problem of negative bias for minority attribute while FAIRMAJ addresses the potential positive bias for majority attribute. The final algorithm, FAIRLABEL, applies both FAIRMIN and FAIRMAJ iteratively to remove bias. We have defined metrics to characterize the performance of the algorithms and have proposed a synthetic data generation framework for validating the approaches. We show that FAIRLABEL reduces DIR by almost 55% in some cases. We believe the results presented in the paper will inspire the adoption of FAIRLABEL to other fairness problems. We also hope that the synthetic data generation will stimulate other data generation as well as label debiasing approaches because it makes benchmarking feasible. Our own goal is to extend FAIRLABEL to other modalities like text and images.

REFERENCES

[1] The U. S. Equal Employment Opportunity Commission (EEOC). Uniform guidelines on employee selection procedures, 1979.

[2] NYC Consumer and Worker Protection. https://www.nyc.gov/site/dca/about/new-laws-rules.page.

[3] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *ACM Computing Surveys*, 2021.

[4] Dana Pessach and Erez Shmueli. A review on fairness in machine learning. *ACM Computing Surveys*, 2022.

[5] Solon Barocas, Moritz Hardt, and Arvind Narayanan. *Fairness and Machine Learning: Limitations and Opportunities*. fairmlbook.org, 2019. http://www.fairmlbook.org.

[6] S. Hajian and J. Domingo-Ferrer. A methodology for direct and indirect discrimination prevention in data mining. *IEEE Transactions on Knowledge and Data Engineering*, 25(7):1445–1459, 2013.

[7] F. Kamiran and T. Calders. Classifying without discriminating. In *2009 2nd International Conference on Computer, Control and Communication*, pages 1–6, 2009.

[8] Faisal Kamiran and Toon Calders. Data preprocessing techniques for classification without discrimination. *Knowledge and Information Systems*, 33(1):1–33, Oct 2012.

[9] Faisal Kamiran and Toon Calders. Classification with no discrimination by preferential sampling. In *Proc. 19th Machine Learning Conf. Belgium and The Netherlands*, pages 1–6. Citeseer, 2010.

[10] Michael Feldman, Sorelle A. Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. Certifying and removing disparate impact. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, page 259–268, New York, NY, USA, 2015. Association for Computing Machinery.

[11] Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé III, and Kate Crawford. Datasheets for datasets. *dd*, 2019.

[12] Misha Benjamin, Paul Gagnon, Negar Rostamzadeh, Chris Pal, Yoshua Bengio, and Alex Shee. Towards standardization of data licenses: The montreal data license. *dd*, 2019.

[13] Emily M. Bender and Batya Friedman. Data statements for natural language processing: Toward mitigating system bias and enabling better science. *Transactions of the Association for Computational Linguistics*, 6:587–604, 2018.

TABLE I
SUMMARY OF DATASETS.

Dataset	Number of records	Number of features	Target	Protected Attribute
UCI Adult	48,842	14	salary>$50k/yr	gender
German Credit Risk	1,000	20	credit risk	gender
Compas	6,167	10	two-year recidivism	race

Fig. 3. CFR vs bias injection rate

Fig. 4. MFR vs bias injection rate

Fig. 5. F1-score vs bias injection rate

TABLE II

FAIRLABEL VS NAIVE MODEL OVERALL RESULTS, BIAS INJECTION RATE = 0.2

Model	Correct Flip Rate (CFR)	Missed Flip Rate (MFR)	F1-score
Naive model	0.7192 ± 0.1647	**0.1145 ± 0.0640**	0.7800 ± 0.09521
FAIRLABEL	**0.8668 ± 0.0727**	0.2002 ± 0.1040	**0.8818 ± 0.0364**
Gain	0.1476	-0.0858	0.1017

TABLE III

DISPARATE IMPACT RATIO (DIR) OF *aif360* DATASETS

Dataset	UCI	German Credit Risk	Compas
Original (not debiased)	0.3122	0.896	0.618
Debiased (FAIRLABEL)	0.665	1.03	1.16
Gain	0.356	0.134	0.542

Fig. 6. Disparate Impact vs bias injection rate, XGboost

[14] Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, and Timnit Gebru. Model cards for model reporting. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, FAT* '19, 2019.

[15] Edward H Simpson. The interpretation of interaction in contingency tables. *Journal of the Royal Statistical Society*, 1951.

[16] Rogier Kievit, Willem Eduard Frankenhuis, Lourens Waldorp, and Denny Borsboom. Simpson's paradox in psychological science: a practical guide. *Frontiers in psychology*, 4:513, 2013.

[17] Nazanin Alipourfard, Peter G Fennell, and Kristina Lerman. Can you trust the trend?: Discovering simpson's paradoxes in social data. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pages 19–27. ACM, 2018.

[18] Nazanin Alipourfard, Peter G Fennell, and Kristina Lerman. Using simpson's paradox to discover interesting patterns in behavioral data. In *Twelfth International AAAI Conference on Web and Social Media*, 2018.

[19] Sarah Holland, Ahmed Hosny, Sarah Newman, Joshua Joseph, and Kasia Chmielinski. The dataset nutrition label: A framework to drive higher data quality standards. *arXiv preprint arXiv:1805.03677*, 2018.

[20] Heinrich Jiang and Ofir Nachum. Identifying and correcting label bias in machine learning. In *International Conference on Artificial Intelligence and Statistics*, pages 702–712. PMLR, 2020.

[21] Weijie Xu, Jinjin Zhao, Francis Iannacci, and Bo Wang. Ffpdg: Fast, fair and private data generation, 2023.

[22] Tamas Madl, Weijie Xu, Olivia Choudhury, and Matthew Howard. Approximate, adapt, anonymize (3a): a framework for privacy preserving training data release for machine learning. *ArXiv*, abs/2307.01875, 2023.

[23] Matt Kusner, Joshua Loftus, Chris Russell, and Ricardo Silva. Counterfactual fairness. In *NeurIPS*, 2017.

[24] Lu Zhang, Yongkai Wu, and Xintao Wu. Achieving non-discrimination in data release. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1335–1344. ACM, 2017.

[25] Joymallya Chakraborty, Suvodeep Majumder, and Tim Menzies. Bias in machine learning software: Why? how? what to do? In *ESC/FSC*, 2021.

Appendix

APPENDIX A
FAIRNESS MEASURES

Table IV summarizes standard fairness metrics.

APPENDIX B
DETAILS OF SYNTHETIC DATA GENERATION

N_hypercubes and Gaussian quantiles are from the `sklearn` package and the linear dataset was a custom function shown in C

TABLE IV
METRICS USED AS FAIRNESS CRITERIA

Metric		Definition
EO	Equalized Odds	$P(\hat{Y}=1|A=0,Y=y) = P(\hat{Y}=1|A=1,Y=y)$, $y \in \{0,1\}$
DP	Demographic Parity	$P(\hat{Y}|A=0) = P(\hat{Y}|A=1)$
DI	Disparate Impact	$\dfrac{\frac{TP_p+FP_p}{N_p}}{\frac{TP_u+FP_u}{N_u}}$
EOO	Equal of Opportunity	$\dfrac{TP_p}{TP_p+FN_p} - \dfrac{TP_u}{TP_u+FN_u}$
KNNC	K-Nearest Neighbors Consistency	
ABAD	Absolute Balanced Accuracy Difference	$\left| \frac{1}{2} \left[TPR_p + TNR_p \right] - \left[TPR_u + TNR_u \right] \right|$
AAOD	Absolute Average Odds Difference	$\left| \dfrac{(FPR_u+FNR_p)-(TPR_u+TPR_p)}{2} \right|$
AEORD	Absolute Equal Opportunity Rate Difference	$|TPR_p - TPR_u|$
SPD	Statistical Parity Difference	$\dfrac{TP_p+FP_p}{N_p} - \dfrac{TP_u+FP_u}{N_u}$

Appendix C
Linear Synthetic Data Generation Code

```python
def generate_linear_dataset(n_samples, n_features, p_noise, seed):
    import pandas as pd

    #compute sample counts
    n_samples_perfect = int(n_samples*(1-p_noise))
    n_samples_noise = n_samples - n_samples_perfect

    #weights of the model
    w = generate_random_coefficients(n_features,seed=seed)

    #random X's
    X = generate_random_x(n_features,n_samples_perfect,seed=seed+20)
    b = 0

    #compute y (perfect y)
    probs = sigmoid(np.dot(X,w) + b)
    y = np.array([1 if i > 0.5 else 0 for i in probs]).reshape(n_samples_perfect,)

    #noisy data
    X_noise = generate_random_x(n_features,n_samples_noise,seed=seed+40)
    y_noise = generate_random_binary(n_samples_noise, seed=50)

    #combine perfect data with noisy data
    X_full = np.concatenate((X,X_noise))
    y_full = np.concatenate((y,y_noise))

    return X_full, y_full
```