UNIVERSIDADE FEDERAL DE MINAS GERAIS
Instituto de Ciências Exatas
Programa de Pós-Graduação em Ciência da Computação

Rodrigo Lemos Cardoso

A Framework for Benchmarking Discrimination-Aware Models in Machine Learning

Belo Horizonte
2020

Rodrigo Lemos Cardoso

A Framework for Benchmarking Discrimination-Aware Models in Machine Learning

Final Version

Thesis presented to the Graduate Program in Computer Science of the Federal University of Minas Gerais in partial fulfillment of the requirements for the degree of Master in Computer Science.

Advisor: Wagner Meira Júnior
Co-Advisor: Virgílio Augusto Fernandes Almeida

Belo Horizonte
2020

2020, Rodrigo Lemos Cardoso.
Todos os direitos reservados

Cardoso, Rodrigo Lemos.

C268f A Framework for benchmarking discrimination-aware models in machine learning / Rodrigo Lemos Cardoso Belo Horizonte – 2020.
1 recurso online (83 f. il., color.) : pdf.

Orientador: Wagner Meira Júnior.
Coorientador: Virgílio Augusto Fernandes Almeida.

Dissertação (Mestrado) - Universidade Federal de Minas Gerais, Instituto de Ciências Exatas, Departamento de Ciência da Computação.
Referências: f. 80-83

1. Computação – Teses. 2. Aprendizado do computador – Teses. 3. Benchmarking (Administração) – Teses. 4. Mineração de dados (Computação). 5. Discriminação – Teses. I. Meira Júnior, Wagner. II. Almeida, Virgílio Augusto Fernandes. III.Universidade Federal de Minas Gerais, Instituto de Ciências Exatas, Departamento de Ciência da Computação. IV. Título.

CDU 519.6*73(043)

Ficha catalográfica elaborada pela bibliotecária Irénquer Vismeg Lucas Cruz
CRB 6/819 - Universidade Federal de Minas Gerais - ICEx

UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

FOLHA DE APROVAÇÃO

A Framework for Benchmarking Discrimination-Aware Models in Machine Learning

RODRIGO LEMOS CARDOSO

Dissertação defendida e aprovada pela banca examinadora constituída pelos Senhores:

PROF. WAGNER MEIRA JÚNIOR - Orientador
Departamento de Ciência da Computação - UFMG

PROF. VIRGÍLIO AUGUSTO FERNANDES ALMEIDA - Coorientador
Departamento de Ciência da Computação - UFMG

PROF. FABRÍCIO BENEVENUTO DE SOUZA
Departamento de Ciência da Computação - UFMG

PROF. PEDRO OLMO STANCIOLI VAZ DE MELO
Departamento de Ciência da Computação - UFMG

Belo Horizonte, 18 de Fevereiro de 2020.

Dedico este trabalho aos meus familiares, amigos, professores, colegas, e ao restante do mundo.

Acknowledgments

Meus mais profundos agradecimentos por fazer parte do Programa de Pós Graduação em Ciência da Computação da UFMG. Acredito que esta experiência transformou minha mentalidade acadêmica.

Agradeço a todos os meus amigos, com quem sempre puder contar; Família, por me apoiaram nas minhas decisões; colegas, que compartilharam experiências e me proporcionaram boas discussões sobre este trabalho; e pessoas que tive o prazer de explicar sobre este trabalho pois assim pude solidificar minha aptidão em fazê-lo.

Dentre meus amigos e gostaria de agradecer Derick e Túlio, com suas opiniões e críticas construtivas sobre o trabalho; Lucas, pelas oportunidades de espairecer; E meus pais, Djalma e Mércia, pela paciência e por entenderem minhas ocasionais absências.

Este trabalho foi parcialmente financiado por CNPq, CAPES, FAPEMIG, e pelos projetos MASWEB, INCT-Cyber e Atmosphere.

"*Do everything by hand, even when using the computer.*"
(Hayao Miyazaki)

Resumo

Nos últimos anos, o campo de pesquisa de aprendizado de máquina ciente de discriminação vem firmemente se desenvolvendo. É considerado um tópico importante pois há evidências que decisões automáticas feitas por tais modelos podem apresentar viés contra certos grupos e podem gerar implicações legais. Muitas técnicas para reduzir viés de processos de aprendizado foram propostas, cada uma com suas vantagens e desvantagens. Apesar de aparentemente este campo de pesquisa já estar bem desenvolvido devido a este possuir muitas métricas de validação e técnicas, ainda há demanda de conjuntos de dados para validação. Com a meta de melhorar o processo de avaliar a qualidade das técnicas, nós propomos um arcabouço para medir qualidade de modelos cientes de discriminação. Também mostramos três maneiras de gerar dados de validação para estes modelos em uma abordagem baseada em Redes Bayesianas. Nossos experimentos mostram resultados promissores que podem ajudar pesquisadores a desenvolverem suas técnicas.

Palavras-chave: equidade algorítmica; aprendizado de máquina; avaliação comparativa.

Abstract

Discrimination-aware machine learning has been steadily improving in the last few years. It is considered an important topic of research since it has been observed that automatic decisions made by machine learning models may present some bias against certain groups, which may have legal implications. A number of techniques aimed at reducing bias of the learning process have already been proposed, each one with different advantages and drawbacks. Although it appears that this field is well developed by having many metrics of evaluation and techniques, it still lacks on assessment. With the objective of improving the process of assessing the quality of a technique, we propose a novel framework for benchmarking discrimination-aware models. We also introduce three ways of generating validation data for those models using a Bayesian Network approach. Our experiments show promising results that may help researchers on discrimination-aware learning to develop their techniques.

Keywords: fairness; machine learning; benchmark.

List of Figures

2.1 Re-weighting (a) and Odds Equalizer (b) for data generation type 'A' on adult census dataset. . . . 25
2.2 Re-weighting (a) and Odds Equalizer (b) for data generation type 'B' on adult census dataset. . . . 26
2.3 Re-weighting (a) and Odds Equalizer (b) for data generation type 'C' on adult census dataset. . . . 27
2.4 Re-weighting (a) and Odds Equalizer (b) for data generation type 'A' on COMPAS dataset. . . . 28
2.5 Re-weighting (a) and Odds Equalizer (b) for data generation type 'B' on COMPAS dataset. . . . 29
2.6 Re-weighting (a) and Odds Equalizer (b) for data generation type 'C' on COMPAS dataset. . . . 30
2.7 Flowchart representing the integration of a technique into our framework. A dataset, in that case, the artificial data generated from our methodology, is used as input for the framework. A train-test split is done, and a technique is selected. The main gray area selects the proper treatment for the technique type, and a Learning Model is the output. The validation is performed according to selected metrics. . . . 31

4.1 Methodology . . . 38
4.2 Bayesian network learned from the adult dataset. . . . 43
4.3 Benchmark result summarizing methodology using 2-degree polynomial regression with root squared standard deviation . . . 48
4.4 Compact arrow representation of summarized result. . . . 49
4.5 Use Case 1 - Comparing two techniques: 2-degree polynomial regression . . . 50
4.6 Use Case 1 - Comparing two techniques: Arrow representation . . . 51
4.7 Use Case 2 - Choosing a technique in a quadrant given Minimum Fairness Metric Improvement (FMI) and Maximum Quality Metric Degradation (QMD): Min. FMI 0.6 Max. QMD 0.6 . . . 52
4.8 Use Case 2 - Choosing a technique in a quadrant given Minimum Fairness Metric Improvement (FMI) and Maximum Quality Metric Degradation (QMD): Min. FMI 0.4 and Max. QMD 0.8 . . . 53
4.9 Use Case 3 - Parameter Tuning of a bias mitigation technique: Dataset 1 (UCI Adult) . . . 54

4.10 Use Case 3 - Parameter Tuning of a bias mitigation technique: Dataset 2 (Dutch census) 55

6.1 Bias vs. Accuracy on artificially generated datasets learned from adult census (the classifier does not use the protected attribute). 65
6.2 Bias vs. Accuracy on artificially generated datasets learned from COMPAS recidivism data (the classifier does not use the protected attribute). 65
6.3 Bias vs. Accuracy on artificially generated datasets learned from Dutch census (the classifier does not use the protected attribute). 66
6.4 Bias vs. Accuracy on artificially generated datasets learned from German credit data (the classifier does not use the protected attribute). 66
6.5 Bias vs. Accuracy for Massaging technique trained on artificially generated datasets learned from adult census. 67
6.6 Bias vs. Accuracy for Re-weighting technique trained on artificially generated datasets learned from adult census. 67
6.7 Bias vs. Accuracy for Uniform Sampling technique trained on artificially generated datasets learned from adult census. 68
6.8 Bias vs. Accuracy for Preferential Sampling technique trained on artificially generated datasets learned from adult census. 68
6.9 Bias vs. Accuracy data Auditor technique at 0.2 repair level trained on artificially generated datasets learned from adult census. 69
6.10 Bias vs. Accuracy data Auditor technique at 0.8 repair level trained on artificially generated datasets learned from adult census. 69
6.11 Overestimation and underestimation on artificially generated datasets learned from adult census (the classifier does not use the protected attribute). 72
6.12 Overestimation and underestimation on artificially generated datasets learned from COMPAS recidivism data (the classifier does not use the protected attribute). 72
6.13 Overestimation and underestimation on artificially generated datasets learned from Dutch census (the classifier does not use the protected attribute). 73
6.14 Overestimation and underestimation on artificially generated datasets learned from German credit data (the classifier does not use the protected attribute). . 73
6.15 Summarized representation quality vs fairness: Adult census. 74
6.16 Summarized representation quality vs fairness: COMPAS. 75
6.17 Summarized representation quality vs fairness: Dutch census. 76
6.18 Summarized representation quality vs fairness: German credit. 77

List of Tables

4.1 Toy conditional probability table . . . 42
4.2 Conditional probabilities of marital status given some age categories (1-younger,10-older) . . . 43
4.3 Conditional probabilities of high income (above $50K a year) given education and relationship . . . 44

5.1 Accuracy obtained from different classifiers on biased data given a bias level on adult dataset . . . 59
5.2 Accuracy obtained from different classifiers on biased data given a bias level on COMPAS dataset . . . 59
5.3 Properties of the Bayesian networks learned using Hill-Climbing algorithm for each dataset . . . 59
5.4 Properties of the Bayesian networks learned using Tabu search algorithm for each dataset . . . 59

6.1 Bias and Accuracy on biased datasets from Adult, COMPAS and Dutch data (classifier does not use the protected attribute) . . . 62
6.2 Bias and Accuracy for data pre-processing techniques trained on artificially generated datasets learned from adult census. . . . 62
6.3 Bias and Accuracy for Auditor technique. . . . 63
6.4 Discrimination and Accuracy for Zafar's in-processing techniques trained on artificially generated datasets learned from adult census. . . . 64
6.5 Overestimation and underestimation for Adult, COMPAS and Dutch (classifier does not use the protected attribute). . . . 70
6.6 Overestimation and underestimation for data pre-processing techniques. . . . 70
6.7 Overestimation and underestimation for Auditor technique. . . . 71

Contents

1 Introduction **14**

1.1 Discrimination-Aware Learning . . . 15

1.1.1 Techniques . . . 16

1.1.2 Assessment . . . 16

1.2 Motivation . . . 17

1.3 Contributions . . . 18

2 Related Work **19**

2.1 Bias Discovery . . . 19

2.2 Bias Mitigation . . . 19

2.2.1 Pre-Processing . . . 20

2.2.2 In-Processing . . . 21

2.2.3 Post-Processing . . . 22

2.3 Data Generation . . . 22

2.4 AIF360 . . . 23

2.4.1 Metrics . . . 23

2.4.2 Techniques . . . 23

2.4.3 Benchmark Framework and AIF360 . . . 24

2.4.3.1 Sample Results . . . 24

2.5 Discussion . . . 24

3 Definitions **33**

3.1 Protected and Non-Protected Attributes . . . 33

3.2 Class Labels . . . 34

3.3 Bias Metric . . . 34

3.3.1 Single Attribute . . . 34

3.3.2 Multiple Attributes . . . 35

3.3.3 Classification Metrics . . . 36

3.3.4 Additional Metrics . . . 37

4 Benchmark Methodology **38**

4.1 Bayesian Networks . . . 38

4.2 Estimating Bayesian Networks . . . 39

4.3 Overview . . . 39

4.4 Generating Bias-variant data . . . 41
4.5 Practical Example . . . 42
4.6 Summarizing results . . . 44
4.7 Framework Use Cases . . . 48

5 Experiments **56**
5.1 Bias Mitigation Techniques . . . 56
5.2 Baseline . . . 57
5.3 Datasets . . . 57
5.4 Bayesian Network Learning . . . 58
5.5 Experimental Setup . . . 60
5.6 Artificially Generated Datasets . . . 60

6 Results and Discussion **61**
6.1 Bias vs. Accuracy tables . . . 61
6.2 Bias vs. Accuracy graphs . . . 64
6.3 Overestimation vs. Underestimation tables . . . 67
6.4 Overestimation vs. Underestimation graphs . . . 71
6.5 Compact Representation . . . 73
6.6 Summary . . . 75

7 Conclusions **78**
7.1 Final Remarks . . . 78
7.2 Future Directions . . . 79

References **80**

Chapter 1

Introduction

In the Social Sciences field, **Discrimination** is defined as the act of distinguishing groups of individuals according to their characteristics. This discrimination can be socially considered **unfair** if the criteria for discrimination is a characteristic that is usually protected by law. Examples of such discrimination may be the act of denying a specific service (i.e., credit approval) to a person under their race, sex, or age.

Brazil's constitution [12] states that the union should *"promote the well-being of all, without prejudice as to origin, race, sex, color, age and any other forms of discrimination"* and it is forbidden *"any difference in wages, in the performance of duties and hiring criteria because of sex, age, color or marital status"*. Many other countries also have similar laws to prevent unfair actions against groups of individuals who might be historically socially disfavored.

Due to the use of advanced techniques of machine learning, those decision-making processes have become increasingly automated. Decision-makers take the result of classification algorithms to choose the prediction that is likely to give the best or most profitable outcome. Usually, these techniques are algorithms designed to learn patterns from past data to build a model that maximizes the accuracy of a prediction considering a training set. However, machine learning models may introduce bias against certain groups of individuals, which can lead to **disparate impact**, that is, the likelihood of a specific outcome being more significant for one or more groups of individuals, which is generally referred to as discrimination, or simply *bias*.

Many new applications for Machine Learning and Artificial Intelligence introduce more challenges that impact society in some degree. Credit Approval, Housing, Face and even Voice Recognition to name a few. Those applications must take into account many features that may identify groups of individuals according to their race, sex, or age. Therefore, the problem of bias arises when decision-makers, who are responsible for determining credit approval, job assignment, and insurance, among others, need to maximize the quality of their decision (i.e., denying credit for possible bad payers, hiring suitable job applicants). Several studies suggest the presence of bias [2] in those services. Even though bias may not be directly targeted in those studies, they provide evidence that it can still be measured according to a bias metric.

ProPublica [1] is a nonprofit newsroom of investigative journalism. It has conducted many studies involving automated decisions and their social impact. Some of their studies have become quite notorious, as it is the case of [3] and [2]. Notably, the study [3] regarding how a machine learning algorithm called COMPAS (Correctional Offender Management Profiling for Alternative Sanctions) that predicts the probability of criminal recidivism can be biased against black defendants. Not limited to this, it also discusses possible social impacts of using this algorithm to predict future criminals. On [22], they discuss how they analyzed the COMPAS algorithm, thus, providing statistical clearance on their findings. Those studies motivate our work and also open room for discussion regarding this research area.

There are relevant studies that elucidate many aspects of discrimination-aware learning and also give proper motivation and discussion regarding this field of study. We describe them next.

1.1 Discrimination-Aware Learning

Discrimination-aware learning, or *Fair ML*, is a topic of research that aims at minimizing the impact of bias against certain groups due to ethical reasons and legal implications. Given a labeled dataset whose records represent individuals, let its attributes be divided into non-protected and protected attributes, such as race and gender. The problem consists of building a classifier that takes the non-protected attributes of an individual and maps them to a class label so that the classifier has minimum bias and maximum accuracy. In other words, given two metrics, a bias metric and a quality metric, a learner aims at providing the best trade-off between these two metrics.

The protected attribute usually maps two groups: A **favored group** that is historically favored in most of the automated predictions and, on the opposite side, a **deprived group**. Those features are usually task dependent, for instance, medical inference must take the feature sex as non-protected attribute and education level as protected attribute. Also in most of classification task there are two classes of interest, a **positive outcome** (i.e approve credit) for a individual and a **negative outcome** (i.e deny credit). A Fair ML model aims at providing fair distribution of positive and negative outcomes among the groups.

[1] https://www.propublica.org/

1.1.1 Techniques

A fair ML technique is often considered as a method to produce a fair classification, mitigate the resulting bias of classification, or tune the result of classification concerning a bias metric.

Several recent works tackle the issue of developing bias mitigation techniques. For example, the tutorial [15] on fairness in machine learning that enumerates several techniques in the literature and makes an extensive study of different approaches to achieve fairness in machine learning. We may also notice an increasing effort in building classifiers that are as bias-free as possible. Most of the works adopt real-world datasets as a ground-truth for validating their models. More specifically, they often assess the techniques through cross-validation based experiments on several datasets and employ quality and bias metrics to analyze how well their techniques perform.

1.1.2 Assessment

It is of our interest to assess the quality of fair ML techniques. Their quality can be assessed when it is taken into account classical classification metrics such as accuracy, balanced accuracy, and f-measure, among others; and metrics defined explicitly for this field.

However, it is interesting to note that there is no consensus on the best technique for a given application scenario since it also depends on the level of bias that is inherent to the dataset being targeted. The recent study by [21] discusses the impossibility of a classifier to satisfy multiple notions of fairness. This impossibility is one of the reasons why researchers have put their efforts on minimizing the effect of biased data on the predictions, thereby reducing bias with minimal loss of accuracy. We argue that an effective strategy to assess fair ML models is by running them on scenarios where the bias-related parameters differ so that we can observe how well the models behave.

1.2 Motivation

As it has been stated in a previous section, automated decisions may introduce bias in the resulting learning model. It may also not be clear how the patterns learned from those models can lead to unfair decisions since, in most cases, the protected attributes are not present in the learning process.

A survey from Bonchi et al. [15] wraps up recent works focused on fair ML. It describes bias metrics, techniques and also lists several possible gaps in the field. We are particularly interested in one of these gaps: the fact that there is little consensus on benchmarking for fair ML techniques. That is, since it is hard to define which technique is the most appropriate for each scenario, we should employ a methodology for assessing them according to comparison metrics and thus be able to choose one of them according to a particular need.

Having many metrics and techniques to tackle on the problem of fair ML represent a great progress in the field of research. However, researches may find themselves confused on which technique to use. Recent studies create and discuss these metrics and techniques on a regular basis [7, 11, 18], therefore, a researcher, decision or even a policy makers must have a set o tools to determine the quality of a fair ML technique. There have been survey attempts [10, 29, 41, 26] and even tools to compare techniques [4].

We argue that a researcher or decision-maker may find difficulties selecting appropriate technique for their machine learning models. Also, relying on a technique that is better at minimizing a given bias metric may not achieve the researcher or decision maker's goals depending on their task. For instance, minimizing a bias metric such as statistical parity difference [11] may not cover cases where another metric such as equalized odds [18] is more important. Nonetheless, we need to build a categorical methodology to evaluate fair ML models and also provide means to compare them to each other. It is of our intent having such methodology so researchers could improve their efficiency on developing new techniques, decision-makers could have thorough results so they can select proper approaches and policy makers could make use of such results to instigate discussion in many other fields beyond Artificial Intelligence such as Ethics, Sociology and Law.

In other words, we need to build a benchmark framework that assists researches and decision makers in process of selecting proper techniques for bias mitigation or discovery tasks in machine learning. This benchmark must provide insightful results where it becomes easier for its users to perceive which bias mitigation techniques and bias metrics are more suitable for their tasks.

1.3 Contributions

The main contribution of this work is a framework for comparing fair ML models – the first such benchmark framework. Our framework comprises systematically generated biased datasets that are sampled from Bayesian networks learned from real-world data. Our primary concern is to explore alternative discrimination scenarios; that is, we want to learn from data representing different levels of bias to analyze the behavior of techniques when bias levels differ. Our work focuses on three metrics for assessing discrimination: statistical parity difference, overestimation, and underestimation. However, our approach may be extended to other metrics.

Going beyond traditional results, we provide an alternative way to compare techniques by using a result representation methodology that summarizes the result while keeping most of the information. This abstraction aims at making comparisons between techniques uncomplicated and delivers a straightforward intuition regarding a technique's behavior. In that sense, our secondary goal is to augment the productivity of researchers during the task of technique comparisons. Our result representation methodology seeks to accomplish that by providing visual cues that expose subtle information about the data.

This thesis has been partially published in [9]. It is important to notice that it is not our intention to survey techniques in the literature. Our goal is to demonstrate use cases of our benchmark, thus, comparing techniques acts as an example of benchmark usage.

Chapter 2

Related Work

Machine learning models for measuring and mitigating bias is considered a concern by researchers and policymakers; several studies on fair ML have been conducted in the past decade. Before this work, an extensive list of works focused on those matters has been studied to foresee possible gaps regarding this issue.

Most of the studies in this area can be divided into two groups [15]: *bias discovery*, which focuses on studying metrics and identifying how does bias manifest in a dataset, and *bias mitigation*, which focuses on building classification models that are less likely to produce biased results.

2.1 Bias Discovery

For *bias discovery* there are a number of works that aim at identifying patterns of bias in data [32, 36], including metrics to quantify the amount of bias. [41] survey several such metrics; she defines the *mean difference* as the most commonly used metric in early works, also called *slifd* [33]. [38] propose a metric called *disparate mistreatment*, which measures the difference between misclassifications.

2.2 Bias Mitigation

Bias mitigation methods can be divided into three groups: *pre-processing*, *in-processing* and *post-processing* techniques [15]. We focus more on *Pre-processing techniques* [19, 13], since they modify the training set in order to make it as bias-free as possible, so that a classifier becomes less prone to exhibit bias. *In-processing techniques* [7, 20, 38] work by changing the classifier to produce less biased models, whereas *Post-*

processing techniques [16] change the outcome of classification models.

2.2.1 Pre-Processing

Pre-processing techniques include methodologies that mitigate the bias from the target dataset, thus allowing it to be used as input for traditional machine learning models and hopefully exhibit less bias on the classification than the original dataset without harming much of the accuracy.

Calder's Techniques [7] propose several pre-processing approaches for dealing with the problem of fair ML. We use four of those approaches as they are similar to each other. They are data pre-processing techniques, which means that they take the training data as input and produce a new dataset. This new dataset is used for conventional learning, and it is expected that the resulting classifier provides less bias than the classifier built using the original data.

- Massaging changes the class label of individuals to balance positive outcomes between groups. Individuals of the deprived group from the negative class are reassigned to the positive class, and individuals of the favored group having a positive class are reassigned to the negative class. Instances are selected for class reassignment based on a score learned by a ranker.

- Re-weighting assigns higher weights to individuals of the deprived group that have a positive class label and to individuals of the favored group that have a negative class label.

- Uniform Sampling applies the following rule on a randomly chosen instance: if the instance is from the deprived group with negative class, it is removed; otherwise, if it is from the positive class, it is duplicated. Likewise, if the instance is of the favored group with a positive class is removed, but if it has a negative class, it is duplicated.

- Preferential Sampling chooses instances based on a ranker like in the Massaging technique. The change rules are the same as in uniform sampling. Individuals close to the ranker's decision boundary are likely to be selected for the Preferential Sampling process.

Black Box Auditing (Auditor)

Black Box Auditing [1] is an implementation of Gradient Feature Auditing (GFA) introduced in [1]. This technique works by repairing the dataset via a pre-processing technique, which means that it changes attribute labels by modifying the dataset according to a criteria defined by the technique's developer(s). The resulting repaired dataset is expected to have less bias. We run the data repairer described by [13] at the different repair levels.

2.2.2 In-Processing

This type of technique now focuses on building a machine learning model in which a metric of bias determines one of its constraints. That way, this learner aims to optimize both a quality metric (i.e., accuracy) and a metric of bias (i.e., mean difference, disparate impact).

Fair Classification Learner

We also consider the implementations from [38, 39], where they design a fair classifier that works by modifying the decision boundary of a classifier. It can maximize fairness under accuracy constraints and maximize accuracy under fairness constraints. It is also possible to add a constraint on misclassification of a positive outcome, which means that it only changes the label for the protected group from unfavorable to favorable class and does not change the label of non-protected groups to the negative outcome. This technique will be used for the bias experiments (Accuracy vs. Bias). In [38], they introduced the notion of disparate mistreatment and proposed a methodology that aims at reducing false positive and false negative rates. The technique consists of training classifiers in such a way that their decision boundaries are modified to avoid those misclassification rates. We can adjust which constraints we want to use: correcting false positive, false negatives, or both. This technique will be used for the disparate mistreatment experiments (Overestimation vs. Underestimation).

Adversarial De-biasing

This technique was introduced in [40]. It uses adversarial learning to produce a classification method that minimizes the bias. In order to achieve that, the authors learn a predictor for the class Y (i.e., income class) and an adversarial that predicts the protected attribute Z (i.e., gender). The predictor should maximize the accuracy of the prediction of the label Y while the adversarial tries to minimize the prediction of the protected variable Z.

[1] https://github.com/algofairness/BlackBoxAuditing

2.2.3 Post-Processing

These techniques usually work by changing prediction labels from already complete traditional machine learning models. The labels are changed to address a specific metric of bias. Post-processing techniques, being less common, though, provide a wrapped-up technique that does not necessarily have to deal with the specifications of the training process as it is stated on [25].

On [18], the researchers propose a post-processing technique that calibrates a predictor and removes the resulting bias for its classification. They employ their techniques on FICO credit score [2], and they have achieved convincing results, so we believe that it would be interesting to study this approach on our work.

2.3 Data Generation

Our work generates artificial data for benchmarking fair ML techniques. Synthetic data generation has been used in the past with a similar goal [18, 42]. However, they create simple models with a few variables and their conditional dependencies, containing protected and non-protected artificial attributes and a binary class. In contrast, we generate biased data sampled from Bayesian networks learned from real-world data, instead of purely synthetic data.

Some works use Bayesian networks to deal with bias discovery and prevention tasks. [6] addresses the problem of learning probabilistic causal structures of bias from datasets. [27] use Bayesian networks for the task of bias prevention. [24] alters Bayesian network structures so it can satisfy Fair Equality of Opportunity. Our work learns these structures, artificially produces bias patterns, and uses them to sample data at different levels of bias. While previous approaches focused on discovering and mitigating bias, we use sampled data from Bayesian networks to evaluate the robustness of bias prevention techniques.

[2] https://www.fico.com/br/products/fico-score

2.4 AIF360

AIF360 [4] is an open-source toolkit that includes some metrics and techniques to measure bias and mitigate it from learning models. It helps researches to asses the quality of their algorithms on the datasets provided by them. It is an effort of various researches and is released as a Python package.

2.4.1 Metrics

One might want just measure the resulting bias of their models according to a specific metric. So it should be possible to perform no technique at all and consider only the discovery of bias (i.e., a policymaker). AIF360 currently supports Group fairness, Distortion Metrics and Generalized Entropy Index [37].

2.4.2 Techniques

Several techniques are available in AIF360 and they represent a joint effort of many researchers and become more accessible due to this toolkit. At the moment of this publication, AIF360 supports techniques such as:

- Optimized Preprocessing [8];
- Disparate Impact Remover [13];
- Equalized Odds Postprocessing [18];
- Reweighing [20].

2.4.3 Benchmark Framework and AIF360

Since this work and AIF360 have similar objectives, it might be handy to couple them in such a way that we can make use of the diversity of metrics and techniques along with an arbitrary number of biased datasets. That way, we can perceive the robustness of each technique in a large number of scenarios depending on a metric.

Since it is important to keep the generality, it produced an abstraction to create custom datasets to fit the AIF360 models. This way, it is possible for practitioners to extend this methodology to their dataset as they please. By doing that, it can be assured that our work can be tested in several different scenarios and also provide ways to simplify its use.

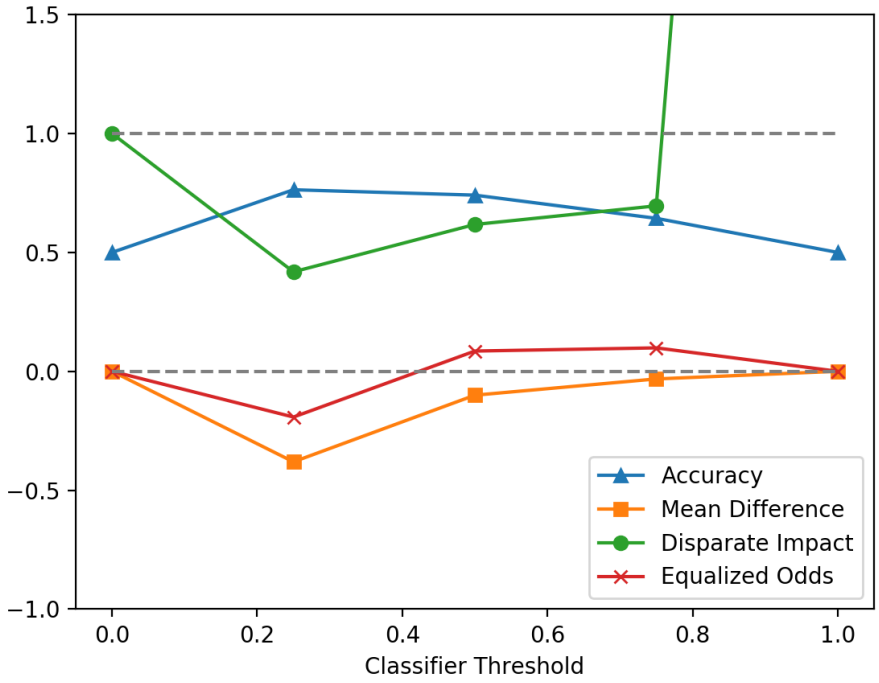
Flowchart 2.7 depicts the integration of our methodology on the AIF360 toolkit. By producing datasets representing different scenarios of bias and splitting data into train and test, we input the fair ML model provided by the toolkit and select the desired technique and proper parameters. Then, the results are interpreted and validated with metrics of quality and bias.

2.4.3.1 Sample Results

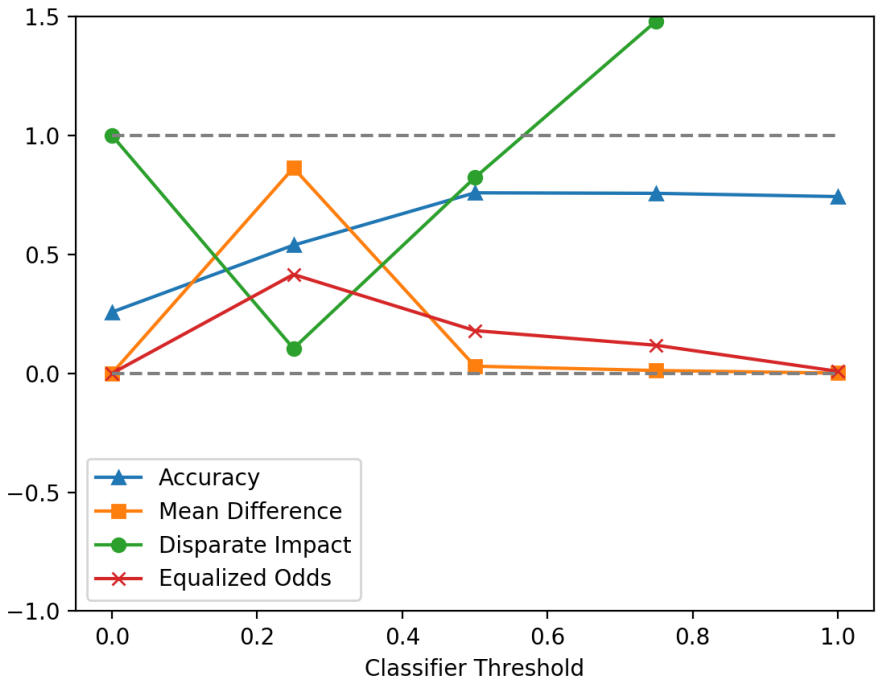
Here are sample results for a demonstration of the use of our methodology along with the AIF360 toolkit. We tested two types of datasets, Adult census, and COMPAS, on each one of the three ways of bias introduction, depicted as A, B and C respectively with section 4, using two techniques, Re-weighting [20] and Odds Equalizer [18]. The bias level of each bias generating process was set to 0.5, and we measured four different metrics: Accuracy, Mean Difference, Disparate Impact, and Equalized Odds, and following AIF360 tutorial, they were given by a classification threshold for logistic regression.

2.5 Discussion

A potential gap one may encounter regarding fair ML research field is the dataset coverage and how well can they certify the robustness of a technique. That means we still don't have many studies on how techniques behave throughout different bias metrics.

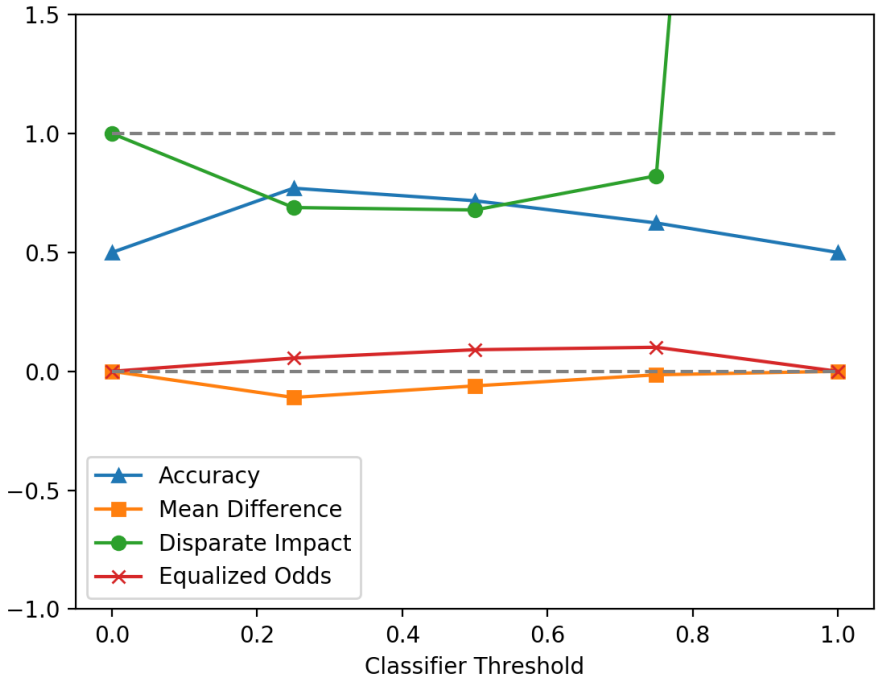


(a) Re-weighting A

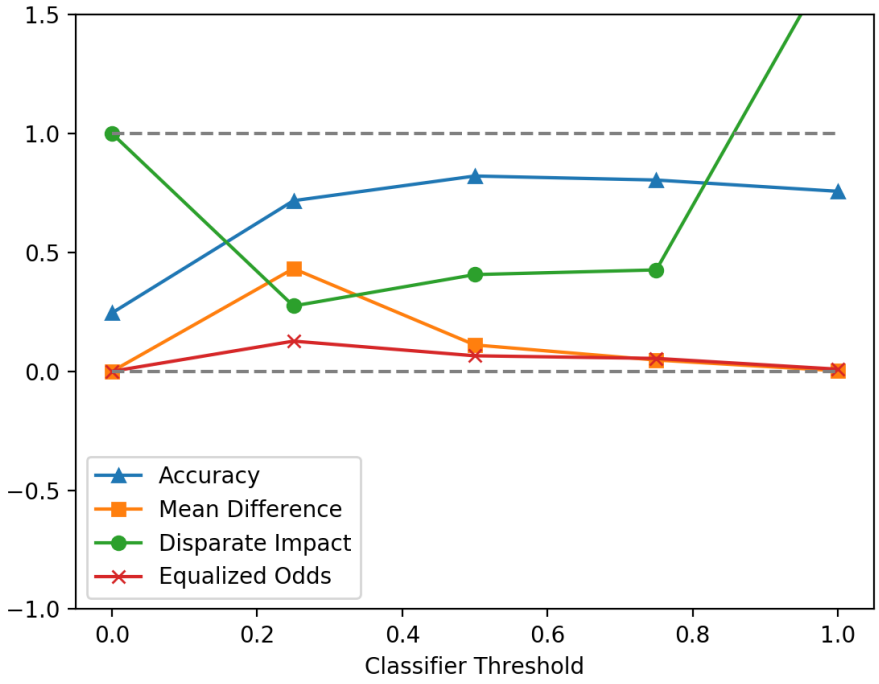


(b) Odds Equalizer A

Figure 2.1: Re-weighting (a) and Odds Equalizer (b) for data generation type 'A' on adult census dataset.

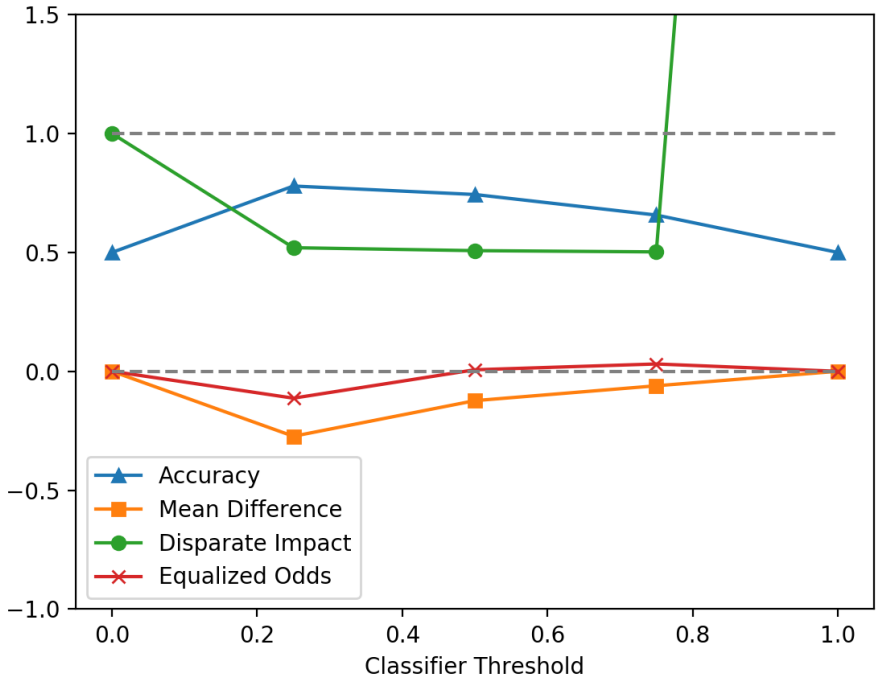


(a) Re-weighting B

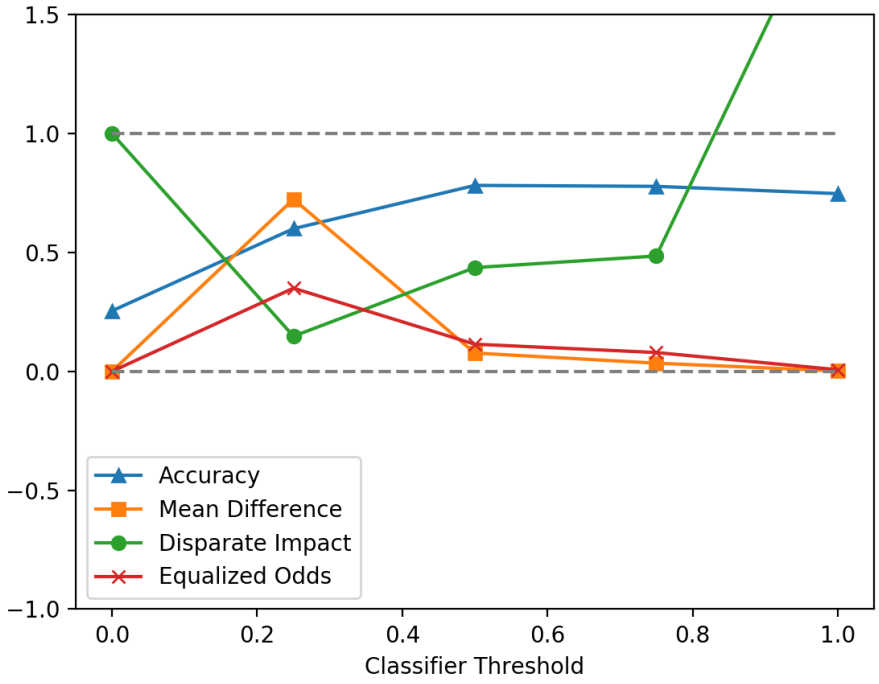


(b) Odds Equalizer B

Figure 2.2: Re-weighting (a) and Odds Equalizer (b) for data generation type 'B' on adult census dataset.

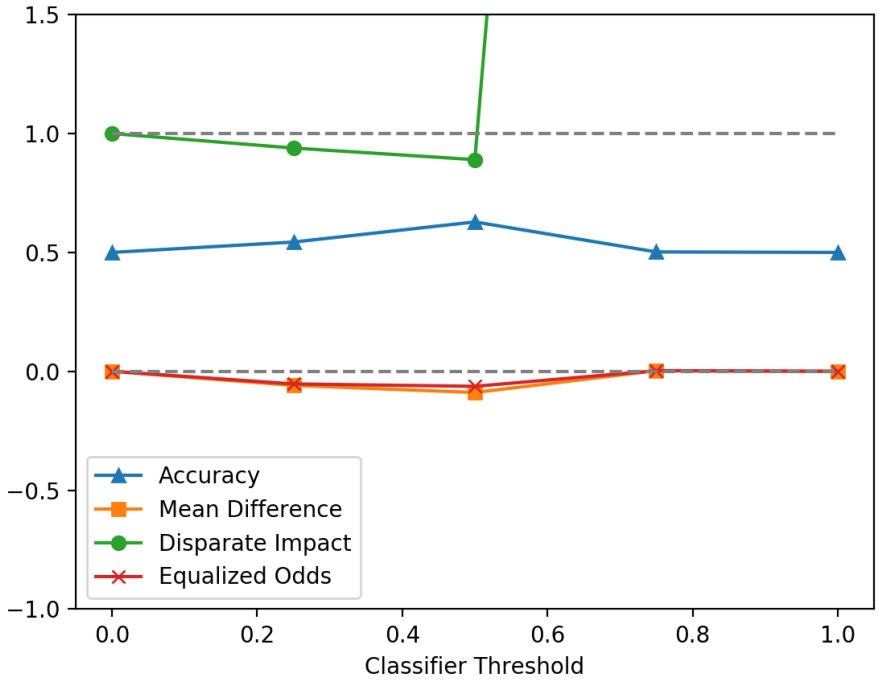


(a) Re-weighting C

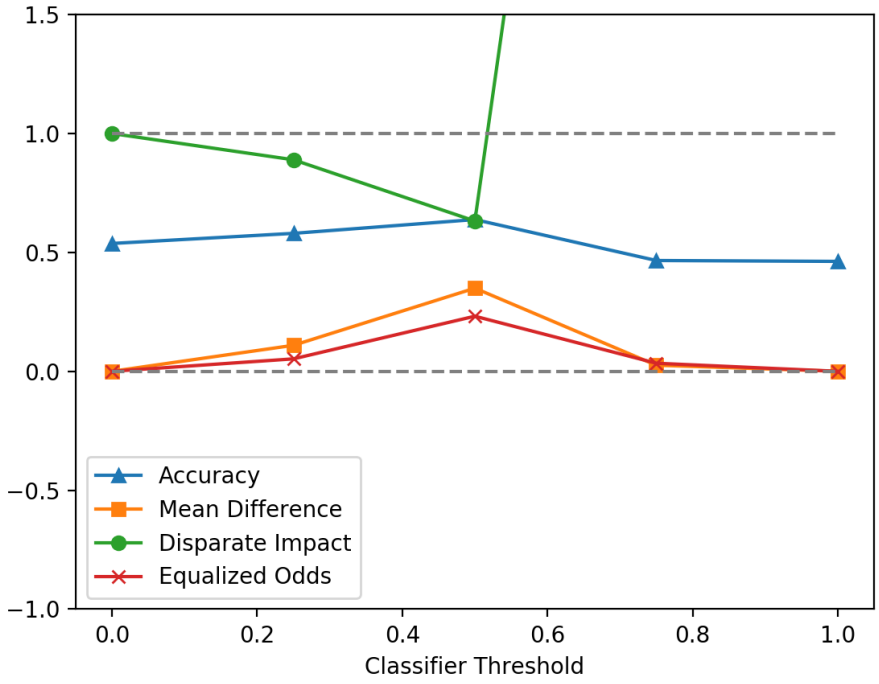


(b) Odds Equalizer C

Figure 2.3: Re-weighting (a) and Odds Equalizer (b) for data generation type 'C' on adult census dataset.

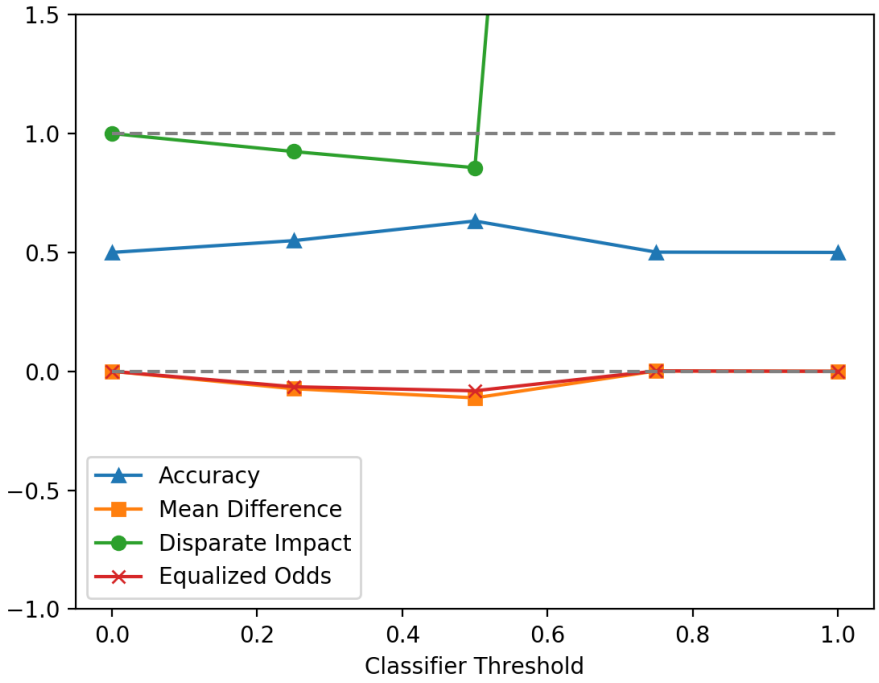


(a) Re-weighting A

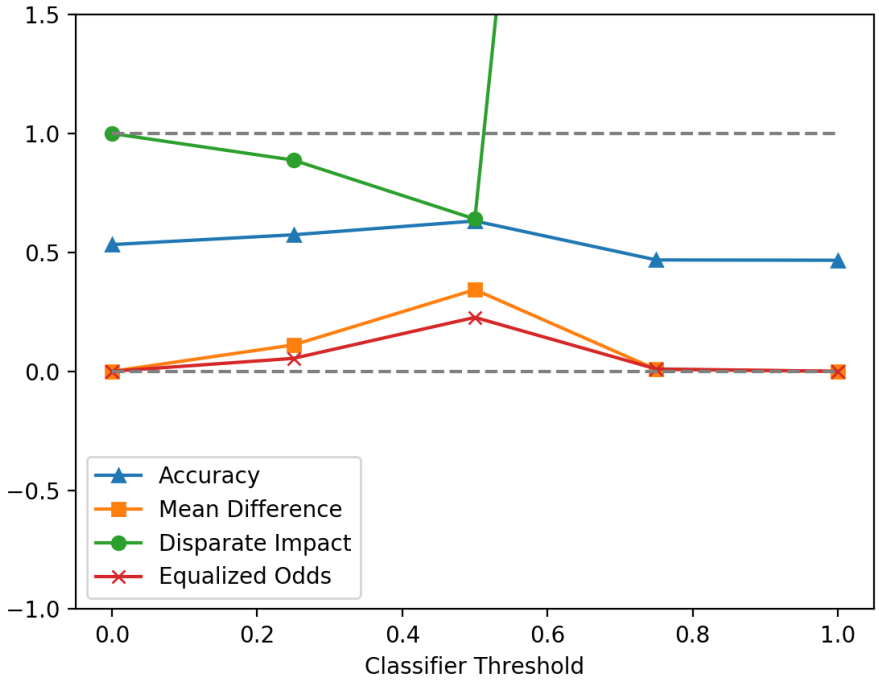


(b) Odds Equalizer A

Figure 2.4: Re-weighting (a) and Odds Equalizer (b) for data generation type 'A' on COMPAS dataset.

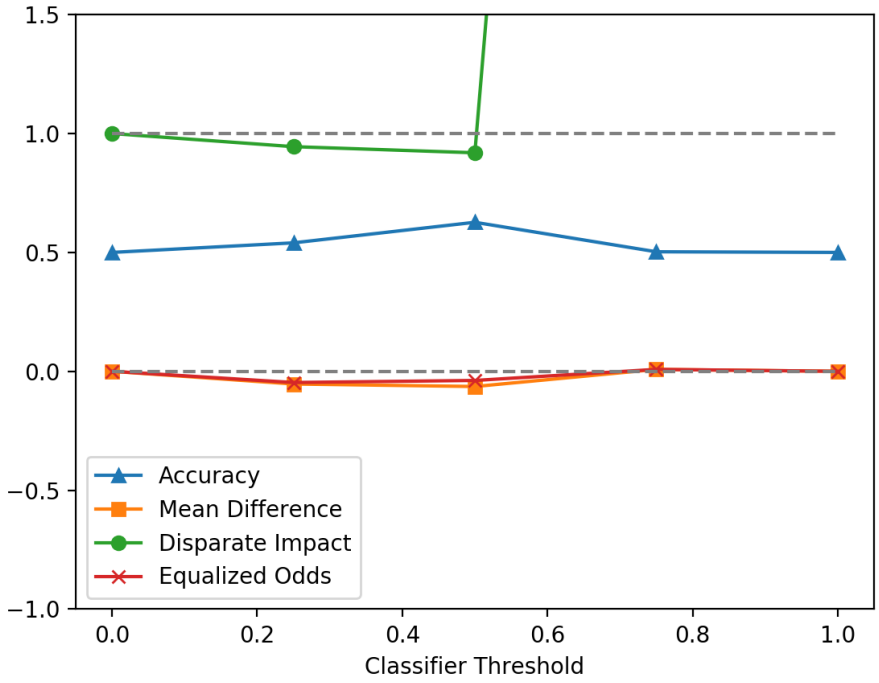


(a) Re-weighting B

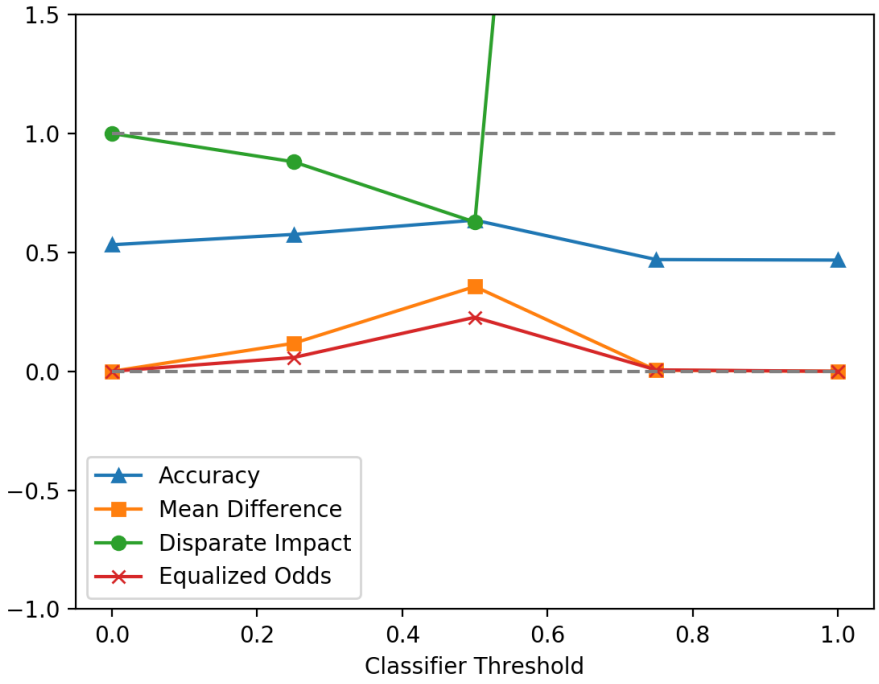


(b) Odds Equalizer B

Figure 2.5: Re-weighting (a) and Odds Equalizer (b) for data generation type 'B' on COMPAS dataset.



(a) Re-weighting C



(b) Odds Equalizer C

Figure 2.6: Re-weighting (a) and Odds Equalizer (b) for data generation type 'C' on COMPAS dataset.

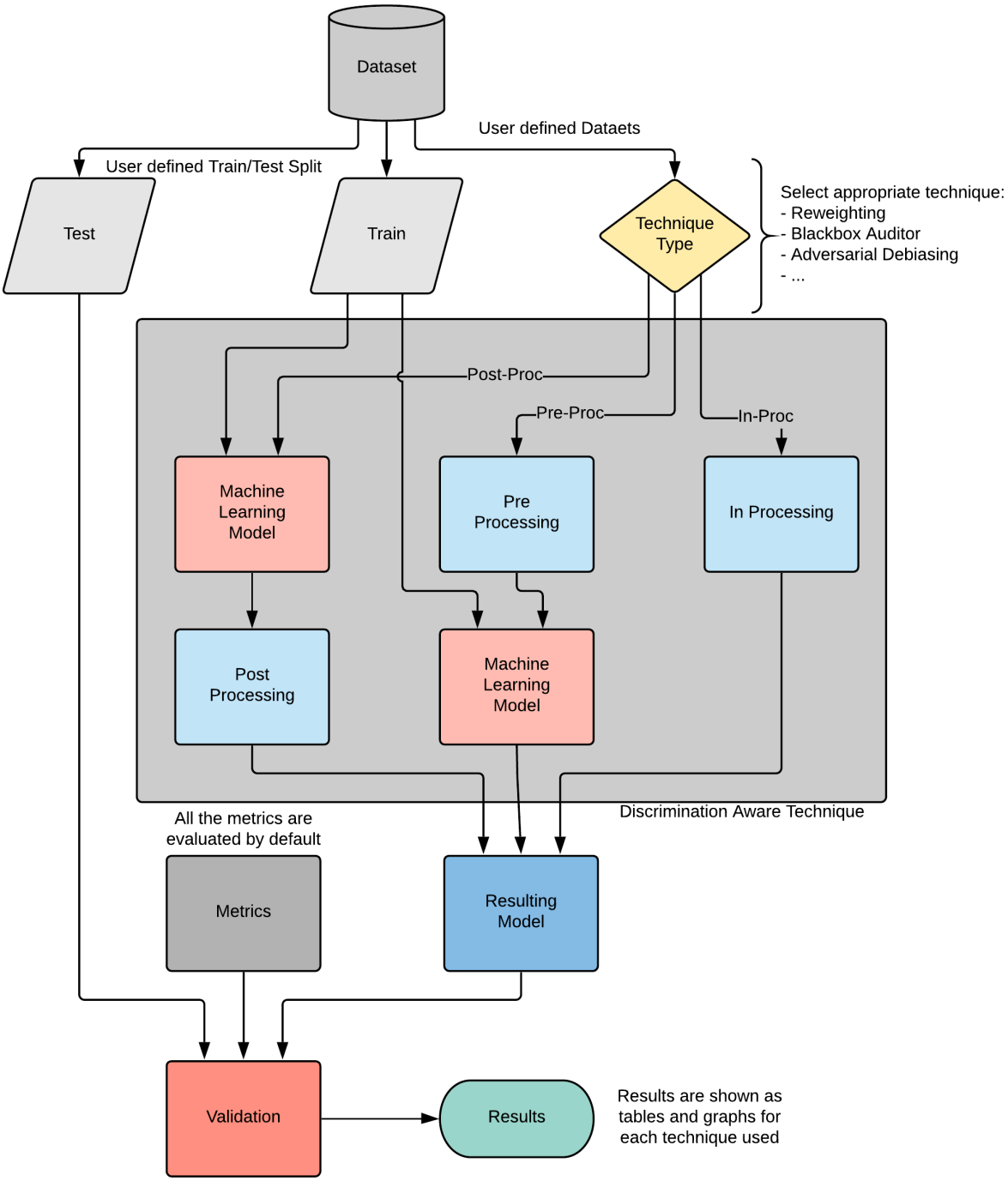


Figure 2.7: Flowchart representing the integration of a technique into our framework. A dataset, in that case, the artificial data generated from our methodology, is used as input for the framework. A train-test split is done, and a technique is selected. The main gray area selects the proper treatment for the technique type, and a Learning Model is the output. The validation is performed according to selected metrics.

It is of our interest to fill this gap, or at least establish the means to, by providing a benchmark framework that offers an approach to dataset completeness by modifying real world data.

As it has been stated in [15] there is no consensus on Benchmark for fair ML methodology, thus, making the assessment of those techniques less convincing. Such a benchmark should consist of a systematic approach of measuring and comparing not only two techniques but also two metrics as well.

Another work from [14] aims at comparing two techniques, but it is somewhat limited because it only accounts for the real-world dataset for its measures while we generate artificial data to simulate specific unfair situations. Also, it is fundamentally different from ours since our main contribution is the generation of simulated data for benchmarking techniques. To the best of our knowledge, no other methodology has been proposed to deal with data scarcity and artificial data generation.

Chapter 3

Definitions

A fair ML model usually exploits the trade-off between a quality and a bias metric. The task is generally framed as a binary classification problem, where not only the accuracy should be maximized, but also the bias metric has to be minimized. The classification score is defined according to the probabilities of outcomes for different groups of individuals. These groups are sets of individuals who share the same value of a specific attribute and, by contrasting their outcomes, we assess the model bias.

3.1 Protected and Non-Protected Attributes

Formally, we are given a labeled dataset D, where each record represents an individual. For each individual there are two sets of attributes $X = \{x_1, ..., x_n\}$ and $S = \{s_1, ..., s_m\}$. We call the set X the legally allowed attributes, or *non-protected attributes*, which, in theory, may be used in decision making without any legal implications, e.g., annual income. We call the set S the non-legally allowed attributes or *protected attributes*. We are not supposed to use any attribute from S in a decision making process because they are protected by law, for instance, race, sex. Those attributes are assumed as protected, however it may depend on the problem in discussion. Classical Fairness in Machine Learning problems such as Credit Approval and Risk Assessment usually adopt race and gender as the protected attribute. On the other hand we may face circumstances such as calculating the risk of transferring a patient to Intensive Care Unit, where socioeconomic status, although commonly related to many other protected attributes, should not interfere.

3.2 Class Labels

The class label $y \in \{+, -\}$ is the variable the model tries to predict for each individual. A positive class label $y = +$ (also denoted y^+) expresses a favorable outcome for the individual; alternatively, a negative class label $y = -$ (also denoted y^-) expresses an unfavorable outcome. An example is in credit scoring, where a bank wants to decide whether a customer has a good credit score or not. A good credit score is represented by $y = +$, while a bad credit score is represented by $y = -$.

3.3 Bias Metric

Let s be a protected attribute taking on two values $\{g, \bar{g}\}$. Here, the assumptions is that individuals belonging to g (that is $s = g$) who usually suffer from discriminatory conditions, and are called the **deprived group**, and individuals not belonging to g (that is, $s \neq g$ or $s = \bar{g}$) who usually have an unfair advantage in models, and they are called the **favored group**.

3.3.1 Single Attribute

In order to measure the bias concerning a protected attribute s for individuals who belong to a group g, we quantify the difference between probabilities of positive outcomes for individuals who belong to g, and those who do not belong to g, as follows:

Statistical parity difference: This metric compares the difference between predicted positive outcome $\hat{y}^+$ for individuals of different groups:

Definition 1 *Let $\hat{y} \in \{+, -\}$ be the predicted labels and $s \in \{0, 1\}$ be the protected attribute that defines individuals belonging to the favored group $s = 0$ or deprived group $s = 1$. The statistical parity difference spd_s with respect to the attribute s is defined by the following equation:*

$$spd_s = p(\hat{y}^+ \mid s = 0) - p(\hat{y}^+ \mid s = 1) \tag{3.1}$$

That is, the difference between the probability of positive class for the favored ($\bar{g}$) versus the disfavored group (g). For example, members of the favored group may have a higher probability of a good credit score. This definition is widely used for measuring bias [19, 16] and it is often referred to as **mean difference**.

This means that it is desirable to have values of $bias_{D,s,g}$ closer to 0, meaning that the likelihood of a positive outcome is similar for different groups. On the other hand, values of $bias_{D,s,g}$ closer to 1 mean that we discriminate against individuals of a group g and value of $bias_{D,s,g}$ closer to -1 means that we are favoring individuals in this group.

This definition can be extended to multiple groups. A bias value is measured from a group to another, which means that Definition 1 can be generalized as follows:

Definition 2 *Let D be a dataset with labeled binary classes $y \in \{+, -\}$, a protected attribute s, an attribute value g_1 that defines individuals belonging to a group and g_2 that defines individuals belonging to another group. The bias $bias_{D,s,g_1,g_2}$ in D with respect to the attribute s for individuals between groups g_1 and g_2 is defined by the following equation.*

$$bias_{D,s,g_1,g_2} = P(y^+|s = g_1) - P(y^+|s = g_2) \tag{3.2}$$

A practical example of this definition would be the case of racial discrimination into three groups: Caucasian, Black, and Hispanics. Most of the racial models only consider bias between Caucasian and Black, or Caucasian and non-Caucasian, but bias can be measured between two subsets of groups, that is, bias between the subset composed only of specific groups and the subset composed of the union of two different groups.

3.3.2 Multiple Attributes

We can also measure bias considering multiple attributes. For example, one wants to measure the bias between Caucasian Men and Hispanic Women in a dataset. In this case, we have two protected attributes: gender and race. We argue that we should not represent multivariate bias into a single value for the following reasons. First, it is easier to quantify bias separately. Second, we do not know which attribute is more important for a bias measure.

Although we can extend our work to multiple groups and multiple protected attributes, we will focus mainly on single groups and single protected attributes because the current techniques for fair ML were developed to deal only with those cases.

3.3.3 Classification Metrics

Definition 3 *Let $\hat{y} \in \{+, -\}$ be the predicted labels, $y \in \{+, -\}$ be the true labels and $s \in \{0, 1\}$ be the protected attribute that defines individuals belonging to the favored group $s = 0$ or deprived group $s = 1$. The overestimation $over_s$ and underestimation $under_s$ with respect to the attribute s are defined by the following equations, respectively:*

$$over_s = p(\hat{y}^+ \mid s = 0, y^-) - p(\hat{y}^+ \mid s = 1, y^-) \tag{3.3}$$

$$under_s = p(\hat{y}^- \mid s = 1, y^+) - p(\hat{y}^- \mid s = 0, y^+) \tag{3.4}$$

where $p(\hat{y}^+ \mid s = 0, y^-)$ denotes the probability of predicting the positive class, when the true class is negative, for the favored group; the other terms are defined similarly.

The favored group $\bar{g}$ may have a higher false-positive rate (called **overestimation**) and a lower false negative rate (called **underestimation**) compared to the disfavored group g. These metrics seek to measure the extent of this problem for a given classifier f.

The advantage of this definition is that it assures that the classifier has caused different error rates between different groups of individuals. This definition would help in scenarios such as the ProPublica study, which pointed out unbalanced error rates of the COMPAS Recidivism Algorithm [1], where African American people are more likely to be assigned a higher risk of re-offending in a crime without actually re-offending. Equation 3.3 is referred to as **Overestimation of the Favored Group**, or simply **Overestimation** because its right-hand side represents the difference in terms of false positives between favored and deprived groups. In a biased situation, the favored group is more likely to be predicted with a positive outcome; for that reason, Equation 3.3 measures the fraction of the favored group that is being wrongly classified as positive minus the fraction of the deprived group in the same situation.

Alternatively, we call Equation 3.4 **Underestimation of the Deprived Group**, or simply **Underestimation**, because it gives the opposite context: biased classification often predicts that the deprived groups have a negative outcome, thus making the deprived group more susceptible to being wrongly classified as negative class. Equation 3.4 measures the difference of wrongly classified as unfavorable between deprived and favored groups. It is important to note that bias is inherent bias in the dataset, since it measures the difference between the favored and deprived groups with respect to the observed class. On the other hand, disparate mistreatment evaluates the bias of the classifier f.

[1] https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing/

Nevertheless, we can still replace y in Equation 3.1 with $\hat{y}$ and measure the bias of a classification execution.

3.3.4 Additional Metrics

We are not constrained to the metrics listed , and other definitions of bias metrics are also commonly used in fair ML. The framework we propose in this work is extensible to these different types of metrics such as equalized odds [18] and statistical parity [11].

Another example, on many aspects of the legal system, there may be definitions for metrics of bias. A work from [34] enumerates some of them, such as risk difference and ratio, selection rate, and the previously mentioned odds ratio. Notice that there may be overlapping of metrics names since many of them are used on legal instances, therefore, presenting diverse semantics.

Chapter 4

Benchmark Methodology

In this section we describe our methodology, which is depicted in Figure 4.1 and has two main inputs: (i) original data, which is a dataset containing both protected and non-protected attributes; and (ii) one or more fair-ML models that we want to assess w.r.t. fairness capabilities. Our framework comprises three main components: (i) generative model; (ii) experimental evaluation; and (iii) summary generation. Our process starts with building a generative model that captures the characteristics of the original dataset. We use this model to generate bias-variant data, upon which the fair-ML model is instantiated. The results are then summarized. In the remaining of this section we describe each component in more detail.

4.1 Bayesian Networks

Bayesian Networks define a conceptual representation of distribution of probabilities as a Graphical Model [30]. This representation consists of a Directed Acyclic Graph where its nodes represent attributes and the edges represent dependencies between the nodes. These dependencies describe conditional probabilities of nodes with their parents.

Such abstraction becomes really useful causality inference. Therefore, this concept becomes appropriate to apply on fair ML because one of the core issues of this field of research is to detect the influence or causality of protected attributes in the outcome.

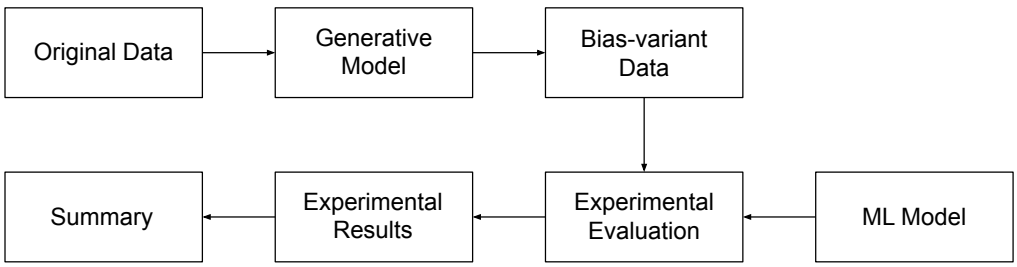


Figure 4.1: Methodology

4.2 Estimating Bayesian Networks

A Bayesian network is a probabilistic graphical model that maps conditional dependencies of random variables into a directed acyclic graph. We use an estimated Bayesian network to generate synthetic data, which allows us to quantify how much influence a protected attribute has on the outcome of a classification model.

To learn the structure of a Bayesian network in order to study causal relationships between the variables, we use the popular `R` library `bnlearn` [28]. Knowing the Bayesian network that represents a given data allows us to reproduce the characteristics inherent to the original data and also to adjust specific parameters to generate diverse scenarios. By learning these structures, we can modify any node's conditional probabilities and are thus able to calibrate the bias of an outcome concerning a protected attribute. Currently, we employ two methods for learning Bayesian Networks, Hill-Climbing greedy search, and Tabu Search algorithm, both described in [28].

4.3 Overview

An algorithmic overview of our framework dynamics is summarized in Algorithm 1. The process starts by building a generative model $\mathcal{M}$ derived from real-world data $\mathcal{D}$ (lines 1-2). Function LearnGM($\mathcal{D}_t$) takes a split from the real world data as input and returns a generative model. Our methodology works for any generative model, and we chose Bayesian networks [5], which are learned through a Hill-Climbing greedy search[28].

This learned model $\mathcal{M}$ is capable of generating synthetic data $\mathcal{D}^{'}$ in such way that their conditional probabilities between features is similar to $\mathcal{D}$. Then, $\mathcal{M}$ is modified according to a value $\beta : \{\beta \in \mathbb{R} \mid 0 \leq \beta \leq 1\}$ which specifies the degree of change applied to it, also named here as **artificial bias level**, or simply bias level. The bias level β will dictate conditional probabilities between the protected attribute s and class label y. It is important to point out that our Bayesian network learner enforces a connection from the protected attribute s to the class label y [31]. It does not necessarily creates a strong dependence between s and y, it only makes the Bayesian network more manageable.

We may then iterate β from 0 (no modification) to 1 (maximum modification), and, in each iteration, we create a modified model $\mathcal{M}^*$ (lines 4-5). This modified model $\mathcal{M}^*$ is used to generate synthetic data $\mathcal{D}_s$ (line 6), which will be given as input to a bias-mitigation technique $\mathcal{T}$ as follows:

Algorithm 1 Benchmark Methodology

Require: $\mathcal{D}, \mathcal{T}, s, y$
1: $\mathcal{D}_t, \mathcal{D}_v \leftarrow$ Split($\mathcal{D}$) ▷ t:training v:validation
2: $\mathcal{M} \leftarrow$ LearnGM($\mathcal{D}_t$)
3: $\mathcal{R} \leftarrow \{\}$ ▷ List of results
4: **for** $\beta \leftarrow 0$ to 1 **do**
5: $\mathcal{M}^* \leftarrow$ ModifyGM($\mathcal{M}, s, y, \beta$) ▷ Modified model $\mathcal{M}^*$
6: $\mathcal{D}_s \leftarrow$ GenerateSynData($\mathcal{M}^*$)
7: **if** $\mathcal{T}$ is pre-proc **then** ▷ Pre-processing techniques modify data
8: $\mathcal{D}_s{}' \leftarrow \mathcal{T}(\mathcal{D}_s)$
9: $r_i \leftarrow$ Classifier($\mathcal{L}, \mathcal{D}_s{}', \mathcal{D}_v$)
10: **else if** $\mathcal{T}$ is in-proc **then** ▷ In-processing techniques use their own classifier
11: $r_i \leftarrow \mathcal{T}(\mathcal{D}_s, \mathcal{D}_v)$
12: **end if**
13: $\mathcal{R} \leftarrow \mathcal{R} \cup r_i$ ▷ Append results into list $\mathcal{R}$
14: $\beta \leftarrow \beta + \epsilon$ ▷ ϵ defines β granularity
15: **end for**
16: **return** $\mathcal{R}$

$$\mathcal{T}: \begin{cases} \mathcal{D}_s \times s \rightarrow \mathcal{D}_s{}', & \text{if } \mathcal{T} \text{ is a pre-processing technique} \\ \mathcal{D}_s \times \mathcal{D}_v \times s \mapsto \mathbb{R}^m, & \text{if } \mathcal{T} \text{ is a in-processing technique} \end{cases}$$

For pre-processing techniques (line 7), technique $\mathcal{T}$ receives the synthetic data $\mathcal{D}_s$ as input and outputs a pre-processed data $\mathcal{D}_s{}'$ (line 8). This pre-processed data $\mathcal{D}_s{}'$ will be the input of a binary classification model (line 9) that uses a traditional classifier $\mathcal{L}$ (in the current implementation we employ J48 decision trees [35]), where it will be validated on a partition $\mathcal{D}_v$ of the original data $\mathcal{D}$. The result of the classification model is an m-tuple $r_i : \{r_i \in \mathbb{R}^m\}$, where m is the number of metrics (quality or fairness) we are computing.

On the other hand, if $\mathcal{T}$ is a in-processing technique (line 10), it will return the m-tuple r_i directly (line 11); since in-processing techniques often employ their own classification model. Handling post-processing techniques will demand the application of a classifier $\mathcal{L}$ to the technique $\mathcal{T}$ and the output would be a post-processed classifier $\mathcal{L}'$, and is left for future work.

Finally, the result r_i is appended into a result list $\mathcal{R}$ (line 13) and the next iteration takes place after incrementing β by $\epsilon : \{\frac{1}{\epsilon} \in \mathbb{N}^+\}$ where ϵ defines the granularity of β increments (in this work we use $\epsilon = 0.05$).

4.4 Generating Bias-variant data

In this section we detail the first component of our framework, which generates synthetic bias-variant data through modifications to the generative model $\mathcal{M}$. As mentioned earlier, we use Bayesian networks as our generative model. Bayesian networks are built as a directed acyclic graph $\mathbb{G}$ along with a conditional probability table P. Algorithm 2 describes the ModifyGM$(\mathcal{M}, s, y, \beta)$ function. Here, the Bayesian network is $\mathcal{M}$, which can also be portrayed as $(\mathbb{G}, P)$; the protected attribute is $s = \{0, 1\}$, class label is $y = \{-, +\}$ and bias level is $\beta : \{\beta \in \mathbb{R} \mid 0 \leq \beta \leq 1\}$. The protected attribute s and class label y are also denoted as nodes in $\mathbb{G}$. A node x may have a set of parent nodes Pa(x).

Algorithm 2 Modifying Generative Model (Bayesian Networks)

Require: $\mathcal{M} = (\mathbb{G}, P), s, y, \beta$
1: $p \leftarrow \mathcal{M}.P$
2: **if** $s \in$ Pa(y) **then** ▷ If s is a parent of y
3: $\quad x \leftarrow$ Pa$(y) \setminus s$
4: $\quad p(y^+ \mid s = 1, x) \leftarrow p(y^+ \mid s = 1, x)(1 - \beta)$
5: $\quad p(y^- \mid s = 1, x) \leftarrow 1 - p(y^+ \mid s = 1, x)$
6: **end if**
7: $\mathcal{M}.P \leftarrow p$

Initially we take the conditional probability table P (line 1). Variable p will be adjusted according to the bias level β. We take all parent nodes from y besides s (line 3) and modify the following conditional probability (line 4):

$$p(y^+ \mid s = 1, x) = p(y^+ \mid s = 1)(1 - \beta) \tag{4.1}$$

Equation 4.1 defines adjustment on the conditional probability of the class label being positive given that $s = 1$ (the individual belongs to the protected group). Naturally, since y is a binary class label, Equation 4.1 has its counterpart (line 5):

$$p(y^- \mid s = 1, x) = 1 - p(y^+ \mid s = 1, x) \tag{4.2}$$

Finally, the conditional probabilities table P from the Bayesian network $\mathcal{M}$ is updated with value p (line 7).

We only change the probabilities for the protected group ($s = 1$) because if we change the conditional probabilities of the privileged group, we would insert twice the amount of artificial bias. Our methodology consists of generating $1/\epsilon$ Bayesian networks for each dataset. Each Bayesian network is generated with a different β, which ranges from 0 to 1. Intuitively, the β represents the degree of change on the conditional probability

of the outcome. For example, if β is 0, the dataset will be generated with its conditional probabilities unmodified. If β is 1, the probability of having a positive outcome given that an individual belongs to the protected group becomes 0.

Table 4.1: Toy conditional probability table

	$s = 0$	$s = 1$
$p(y^- \mid s)$	0.63	$0.80 \rightarrow 0.90$
$p(y^+ \mid s)$	0.37	$0.20 \rightarrow 0.10$

Table 4.1 shows a toy example of conditional probabilities of the outcome given a variable s. Using the fairness metric (statistical parity), we can see that the bias against protected group $\{s = 1\}$ is 0.17 since $p(y^+ \mid s = 0) - p(y^+ \mid s = 1) = 0.37 - 0.20 = 0.17$. Suppose we want to insert a bias level β of 0.5. If we change the probability under the column $\{s = 1\}$, as shown in Table 4.1 (where $0.80 \rightarrow 0.90$ denotes that we change the probability from 0.80 to 0.90; likewise for $0.20 \rightarrow 0.10$), we can observe that the bias against the protected group is 0.27 $(0.37 - 0.10)$, which means that we have increased the resulting bias, therefore, sampling from this Bayesian network will generate artificial data with higher degree of bias.

4.5 Practical Example

We take the **Adult** dataset (also known as **census income** dataset) from UCI Machine Learning Repository [23] in order to illustrate our methodology. This dataset is a demographic census, and it is commonly used to predict whether an individual's income is higher than \$50K per year or not, given a list of attributes that include age, education level, marital status, occupation, and sex.

A data cleansing step is performed by discretizing some of the attributes and removing others. The clean dataset is used as input for the `bnlearn` Hill-Climbing learner.

Figure 4.2 is a visual representation of the generated Bayesian network (BN) using the adult census dataset as input. It can be observed that there is conditional dependency between the variables *relationship* and *income_class* and also there is conditional dependency between *relationship* and *sex*. This evidence supports the fact that these data are biased concerning sex because the relationship describes sex, i.e., a *wife* is a *woman*, and the relationship also describes income class, e.g., being a *wife* may influence income.

For the synthetic data generation process, we have the corresponding BN for the Adult census dataset and its conditional probabilities tables. Each row in this dataset

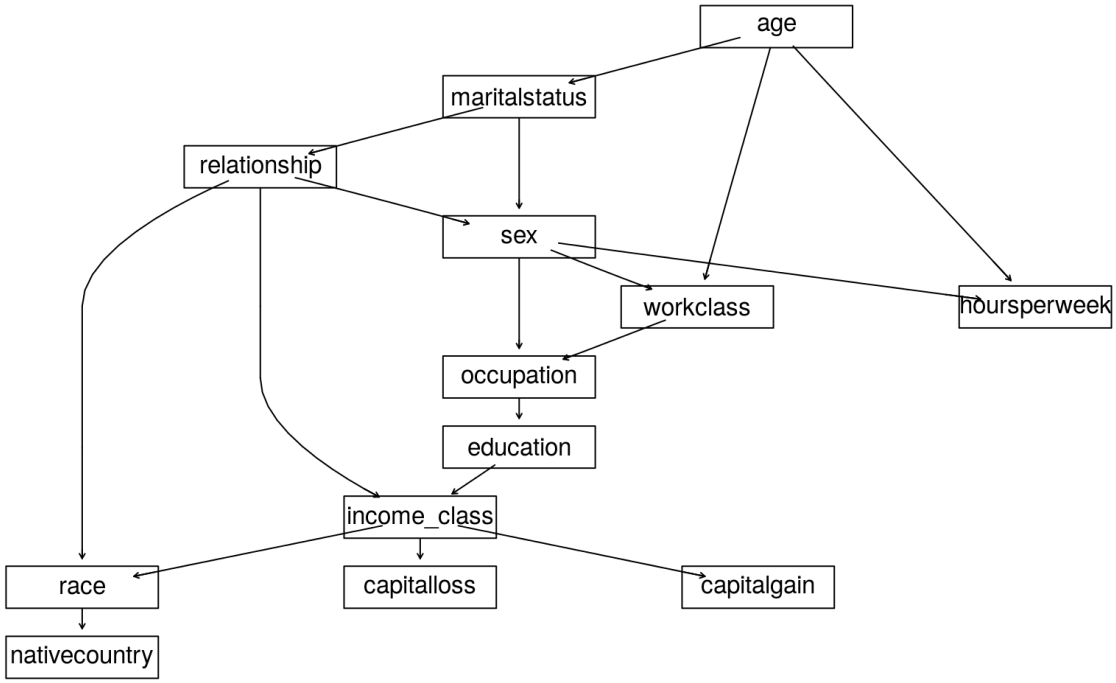


Figure 4.2: Bayesian network learned from the adult dataset.

is randomly generated by observing the BN. The list of dependencies is represented by the directed edges in Figure 4.2. For sampling, we start with the independent node in the BN, which is age. We sample age according to its distribution. After sampling age, we can sample the marital status. The conditional probability table for marital status is given in Table 4.2. Notice that age is divided into bins, which means that younger people are represented by bins identified by smaller numbers and older ones by larger numbers. We omit some of the bins due to space constraints.

Table 4.2: Conditional probabilities of marital status given some age categories (1-younger,10-older)

Marit. Status \ Age Cat.	2 (22-28)	5 (46-52)	8 (70-76)
Divorced	0.087	0.206	0.081
Married-AF-spouse	0.001	0.000	0.002
Married-civ-spouse	0.372	0.630	0.560
Married-spouse-absent	0.011	0.016	0.014
Never-married	0.490	0.077	0.060
Separated	0.032	0.033	0.014
Widowed	0.003	0.035	0.265

This process continues until we have sampled every feature. We then repeat this process until we have enough rows sampled, thus generating a synthetic dataset that follows the same distribution as the original adult data.

In order to produce biased samples, consider Table 4.3, which contains conditional probabilities for the income class node given education levels (Edu-level) and relationships

(r). We can see that, according to this table, the probability of a positive outcome (the individual earns more than \$50K per year), because the individual is a wife, is slightly better with higher education levels, but it is worse for lower education levels. In order to produce even more bias, we must decrease the probabilities in the first line of Table 4.3. For instance, we may decrease the probability of high income, considering that the relationship is *wife*. This process is done gradually, each time changing by 10% of the original probability, until the probability reaches 0, as described in Section 4.2. For example, the probability of an individual having a higher income, considering she is a woman and her education level is 10th grade, is 0.100. The first artificial dataset is sampled by lowering this probability to 0.090, the second one to 0.080, and so forth.

The resulting sampled datasets are then used as training data for a classifier. We chose the C4.5 decision tree classifier due to its good results on previous works of fair ML. Since we aim to evaluate bias mitigation techniques, the choice of the classifier is not that important.

Table 4.3: Conditional probabilities of high income (above \$50K a year) given education and relationship

Cond. prob \ Edu-level	1st-4th	10th	HS	Doctorate
$P(y = high \mid r = wife)$	0.000	0.100	0.343	0.850
$P(y = high \mid r = husband)$	0.085	0.128	0.311	0.837

4.6 Summarizing results

Fair ML works often present the result of their bias-mitigation techniques as a 2D plot of quality and fairness metrics, which is, indeed, a good way to demonstrate their performance because the trade-off between these two metrics becomes evident. However, relying only on such representation may become quite confusing when several experiments are performed, or when parameter tuning and comparisons between two or more techniques become necessary. Since we are designing a benchmark framework, it is important to describe how the results will be presented. Therefore, we introduce a new approach to present the performance of a technique in a more compact yet informative way.

Our approach is based on a 2D plot where the x-axis represents the Fairness Metric Improvement and the y-axis the Quality Metric Degradation. Each technique is associated with a vector that represent the impact of applying the technique in terms of direction

and magnitude of change of the two metrics. Algorithm 3 shows how we summarize the results obtained by Algorithm 1.

Algorithm 3 Result Summarizing

Require: $\mathcal{R}, q_c, f_c$

1: **for** $r = \{q_t, f_t\} \in \mathcal{R}$ **do** ▷ $\mathcal{R}$ stores pairs quality-fairness of results
2: $q_d \leftarrow \text{MIN}(\frac{2q_t}{q_c} - 1, 1)$
3: $f_i \leftarrow \text{MAX}(1 - |\frac{f_t}{f_c}|, 0)$
4: **end for**
5: $\bar{q_d} \leftarrow \text{AVG}(\mathcal{R}_q)$
6: $\bar{f_i} \leftarrow \text{AVG}(\mathcal{R}_f)$
7: $\theta \leftarrow \text{TENDENCYFN}(\mathcal{R})$
8: $\{\omega_q, \omega_f\} \leftarrow \text{RELIABILITYFN}(\mathcal{R})$
9: $q_d^* \leftarrow \bar{q_d} - \omega_q \bar{q_d} \text{COS}(\theta)$
10: $f_i^* \leftarrow \bar{f_i} - \omega_f \bar{f_i} \text{SIN}(\theta)$

Initially we introduce the concept of "Quality Metric Degradation" q_d (line 2) which quantifies by how much a technique compromises the quality metric according to the value q_c obtained by a traditional classifier. The same goes for "Fairness Metric Improvement" f_i (line 3), which quantifies how much the technique improves the fairness metric. We define them as follows:

Definition 4 *Given a quality metric value $q_c \in \mathbb{R} : \{0 \leq q_c \leq 1\}$ obtained by measuring quality (e.g., accuracy) of a classifier when no bias mitigation technique is used and $q_t \in \mathbb{R} : \{0 \leq q_t \leq 1\}$ as the quality measure after applying a bias mitigation technique. We define the Quality Metric Degradation q_d (line 2) by:*

$$q_d = min\left(\frac{2q_t}{q_c} - 1, 1\right) \tag{4.3}$$

Definition 5 *Given a fairness metric value $f_c \in \mathbb{R} : \{-1 \leq f_c \leq 1\}$ obtained by measuring the fairness metric of a classifier (e.g., statistical parity) when no bias mitigation technique is used and f_t as fairness metric after applying a bias mitigation technique. We measure the Fairness Metric Improvement f_i (line 3) by:*

$$f_i = max\left(1 - abs\left(\frac{f_t}{f_c}\right), 0\right) \tag{4.4}$$

We calculate (f_i, q_d) for each r in $\mathcal{R}$ and then we calculate their average values (lines 5-6), a pair $(\bar{f_i}, \bar{q_d})$ indicating our initial point for a technique, that is, the technique's performance according to the quality and fairness metrics. An optimal point is $(1, 1)$, which means the technique has successfully mitigated all bias inserted and also kept the quality on par with the classifier. On more realistic scenarios, we may have $(0, 1)$, as it is the expected result of not applying any technique and using the same result as the

classifier to compute $(\bar{f}_i, \bar{q}_d)$; that is, it doesn't improve the fairness metric but also doesn't degrade the accuracy.

Although the initial point $(\bar{f}_i, \bar{q}_d)$ may describe the technique's outcome, we still need to identify the trend as we vary β. Therefore, we must represent this trend across multiple scenarios (in our case, multiple levels of artificial bias inserted) and also its overall reliability. Our proposal is to express this trend through a terminal point (f_i^*, q_d^*) and the vector defined by both $(\bar{f}_i, \bar{q}_d)$ and (f_i^*, q_d^*). Depending on the vector's direction and length, we grasp the technique's tendency and reliability. The terminal point (f_i^*, q_d^*) should be closer to the the plot origin, which means we need to find $(\delta_f, \delta_q) \in \mathbb{R}^2 : (0 \leq \delta_f \leq \bar{f}_i, 0 \leq \delta_q \leq \bar{q}_d)$ to determine the terminal point:

$$(f_i^*, q_d^*) = (\bar{f}_i - \delta_f, \bar{q}_d - \delta_q) \tag{4.5}$$

The reasoning behind terminal points captures two concepts. The first concept is to understand which metric the technique is prioritizing (i.e., fairness or quality). The second is the reliability of a technique w.r.t. each metric. Explaining the first concept, $\delta_f < \delta_q$ suggests that a technique prioritizes fairness over accuracy, $\delta_q < \delta_f$ suggests quality prioritization, and $\delta_f = \delta_q$ suggests equivalent treatment. The second concept is explained similarly; larger values of δ_q suggests that a technique may not be much reliable on keeping the quality metric high, the same goes for δ_f and the fairness metric. Visually, larger values of δ_q reduces the Quality Metric Degradation and higher values of δ_f reduces the Fairness Metric Improvement.

We can calculate (δ_f, δ_q) by considering both concepts at the same time through the modeling of Equation 4.5 as the following trigonometrical equation:

$$(\bar{f}_i - \delta_f, \bar{q}_d - \delta_q) = (\bar{f}_i - \omega_f \bar{f}_i cos\theta, \bar{q}_d - \omega_q \bar{q}_d sin\theta) \tag{4.6}$$

In order to calculate (δ_f, δ_q), we define a direction $\theta \in \mathbb{R} : 0 \leq \theta \leq \pi/2$ and normalization values $(\omega_f, \omega_q) \in \mathbb{R}^2 : (0 \leq \omega_f \leq 1; 0 \leq \omega_q \leq 1)$. θ represents the metric prioritization trend in adverse scenarios. The normalized values (ω_f, ω_q) represent our trust in the initial point regarding each metric. Value θ is computed by a tendency function TendencyFn($\mathcal{R}$) and the pair (ω_f, ω_q) is computed by a reliability function ReliabilityFn($\mathcal{R}$).

Currently, our trend function comes from the quadratic coefficient α of a 2-degree polynomial regression on the results r.

Figure 4.3 is a 2D plot of statistical parity vs. accuracy for five different techniques as we vary β. We run a polynomial regression for each technique, thus, obtaining the curves. We argue that the spatial distribution of the points may describe whether the technique prioritizes quality over fairness metric or not. In a 2-degree polynomial function, a highly positive or negative quadratic coefficient α suggests that the points don't deviate

too much on x-axis (fairness metric), therefore the technique might be prioritizing the fairness metric over quality. On the other hand, α values close to 0 (flatten curves) may indicate that a technique keeps the points steady on y-axis (quality metric), thus, prioritizing quality metric over fairness.

In Equation 4.7, we calculate the *arctan* (tan^{-1}) of $|\alpha|$. We use the absolute value because a highly positive and a highly negative quadratic coefficient will both indicate that the points are very spread vertically. If $|\alpha|$ increases, $tan^{-1}(|\alpha|)$ approaches $\pi/2$, thus, increasing the value of δ_q, because $sin(\theta)$ approaches 1. Alternatively, if $|\alpha|$ approaches 0, so does θ, therefore, δ_f increases as $cos(\theta)$ approaches 1.

$$\theta = tan^{-1}(|\alpha|) \tag{4.7}$$

We also propose a reliability function that calculates (ω_f, ω_q) by taking the mean of the standard deviations from the fairness and quality metrics, respectively. We argue that, although simple, the standard deviation tells whether a bias mitigation technique kept the results stable as we vary β. Notice that, for each bias level β, a bias mitigation technique should be run a sufficient amount of times (usually 30) in order to get statistically significant results. For each one of these runs, we compute the standard deviation for both fairness and accuracy and, finally, the pair (ω_f, ω_q) is calculated by taking the average of these standard deviations.

The resulting summary can be seen in Figure 4.4. The arrows represent, for each technique, the vector from point (f_i, q_d) to (f_i^*, q_d^*). We also depict a separation defined by the line $y = x$. If a point lies above this line, we interpret it as benefiting quality over fairness, otherwise, the reverse. Moreover, the direction an arrow points to may suggest the technique's trend.

Going over the results in Figure 4.4, we have a vertical blue arrow starting from $(0, 1)$, this arrow represents a default result of the classifier (Technique 1), which means that no bias mitigation technique is applied, leaving it just the classifier that optimizes only the quality. Notice that it has a little variance, thus, it has a small value for ω. The two vertical arrows on the right-side (Yellow and Red, Techniques 2 and 4) look very similar, they share a similar Polynomial Regression as shown in Figure 4.3. One may argue that Technique 2's results are more spread, therefore, it is more susceptible to variance, however, we measure this ω according to the overall variance of each point, not the result as a whole. Also, since we take the absolute value of the slope, it doesn't matter whether Technique 2 and 4 have opposite signs, we only take into account how steep or how flat is the curve. We have strong evidence that the almost horizontal arrow represented by Technique 3, in green, prioritizes the quality metric given its result depicted in Figure 4.3. Finally, the more diagonal arrow representing Technique 5 in purple starts from a point that prioritizes quality over fairness, however, due to high variance, decays considerably and in almost equal proportion. This can also be naturally perceived in Figure 4.3.

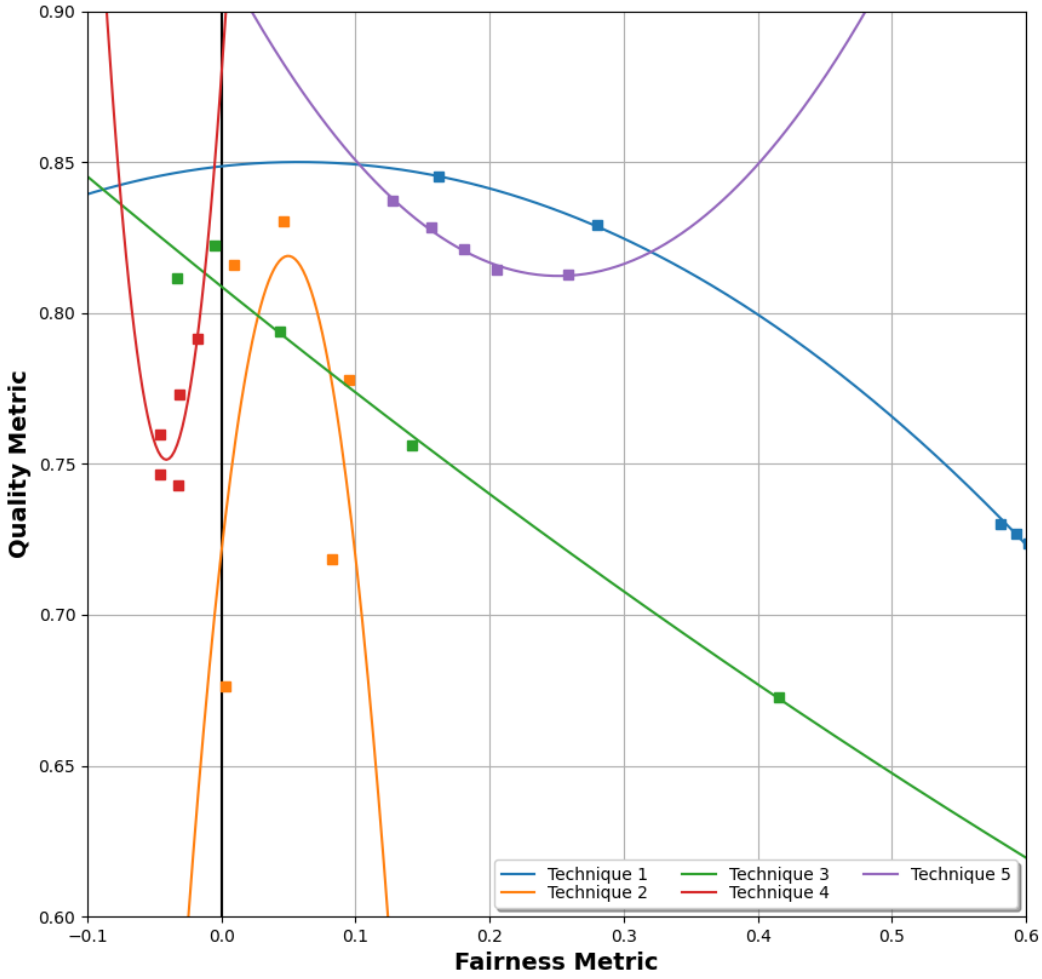


Figure 4.3: Benchmark result summarizing methodology using 2-degree polynomial regression with root squared standard deviation

4.7 Framework Use Cases

In this section we present three uses cases of our framework, demonstrating its generality and applicability. These use cases illustrate a number of potential trade-offs the decision maker should consider to interpret the benchmark results. The first use case depicts a comparison between two techniques, followed by multiple comparisons with quality and fairness constraints set, and then we show how to tune parameters of a technique using the framework.

In Figure 4.5 and Figure 4.6 we have a **Comparison between two different bias mitigation techniques**. Figure 4.5. represents the benchmark result obtained by

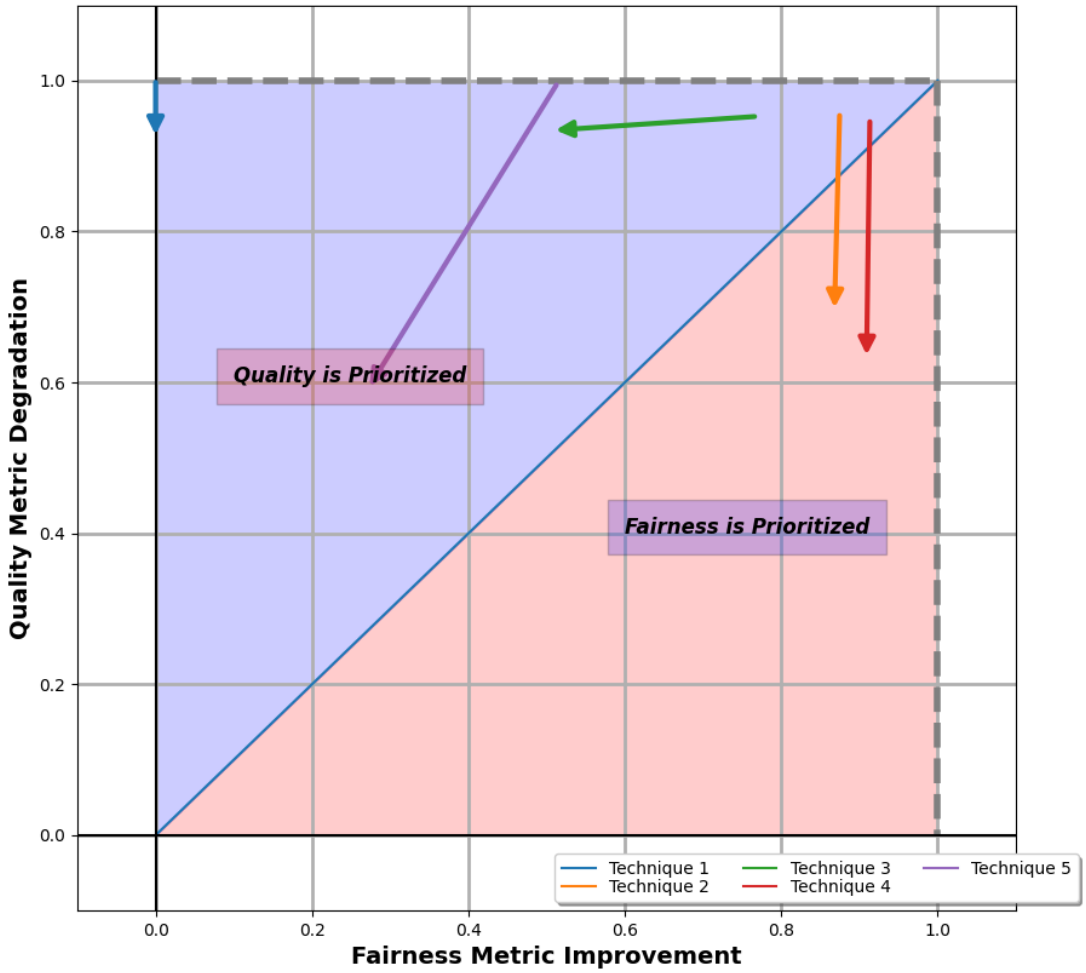


Figure 4.4: Compact arrow representation of summarized result.

varying bias level β and using artificially generated datasets. Technique 1 provides a set of more spread points on the positive side of fairness, as opposed to Technique 2, which has points that are more concentrated on the negative side.

Translating this first result to our compact representation we obtain the arrows in Figure 4.6. We can notice how similar those techniques are to each other, the arrows are very close and aim at the same direction (prioritizing the fairness metric). This accords with our approach since it can be observed that both curves have a high absolute first coefficient values, therefore the prioritization should lean to such direction. Regarding the norm of those vectors and considering that it is calculated given the variance of the results, one may ask why the arrows are similar despite Technique 1 points being more spread than those from Technique 2. It is worth mentioning that the variance is averaged from each point, which means that even though a set of points is more concentrated, it

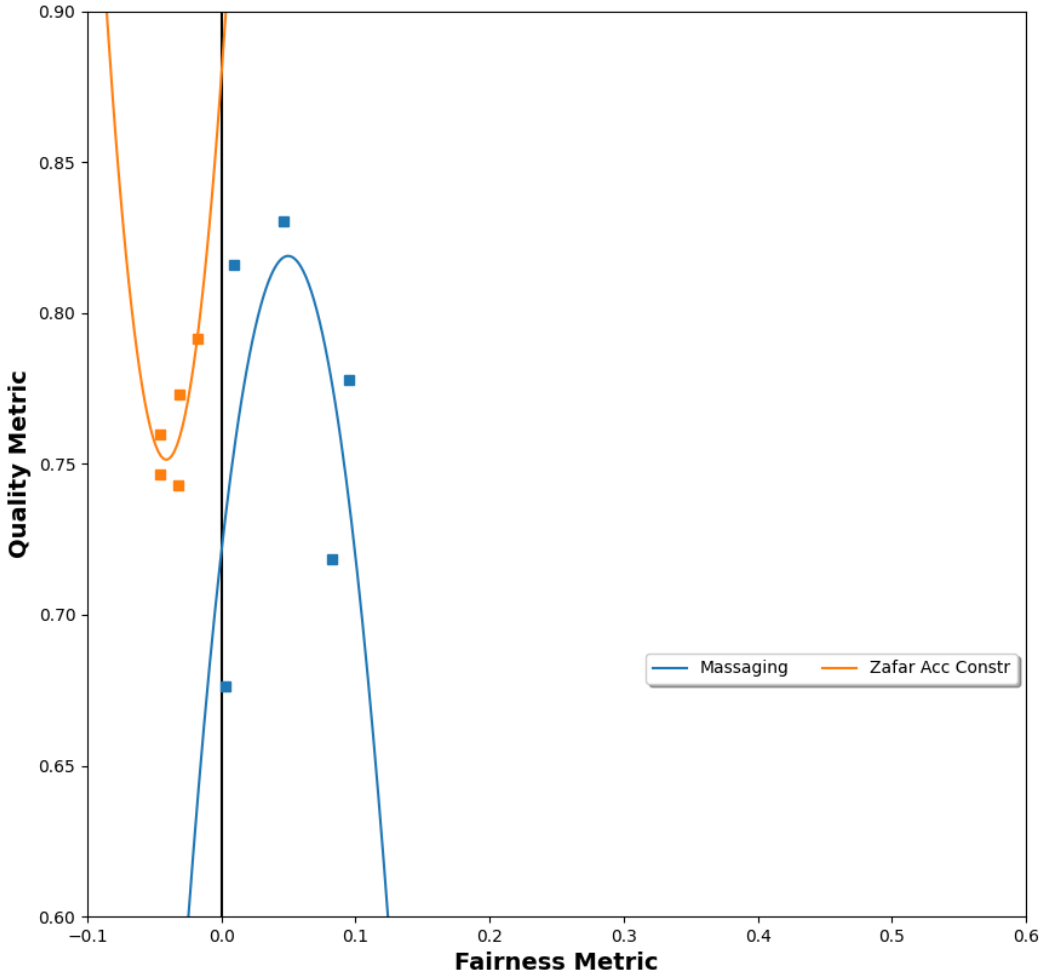


Figure 4.5: Use Case 1 - Comparing two techniques: 2-degree polynomial regression

may result in higher variance if each point provides higher variance.

We expect that, in most cases, decision makers either want to choose a technique or do **Multiple Comparisons** among a number of pre-selected ones. In this new hypothetical situation, we have a list of techniques applied to the same data. One of them must be selected to be applied to the future. Suppose, due to an application constraint, that the quality metric must not degrade below 60% of its original result using a traditional classifier. The same goes for the Fairness Metric Improvement, which is calculated according to a target.

Let f_c denote the fairness metric obtained using the traditional classifier and $f_t < f_c$ a target fairness metric we must obtain using a technique (e.g, the classifier provides 0.5 and our target is 0.2). The ratio $\frac{f_c - f_t}{f_c}$ (e.g., 0.6) is the minimum fairness metric improvement. In other words, a technique must make the fairness metric drop from 0.5

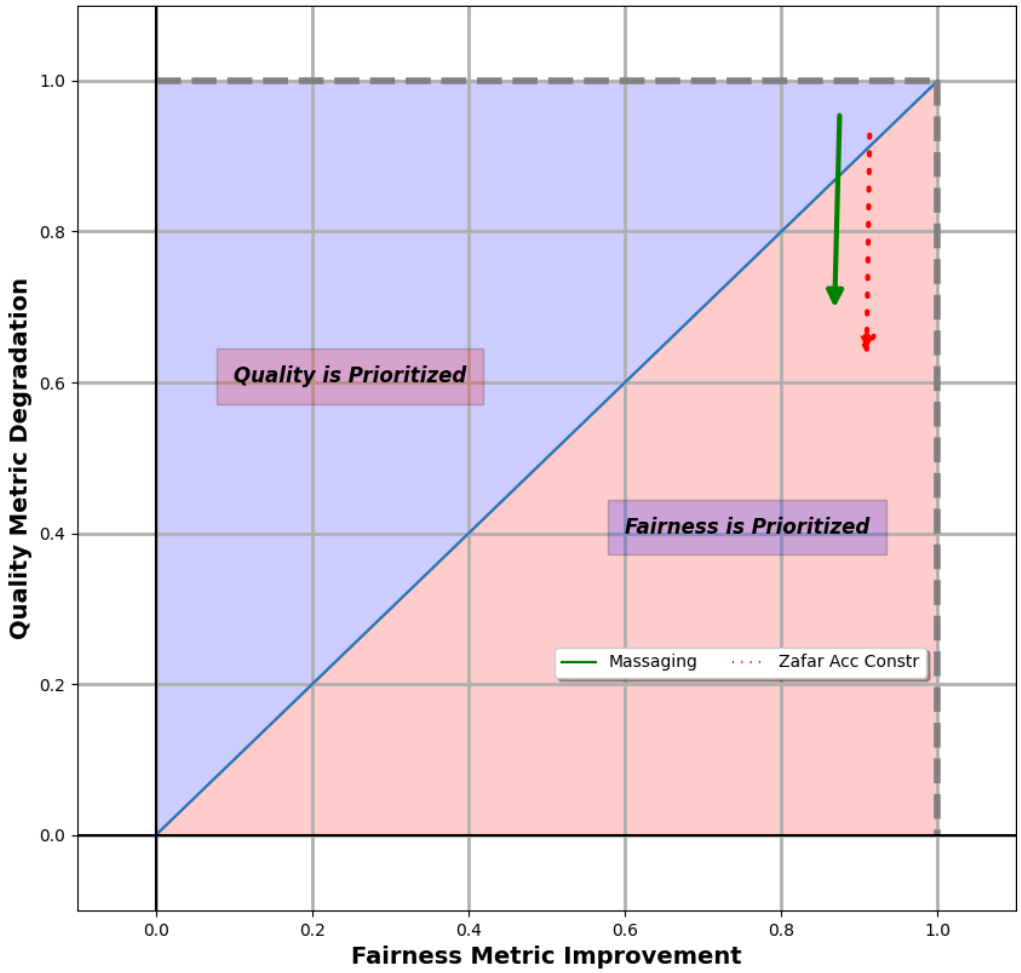


Figure 4.6: Use Case 1 - Comparing two techniques: Arrow representation

to at least 0.2 in order to meet the requirement of improving it by 60%. Thus, we have the graph in Fig 4.7, where the shadowed area represents all techniques that satisfy this constraint.

Alternatively, in Fig 4.8 we have another pair of constraints, now more restrictive on the quality metric as it cannot degrade more than 80% and can be more relaxed on the fairness metric as it is acceptable to improve it at least by 40% (e.g., must drop from 0.5 to at least 0.3). The shadowed area in Fig 4.8 includes different techniques from Fig 4.7 and the decision maker must reason on those new included techniques.

Some bias mitigation techniques require **parameter tuning** to be applied (e.g., for BlackBoxAuditor [13]). Researchers in the field must have appropriate evaluation techniques to calibrate their parameters according to their objectives. This third use case describes an example of a parameter tuning task and how it can be useful for the re-

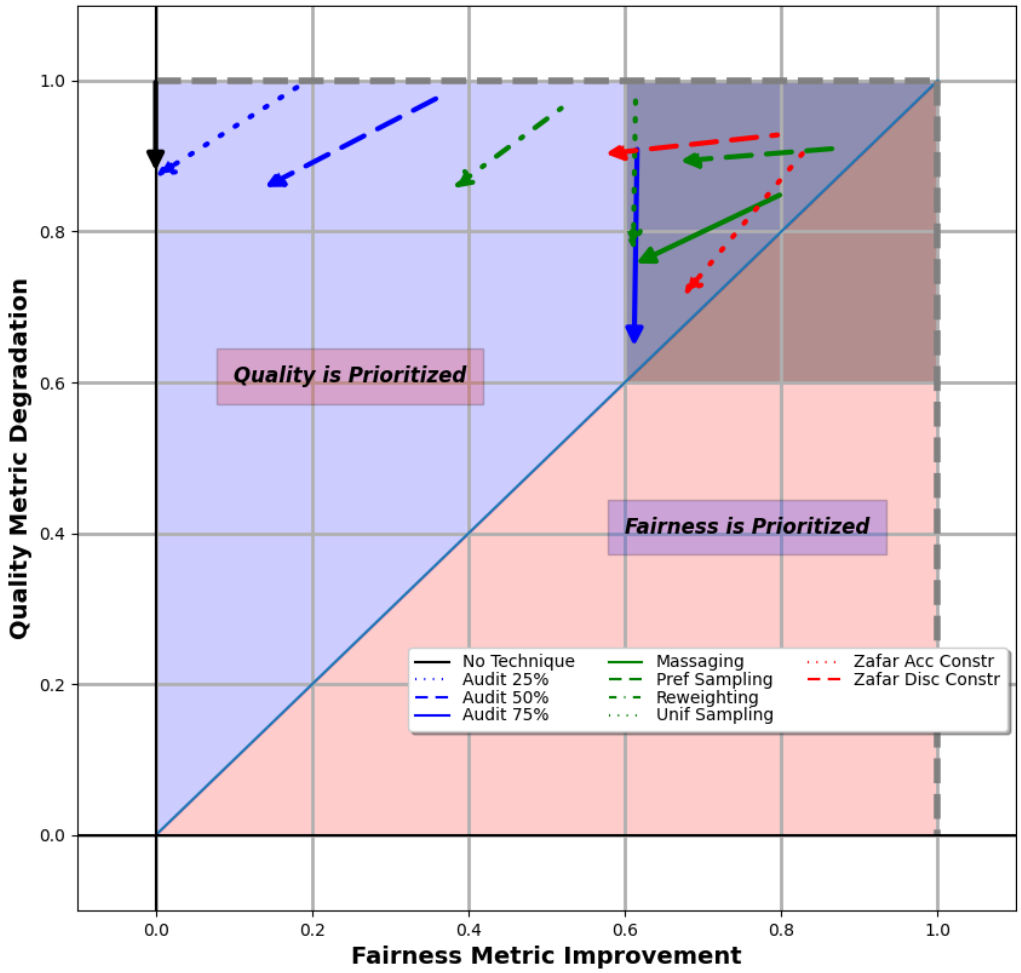


Figure 4.7: Use Case 2 - Choosing a technique in a quadrant given Minimum Fairness Metric Improvement (FMI) and Maximum Quality Metric Degradation (QMD): Min. FMI 0.6 Max. QMD 0.6

searchers, who may discover inherent behaviors of a technique given the arrow dispositions and consider further improvements in their implementations.

Figure 4.9 and Figure 4.10 compare two different parameter tuning trials using different datasets for the Auditor technique. Each blue arrow represent a result for Auditor using repair levels of 0.25 (dotted line), 0.50 (dashed line) and 0.75 (solid line). The first impression on both datasets is how the arrows lean towards fairness prioritization as the repair level increases, but it can be noticed that the quality metric is compromised. In Adult dataset a repair level of 0.75 degrades the quality the most, so it would be interesting to try additional experiments varying the repair level between 0.50 and 0.75. Moreover, in Dutch dataset there is not much variance when using 0.25 for repairing, therefore, the

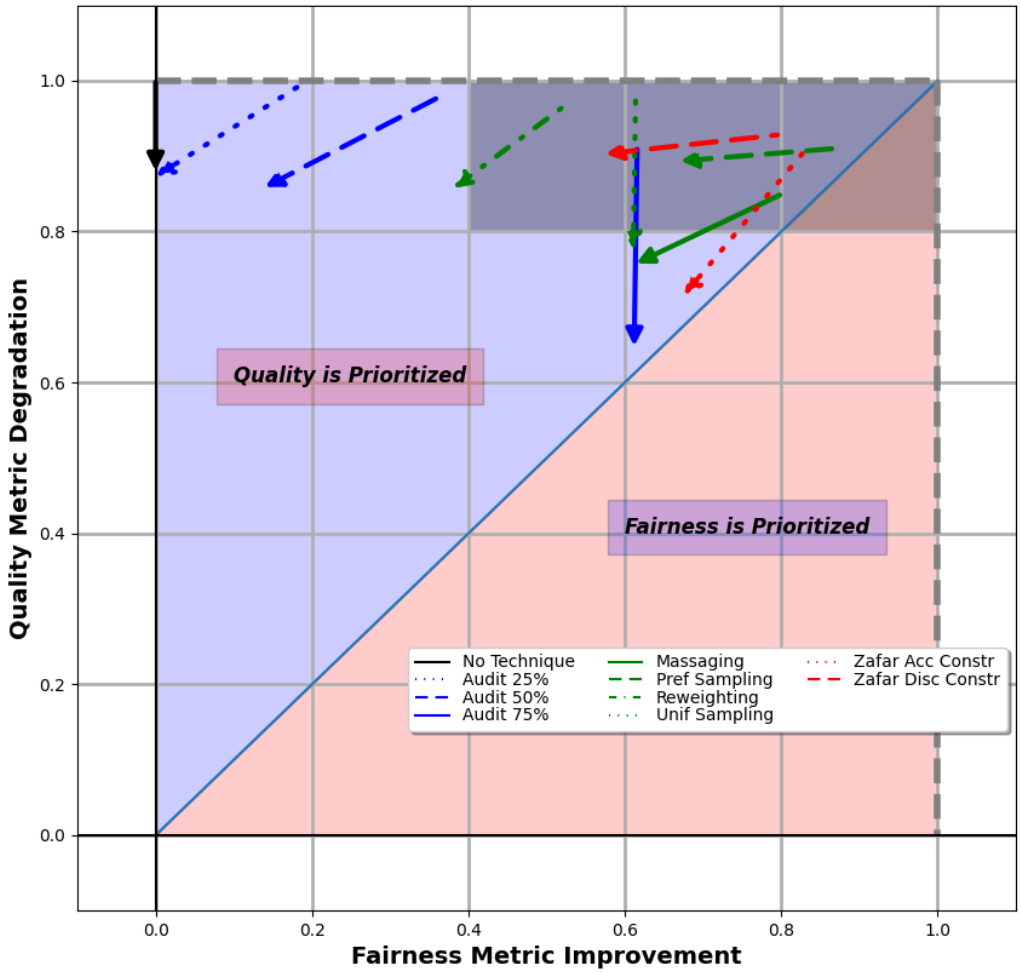


Figure 4.8: Use Case 2 - Choosing a technique in a quadrant given Minimum Fairness Metric Improvement (FMI) and Maximum Quality Metric Degradation (QMD): Min. FMI 0.4 and Max. QMD 0.8

vector norm is not too high, but it doesn't improve much on the fairness.

Naturally one must argue that the best case scenario would be selecting the closest technique to the (1,1) point in the graph. That is generally true, however, a decision maker may not always select the technique that obtains the best result using this approach due to a number of reasons. Firstly, there are costs involved for each technique and applying the technique with the highest performance may not be always feasible. Also, the results are presented for a single dataset, whereas it would be interesting to compare the techniques using a broad set of results.

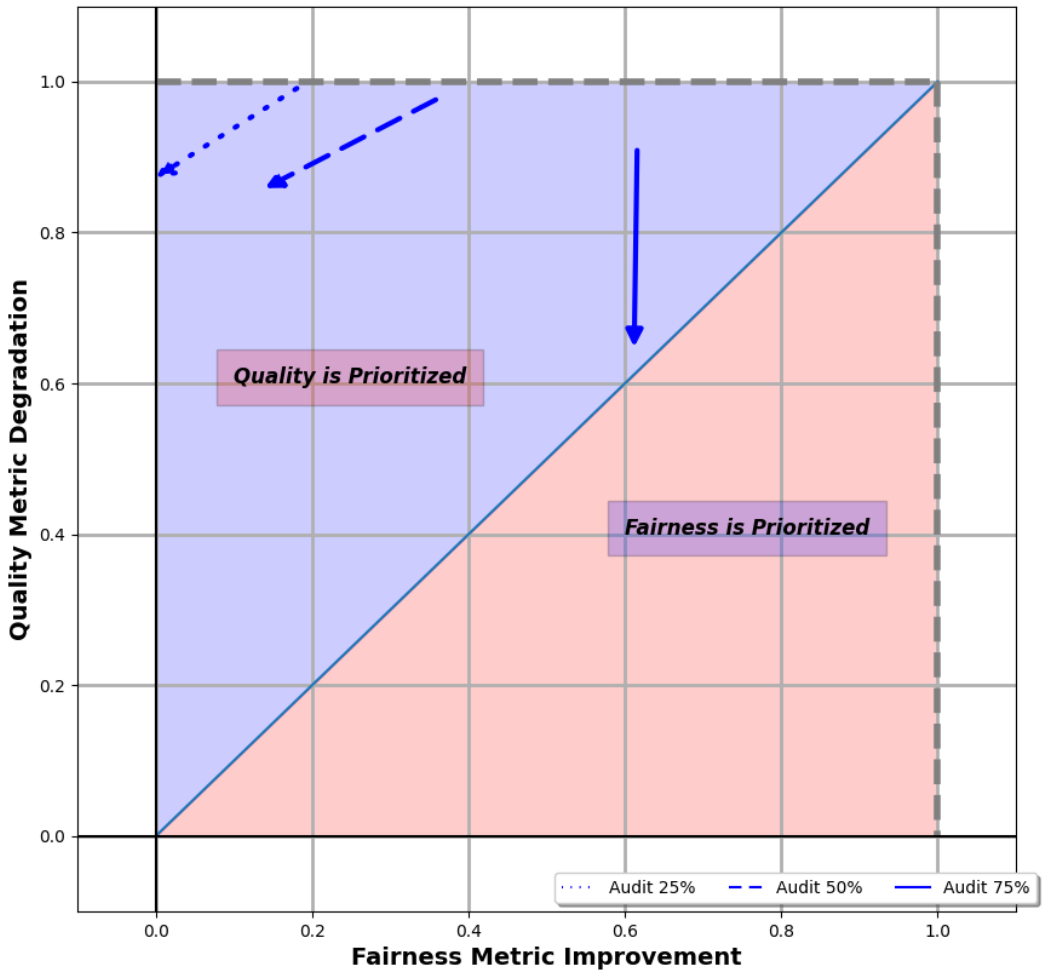


Figure 4.9: Use Case 3 - Parameter Tuning of a bias mitigation technique: Dataset 1 (UCI Adult)

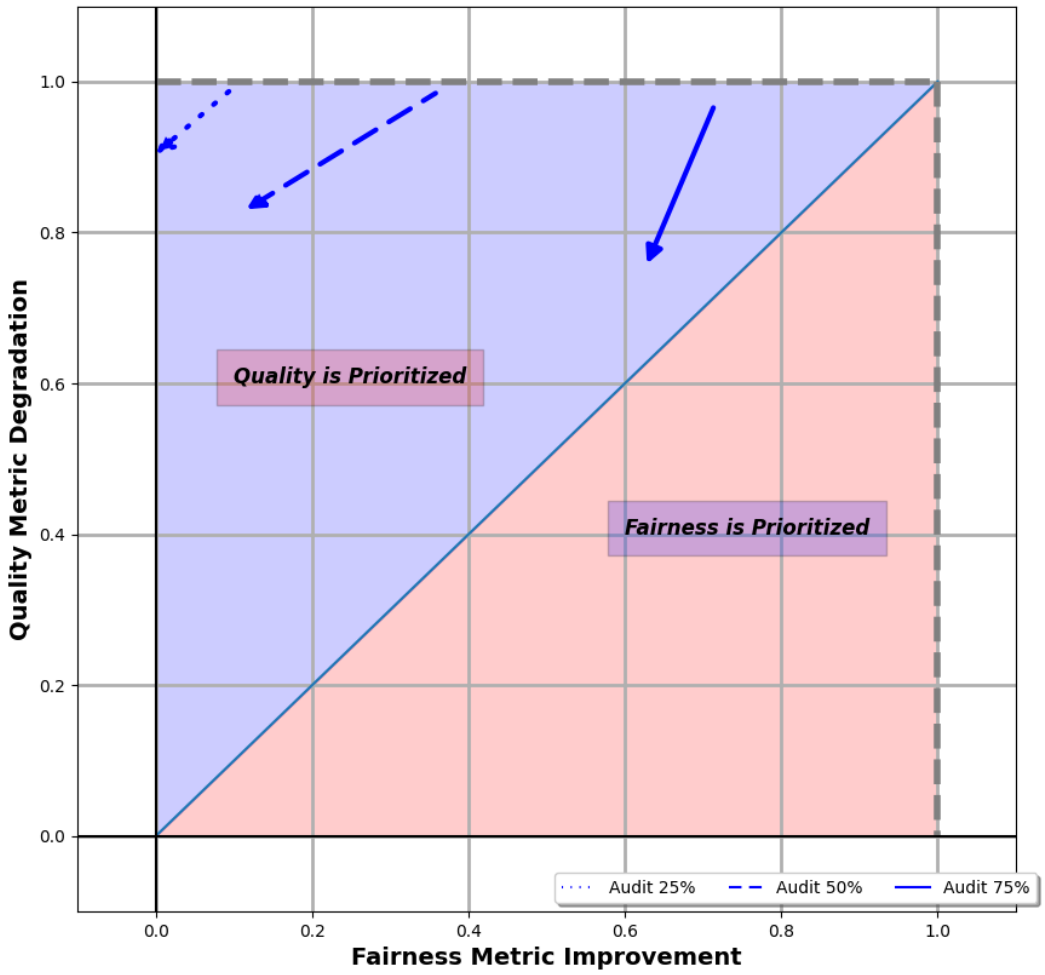


Figure 4.10: Use Case 3 - Parameter Tuning of a bias mitigation technique: Dataset 2 (Dutch census)

Chapter 5

Experiments

We compare well-known bias mitigation techniques by testing them on the artificially biased datasets. As mentioned earlier, when a pre-processing technique is used, it outputs a modified dataset that is used by a traditional classification models. The validation dataset is a split of the real data that was used to build the generative models (Bayesian networks). Evaluation is done in two different ways. Firstly, by comparing the fairness metric of the prediction to the quality metric on the validation dataset. In this work we assume **accuracy** of the prediction as the quality metric and **statistical parity** as the fairness metric, which will be named 'bias' for the sake of simplicity. We also measure overestimation (3.3) and underestimation (3.4) of the resulting predictions, which are two related fairness metrics that will be compared against each other. Finally, we employ our visualization method to compress the benchmark results in a more compact way to represent them.

5.1 Bias Mitigation Techniques

The techniques we compare are listed in the Related Work section. We made use of Weka's original implementation of those techniques [42] [1]. We had to make some modifications in order those techniques could work on our models and we switched to AIF360's implementation later on. We will omit the Adversarial De-biasing technique, and the post-processing technique as the main objective here is to show the framework's concept of comparison.

[1] https://sites.google.com/site/conditionaldiscrimination/

5.2 Baseline

The **baseline** consists of removing the protected attribute from the training set. It has been argued that this may even increase bias in some cases [32] since some of the attributes might be conditionally dependent to the protected attribute. For example, the neighborhood information may carry racial information about individuals. This problem is known as *redlining*.

For the first batch of results, we chose several pre-processing techniques such as massaging, re-weighting, uniform sampling and preferential sampling [19], and also the BlackBox Auditing from [13], for that one we use repairing levels of 0.20, 0.50 and 0.80. We also consider one in-processing technique: Fair Classification Learner from [39]

5.3 Datasets

We used four datasets in our experiments: Adult Census, COMPAS, Dutch Census and German Credit. It would be interesting to perform these experiments on a wider variety of scenarios, however our main focus is to demonstrate the benchmark concept, its advantages and where it can be extended.

The **Adult dataset** [23], also known as Census Income, is a widely used dataset in previous fair ML studies. The task is to predict whether an individual has a yearly income greater than \$50K or not (i.e., high vs. low income). It has 48,842 instances with 14 attributes, although we clean the data by removing the `fnlwgt` attribute (because it is a noisy attribute that does not contribute to the analysis) and discretizing some of them. We decided to use this dataset because it is arguably the most widely used in the task of fair ML. The protected attribute is sex, and the original dataset has an inherent bias against women (equal to 0.19). For generating biased data, we increase the influence of the feature *relationship* on the outcome, thus making it less likely that wives have a high income, thus increasing the bias against women.

The **Pro Publica COMPAS dataset** [2] records racial bias on recidivism scores. The data contain information from defendants such as race, age, criminal history, and whether the defendant had committed a crime within a two-year window. It has 6,150 instances with 13 attributes. The sensitive attribute is `race`, which can be either "Caucasian" or "African-American". We removed redundant and non-relevant features such as

[2] https://github.com/propublica/compas-analysis

day of arrest, name, and case number. We modified the influence of the variable race on the outcome for generating biased data. We consider only defendants described as Caucasian and African-American.

Dutch census [7] is also a demographic census from Netherlands. It was considered the same sample used by [19]. We use it to make predictions of whether an individual has a "high level" occupation or not. This dataset has 11 attributes, and we define sex as the protected attribute.

The **German credit data**[3], composed of 1000 tuples with 20 attributes, is also from the UCI Machine Learning Repository. This dataset has, for each tuple, a classification of credit score (either *good* or *bad*) for the individual represented by that tuple. We use the categorical version of this dataset, and the protected attribute is age category, either *young* or *old*.

5.4 Bayesian Network Learning

As discussed earlier, we tested two methods for Bayesian Network Learning: Hill-Climbing greedy search and Tabu Search algorithm [28]. They both showed very similar log-likelihood scores when we learned Bayesian Networks for both the Adult dataset and COMPAS dataset. The same Bayesian Network was learned for the case of Adult dataset using either Hill-Climbing or Tabu search. Another way to assess the quality of the Bayesian Network we have learned is to use the data sampled from it in a classification task. By doing that, we have evidence that our synthetic data still describe the relations between the variables. We generated synthetic datasets with different levels of bias for both Adult and COMPAS using the Hill-Climbing greedy search (which is faster), and we used them as training sets in a classification task where the real dataset is used for validation. Table 5.1 and Table 5.2 show the performance of different classifiers on each dataset used for different levels of bias. This result suggests that we should indeed use the C4.5 decision tree for the classification task.

Table 5.3 and Table 5.4 present some properties of the Bayesian networks for Hill-Climbing and Tabu search algorithms, respectively. Notice, for each one of these tables, the results regarding the classification task: Accuracy and Bias. This classification task consists of using the data sampled from the learned Bayesian networks for each dataset and using this data as a training set. 80% of each dataset was used for the learning process, and the remaining part was used for evaluation. The classifier used was the C4.5 decision tree, and the experiment was performed 30 times with different samples. We

[3]https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data)

Table 5.1: Accuracy obtained from different classifiers on biased data given a bias level on adult dataset

Bias \ Class.	N. Bayes	Log. Regr.	IB1	IBk	ANN	J48
0.00	0.802 (0.028)	0.844 (0.006)	0.762 (0.031)	0.817 (0.013)	0.830 (0.055)	**0.845** (0.007)
0.25	0.744 (0.009)	0.828 (0.020)	0.749 (0.027)	0.802 (0.025)	0.795 (0.235)	**0.829** (0.024)
0.50	0.731 (0.003)	**0.759** (0.026)	0.733 (0.022)	0.740 (0.020)	0.740 (0.154)	0.727 (0.032)
0.75	0.727 (0.002)	**0.732** (0.009)	0.716 (0.020)	0.718 (0.004)	0.727 (0.015)	0.727 (0.007)
1.00	0.722 (0.002)	0.724 (0.007)	0.695 (0.016)	0.715 (0.002)	**0.725** (0.017)	0.724 (0.004)

Table 5.2: Accuracy obtained from different classifiers on biased data given a bias level on COMPAS dataset

Bias \ Class.	N. Bayes	Log. Regr.	IB1	IBk	ANN	J48
0.00	0.877 (0.008)	0.888 (0.005)	0.821 (0.105)	0.887 (0.012)	0.885 (0.024)	**0.888** (0.004)
0.25	0.876 (0.014)	**0.889** (0.008)	0.827 (0.078)	0.887 (0.011)	0.885 (0.022)	0.888 (0.003)
0.50	0.877 (0.011)	**0.889** (0.006)	0.820 (0.094)	0.887 (0.012)	0.884 (0.028)	0.887 (0.004)
0.75	0.877 (0.009)	**0.888** (0.012)	0.819 (0.106)	0.887 (0.009)	0.885 (0.030)	0.887 (0.003)
1.00	0.876 (0.012)	**0.888** (0.012)	0.824 (0.095)	0.887 (0.013)	0.884 (0.036)	0.887 (0.005)

Table 5.3: Properties of the Bayesian networks learned using Hill-Climbing algorithm for each dataset

	Adult census	COMPAS data	Dutch census	German Credit data
Dataset size	32561	6150	60420	1000
Nodes	13	13	12	21
Arcs	18	15	21	17
Score	-12.09	-7.18	-9.60	-21.34
Accuracy	84.50	87.84	83.27	71.58
Bias	16.23	12.15	32.22	19.68

take both the accuracy and the resulting bias according to Definition 1 (statistical parity difference). Notice that both methods of estimating Bayesian networks, Hill-Climbing and Tabu search, achieved similar results.

Table 5.4: Properties of the Bayesian networks learned using Tabu search algorithm for each dataset

	Adult census	COMPAS data	Dutch census	German Credit data
Dataset size	32561	6150	60420	1000
Nodes	13	13	12	21
Arcs	18	15	21	17
Score	-12.09	-7.18	-9.60	-21.34
Accuracy	84.50	87.84	83.27	71.58
Bias	16.23	12.15	32.22	19.68

5.5 Experimental Setup

We evaluate various bias mitigation techniques on our set of systematically-biased datasets generated from the real-world data mentioned above. This set consists of 4 training datasets with increasing levels of artificial bias (β) against individuals of the defined deprived group. Each dataset is used as an input for the technique, and the expected output is either a new dataset that should be used in the training process (pre-processing techniques) for classification or an already built bias mitigation classification model (in-processing techniques). For the classifier, we use the Weka implementation of the C4.5 decision tree [17]. We reproduced the experimental setup described by the authors of the techniques and used default parameters [23, 42]. After that, we measure the accuracy and bias observed when the original data is used as the testing data. We use the biased test sets for validation since there is no known non-biased ground truth. Note that we refer to statistical parity difference as *bias* (3.1), and measure disparate mistreatment via *overestimation* and *underestimation* (3.3 and 3.4).

5.6 Artificially Generated Datasets

The task of making an artificially generated dataset consists of changing the influence of the sensitive attribute's node on the outcome and so making it more biased according to a bias level. Section 4.6 explained through a practical example how we artificially generate a biased **adult** dataset. We increase the influence of the feature *relationship* on the outcome, thus making it less likely that wives have the high-income class label, which means that there is a bias against women as well. We do the same for **Dutch Census**. In **COMPAS** dataset we increase the influence of the feature *race* on the outcome and finally, for **German Credit**, we increase the influence of Age on the outcome.

Chapter 6

Results and Discussion

In this section, we present the results of applying our framework to compare several techniques. An experiment consists of, given a real-world dataset $\mathcal{D}$, using an dataset $\mathcal{D}_t$ artificially generated by our methodology as input for a fair ML technique $\mathcal{T}$, making predictions on the real world data split $\mathcal{D}_s$ and evaluating those predictions with both quality and fair classification metrics. Initially, we display a traditional approach to represent results for Adult Census Dataset, then we apply the methodology described in Section 4.6

6.1 Bias vs. Accuracy tables

We perform the following experiment for a couple of reasons. The first one is to make sure that removing the protected attribute does not necessarily reduce the resulting bias (here, statistical parity difference) in the classification task. The second reason is to define upper bounds for both bias and the quality metric, (here, accuracy). That is, the bias without any technique being applied is expected to be higher than when some technique is used; otherwise, this bias mitigation technique would be unnecessary. It is also expected that the accuracy decreases when a bias mitigation technique is applied because the technique tries to lower the bias at, presumably, the cost of the accuracy.

We can observe in Table 6.1 that, as we increase level of artificial bias inserted, β, the bias on the Adult census increases, and the accuracy decreases, which means that highly biased data is worse for the performance of a traditional classifier. The COMPAS case shows better performance (despite not high accuracy) since it does not increase the bias as we raise β. We can argue that, in COMPAS, the resulting decision trees learned for each β are similar, which might explain its behavior. That being said, it would be ideal that the bias mitigation techniques lowered the bias below 0.15 without losing much of their accuracy. The same pattern of the Adult census can be observed in the Dutch census. As expected, removing the protected attribute contributes poorly in reducing bias

Table 6.1: Bias and Accuracy on biased datasets from Adult, COMPAS and Dutch data (classifier does not use the protected attribute)

β \ Dataset	Adult		COMPAS		Dutch	
	Bias.	Acc.	Bias.	Acc.	Bias.	Acc.
0.00	0.165	0.842	0.149	0.674	0.379	0.827
0.25	0.275	0.829	0.149	0.672	0.377	0.827
0.50	0.584	0.729	0.143	0.672	0.405	0.822
0.75	0.586	0.728	0.139	0.672	0.505	0.790
1.00	0.600	0.723	0.144	0.675	0.675	0.727

because other attributes could be highly correlated with the protected one.

Table 6.2: Bias and Accuracy for data pre-processing techniques trained on artificially generated datasets learned from adult census.

	β \ Tech.	Massaging		Reweighting		Unif. Sampling		Pref. Sampling	
		Bias	Acc.	Bias	Acc.	Bias	Acc.	Bias	Acc.
Adult	0.00	0.034	0.827	0.103	0.836	0.098	0.835	-0.002	0.814
	0.25	0.023	0.814	0.116	0.835	0.118	0.833	-0.034	0.808
	0.50	0.012	0.754	0.234	0.801	0.240	0.796	0.049	0.785
	0.75	0.066	0.719	0.463	0.707	0.462	0.705	0.096	0.757
	1.00	0.013	0.689	0.472	0.690	0.471	0.689	0.411	0.671
COMPAS	0.00	0.130	0.674	0.073	0.671	0.076	0.673	0.101	0.667
	0.25	0.121	0.672	0.063	0.669	0.073	0.671	0.086	0.665
	0.50	0.100	0.672	0.053	0.671	0.053	0.671	0.066	0.664
	0.75	0.083	0.672	0.049	0.668	0.052	0.669	0.050	0.664
	1.00	0.102	0.676	0.055	0.672	0.053	0.672	0.070	0.668
Dutch	0.00	0.113	0.787	0.146	0.814	0.150	0.814	0.053	0.785
	0.25	0.014	0.769	0.149	0.815	0.149	0.814	0.029	0.779
	0.50	-0.036	0.754	0.153	0.815	0.152	0.814	0.004	0.770
	0.75	-0.093	0.727	0.152	0.812	0.152	0.811	-0.020	0.757
	1.00	-0.143	0.696	0.450	0.714	0.214	0.780	-0.014	0.737

Table 6.2 shows Bias vs. Accuracy results for different pre-processing techniques. We can see on this table that most techniques increase bias when subject to a more biased training set. Re-weighting and Uniform Sampling behave very similarly (this behavior is held on other datasets). Preferential sampling has a steady decrease in accuracy and an increase in bias, but its result for 100% bias is quite different. We hypothesize that, when the dataset becomes entirely biased, its results are not significant anymore, but we include them for the sake of providing an upper bound.

Table 6.3 presents results for Auditor at repair levels 0.20, 0.50, and 0.80. On **Adult**, we see it works well for lower β, but it performs poorly in more biased scenarios. It is worth noting that scenarios with very high bias are less realistic, which means that Auditor's performance may be considered satisfactory at some extent because it has

minimal losses concerning accuracy. When the repair level is set to 0.80, Auditor keeps the accuracy and bias at the same levels for every β with low deviation, except when maximum artificial bias is inserted.

Table 6.3: Bias and Accuracy for Auditor technique.

	Tech / β	Repair 0.20		Repair 0.50		Repair 0.80	
		Bias.	Acc.	Bias.	Acc.	Bias.	Acc.
Adult	0.00	0.157	0.841	0.103	0.827	0.028	0.798
	0.25	0.225	0.835	0.139	0.827	0.024	0.795
	0.50	0.540	0.747	0.285	0.819	0.016	0.788
	0.75	0.592	0.727	0.590	0.726	0.010	0.783
	1.00	0.604	0.723	0.602	0.723	0.331	0.749
COMPAS	0.00	0.156	0.673	0.157	0.673	0.150	0.674
	0.25	0.155	0.674	0.165	0.674	0.163	0.674
	0.50	0.157	0.671	0.155	0.671	0.145	0.672
	0.75	0.148	0.671	0.165	0.672	0.145	0.670
	1.00	0.152	0.673	0.163	0.673	0.157	0.668
Dutch	0.00	0.356	0.827	0.286	0.823	0.252	0.819
	0.25	0.348	0.826	0.273	0.823	0.257	0.820
	0.50	0.355	0.822	0.261	0.821	0.251	0.818
	0.75	0.428	0.797	0.347	0.796	0.305	0.807
	1.00	0.620	0.729	0.465	0.760	0.000	0.477

One should choose between techniques based on the application. If a method that removes the bias entirely is needed, then it should be interesting to check out Massaging. At zero artificial bias inserted, Preferential Sampling did offer the highest accuracy with almost no bias, but in more biased scenarios, it did not sustain its performance.

On **COMPAS**, Auditor performed similarly for each one of the repairing levels but did not entirely remove the bias. The results consistency is also demonstrated by the similarity of the decision trees generated by J48 algorithm at every repairing level. In **Dutch** census scenario, Auditor kept the accuracy around the 80%, but it did not prevent the bias, so the other pre-processing techniques performed better regarding bias mitigation. However, it maintains the best accuracy overall. It can be observed that the 0.80 repair level performs slightly better than the repair level at 0.20 and 0.50.

In Table 6.4 we present the results for Zafar's techniques. In particular, the technique described in [39] for fairness, accuracy, and positive misclassification constraints. We can notice in the **Adult** scenario that the fairness constraint indeed keeps the bias around the same level at the cost of accuracy when the β increases. On the other hand, for accuracy constraints, it keeps the accuracy around the same level, and the bias increases with higher β levels. An interesting result can be observed when the positive misclassification constraint is applied, it causes lower bias levels but at a great cost of accuracy. In **COMPAS** case, Zafar's techniques tried to enforce less bias but also decreased accuracy.

As for the **Dutch** scenario, Zafar's techniques achieved competitive accuracy compared to pre-processing techniques and decreased accuracy. It provided a trade-off between accuracy and bias.

Table 6.4: Discrimination and Accuracy for Zafar's in-processing techniques trained on artificially generated datasets learned from adult census.

	Tech / β	Acc. cons.		Fair. cons.		Misclass.	
		Bias.	Acc.	Bias.	Acc.	Bias.	Acc.
Adult	0.00	0.128	0.837	-0.018	0.791	-0.066	0.590
	0.25	0.156	0.829	-0.032	0.773	-0.016	0.571
	0.50	0.181	0.821	-0.046	0.760	-0.015	0.530
	0.75	0.205	0.814	-0.046	0.747	-0.003	0.503
	1.00	0.258	0.813	-0.032	0.743	0.063	0.528
COMPAS	0.00	0.010	0.631	0.032	0.629	0.020	0.617
	0.25	0.014	0.630	0.014	0.628	0.029	0.615
	0.50	0.005	0.628	-0.002	0.622	0.022	0.610
	0.75	0.005	0.628	-0.027	0.620	0.040	0.610
	1.00	0.004	0.626	-0.043	0.615	0.033	0.607
Dutch	0.00	0.108	0.806	-0.031	0.746	-0.020	0.695
	0.25	0.091	0.801	-0.003	0.765	-0.059	0.647
	0.50	0.061	0.792	0.044	0.785	-0.105	0.596
	0.75	-0.025	0.762	0.127	0.812	-0.090	0.558
	1.00	-0.147	0.719	0.252	0.820	0.032	0.705

6.2 Bias vs. Accuracy graphs

Figure 6.1, Figure 6.2, Figure 6.3 and Figure 6.4 represent the bias versus accuracy of each dataset when no technique is used. Gray-scale dots depict how the technique performs at a specific β according to a scale on the right. We measure bias, or statistical parity difference, (x-axis) versus accuracy (y-axis). Each dot is the performance of the technique on an artificially generated dataset given these metrics. We included the results for German dataset on Figure 6.4.

In these graphs, the more concentrated the plots are next to 100% accuracy and 0 bias, the better the model has performed since it kept its consistency along with the biased samples. We also plotted the standard deviation (dotted lines) of every result, as we have sampled 30 different biased datasets for every level of bias. Figure 6.5, Figure 6.6, Figure 6.7, Figure 6.8, Figure 6.9 and Figure 6.10 compile the results regarding the dif-

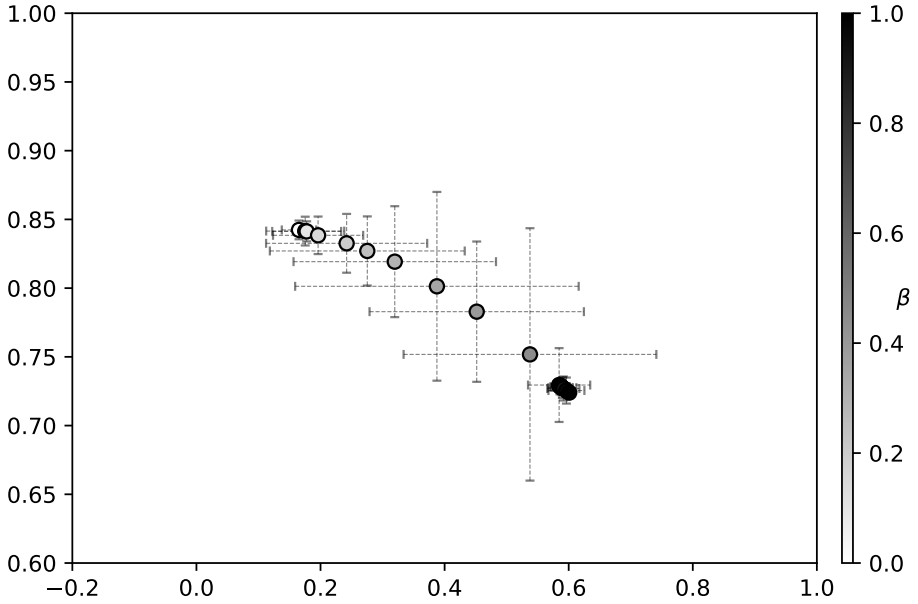


Figure 6.1: Bias vs. Accuracy on artificially generated datasets learned from adult census (the classifier does not use the protected attribute).

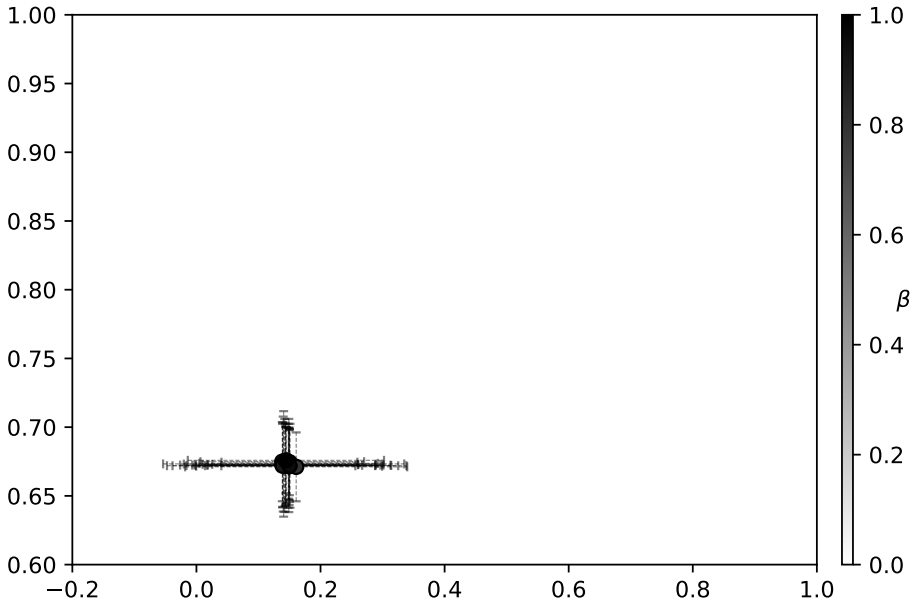


Figure 6.2: Bias vs. Accuracy on artificially generated datasets learned from COMPAS recidivism data (the classifier does not use the protected attribute).

ferent techniques that we are benchmarking on the adult census dataset. They present the results for pre-processing techniques and Auditor with repair levels of 0.20 and 0.80.

The most intriguing result is for the Massaging technique. Like Table 6.2, Figure 6.5 show that it has lower bias for lower and higher β, but when we introduce moderate β, it increases the resulting bias. Massaging works by defining a decision boundary between positive and negative outcomes through a Naive Bayes classifier, then it selects

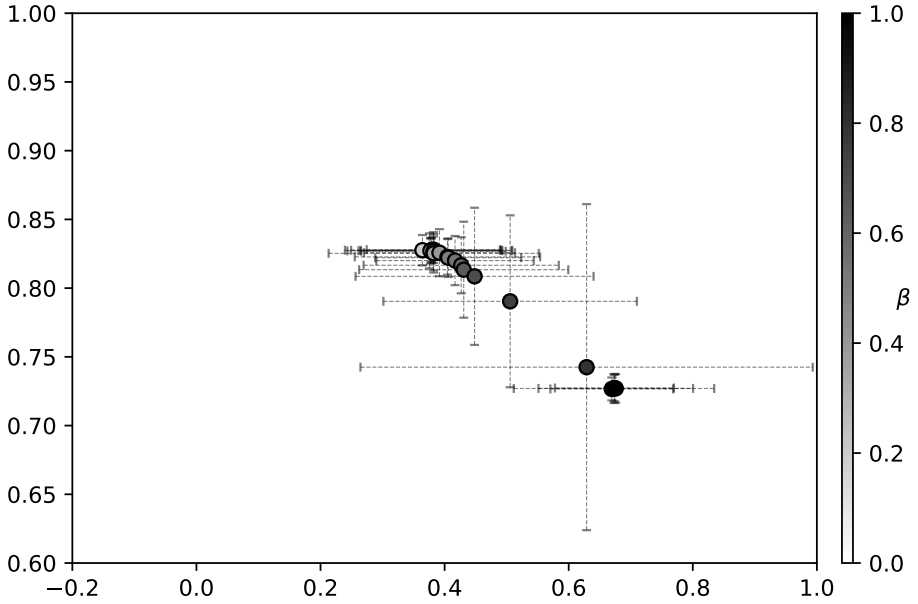


Figure 6.3: Bias vs. Accuracy on artificially generated datasets learned from Dutch census (the classifier does not use the protected attribute).

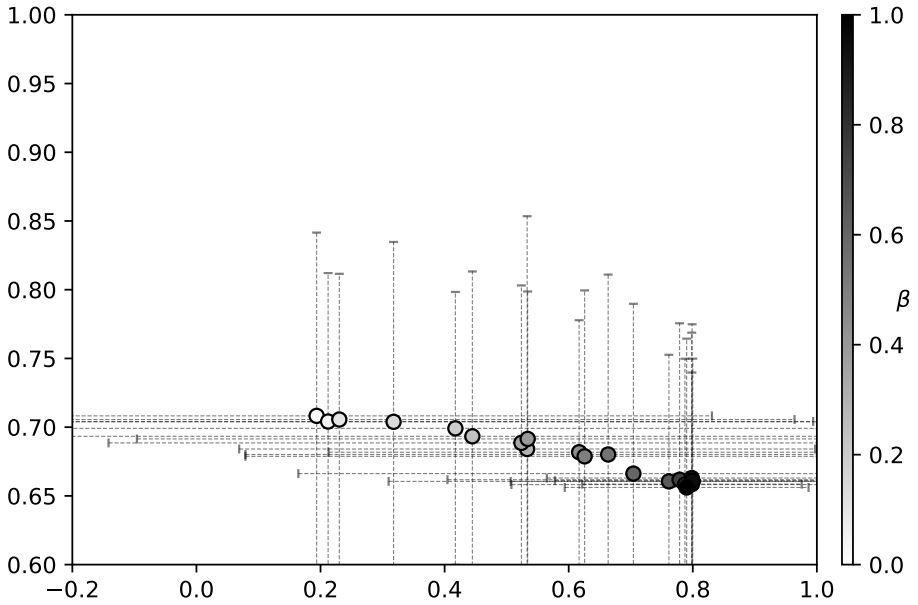


Figure 6.4: Bias vs. Accuracy on artificially generated datasets learned from German credit data (the classifier does not use the protected attribute).

points from the dataset and changes their labels in order to balance the ratio of positives and negatives between the two classes of the sensitive attribute, thus, forcing the deprived group to have more positive outcomes and the favored group to have more negative outcomes.

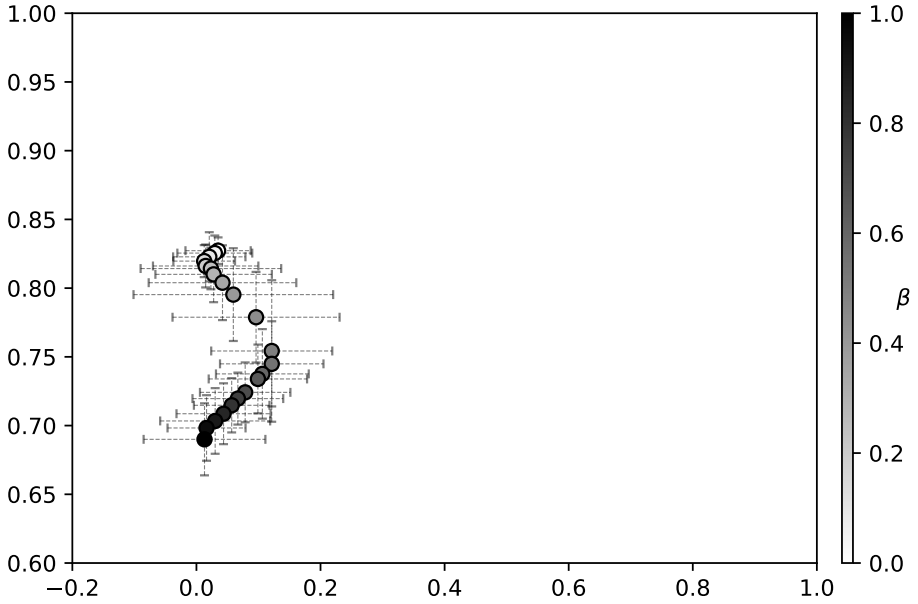


Figure 6.5: Bias vs. Accuracy for Massaging technique trained on artificially generated datasets learned from adult census.

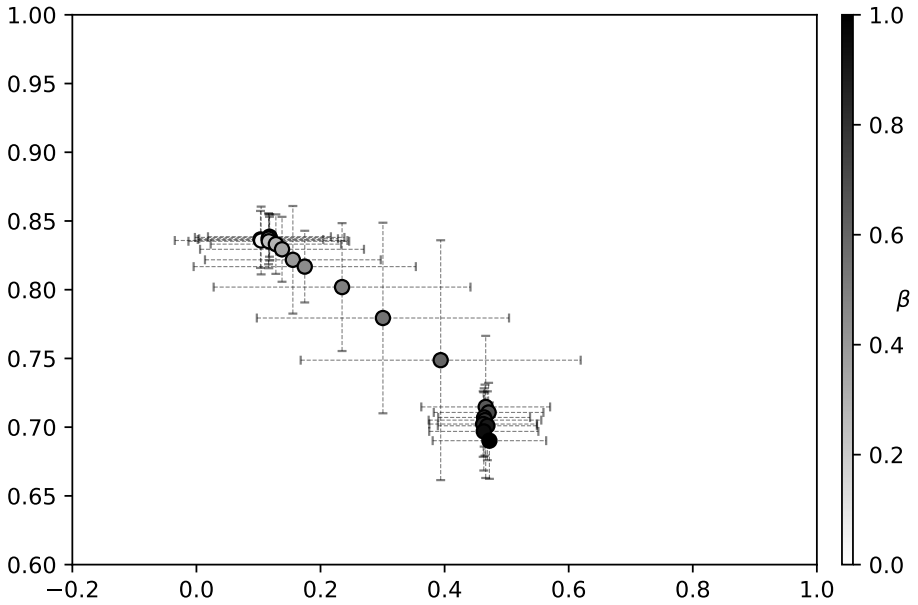


Figure 6.6: Bias vs. Accuracy for Re-weighting technique trained on artificially generated datasets learned from adult census.

6.3 Overestimation vs. Underestimation tables

Table 6.5 shows results on overestimation and underestimation for each dataset when no technique is applied. It is important to consider these values as upper bounds, which means that the objective of each technique is to reduce these (absolute) values. We

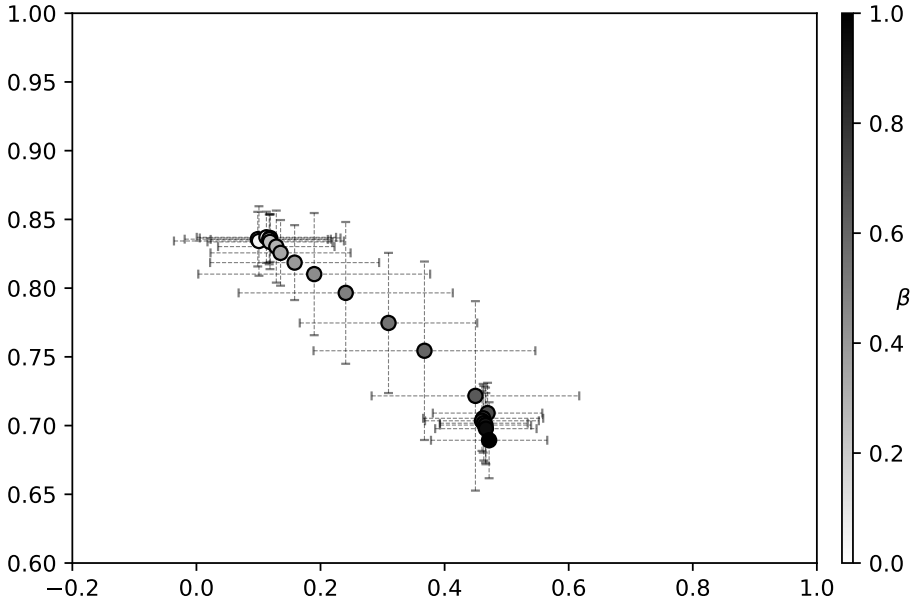


Figure 6.7: Bias vs. Accuracy for Uniform Sampling technique trained on artificially generated datasets learned from adult census.

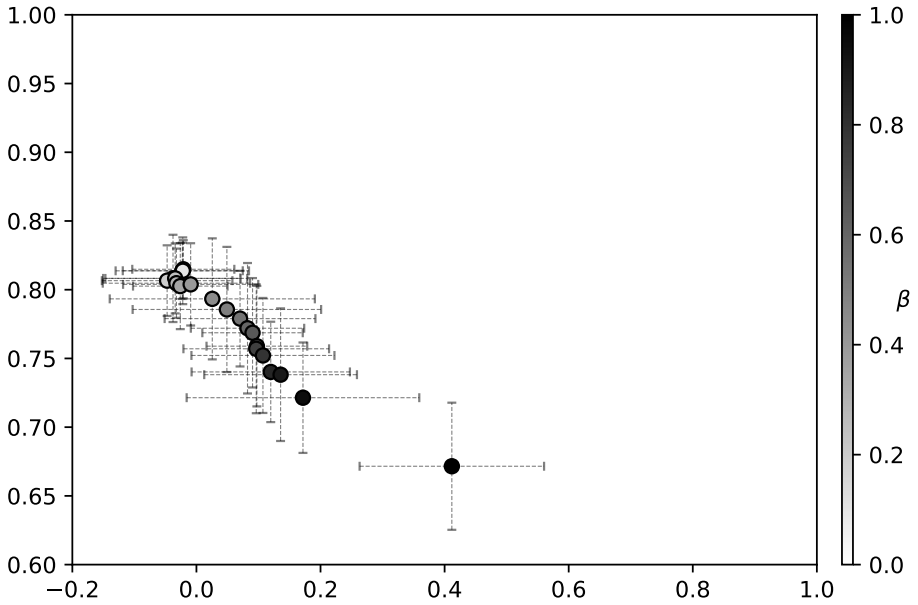


Figure 6.8: Bias vs. Accuracy for Preferential Sampling technique trained on artificially generated datasets learned from adult census.

can see that Adult and Dutch have high overestimation and underestimation at higher values of β. Curiously, COMPAS keeps the values close. These results suggest that for COMPAS higher β may have little influence on the resulting overestimation and underestimation, which is also explained by the fact that the decision trees generated by the classifier are similar regardless the β.

Table 6.6 shows the results for overestimation and underestimation for data pre-

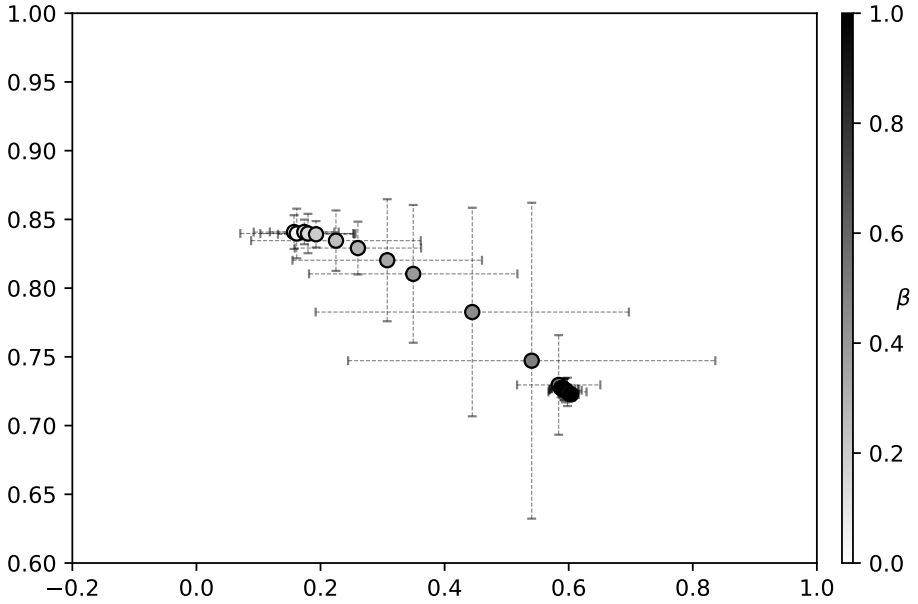


Figure 6.9: Bias vs. Accuracy data Auditor technique at 0.2 repair level trained on artificially generated datasets learned from adult census.

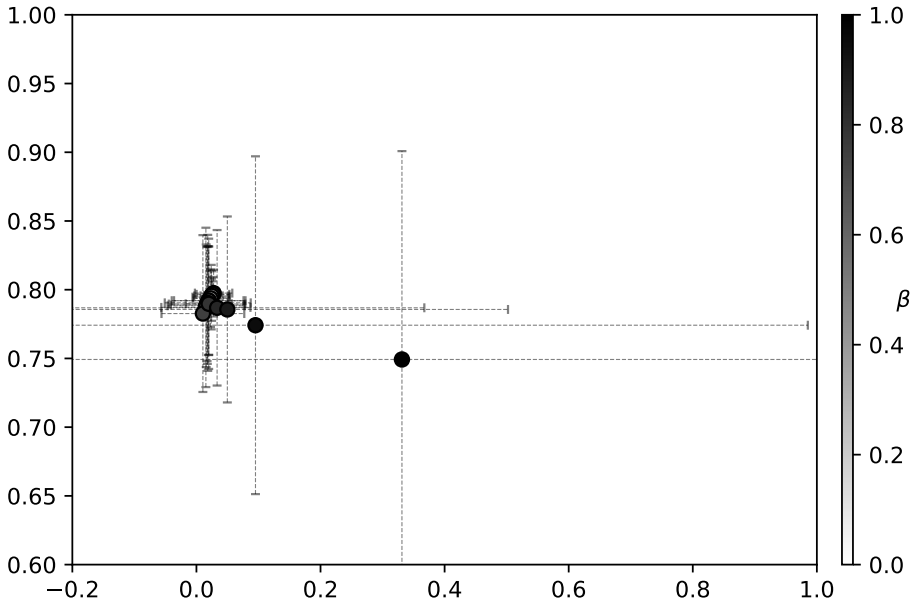


Figure 6.10: Bias vs. Accuracy data Auditor technique at 0.8 repair level trained on artificially generated datasets learned from adult census.

processing techniques. For **Adult**, Massaging reduces overestimation and underestimation in more biased scenarios. It is also interesting to note that almost every result of Massaging was a negative value. This technique keeps overestimation close to 0, meaning that it balances false positives between the two groups. The underestimation has slightly higher negative values, but these values get closer to 0 as we insert bias, which suggests that the technique underestimates the favored group for lower β, but it manages to balance

Table 6.5: Overestimation and underestimation for Adult, COMPAS and Dutch (classifier does not use the protected attribute).

β \ Dataset	Adult		COMPAS		Dutch	
	Over.	Under.	Over.	Under.	Over.	Under.
0.00	0.069	0.101	0.148	0.129	0.282	0.113
0.25	0.150	0.393	0.149	0.128	0.278	0.114
0.50	0.476	0.691	0.144	0.123	0.291	0.173
0.75	0.479	0.706	0.144	0.116	0.323	0.415
1.00	0.479	0.817	0.136	0.126	0.355	0.866

the underestimation between groups under higher β. Re-weighting and Uniform Sampling performed similarly by increasing both overestimation and underestimation, although they exhibit high deviation on both. Preferential Sampling has hard to predict behavior; it has low overestimation, and its underestimation started very negative and became closer to 0, except for the last biased sample. For **COMPAS**, we can see that Massaging and Preferential Sampling decrease underestimation and overestimation at higher β. Re-weighting and Uniform Sampling kept both underestimation and overestimation closer to 0, which is desirable. **Dutch** scenario suggests that Massaging and Preferential Sampling work similarly by inserting negative overestimation and underestimation when we increase β. Re-weighting and Uniform Sampling did a great job of keeping the values close (except for the case of higher β on Re-weighting).

Table 6.6: Overestimation and underestimation for data pre-processing techniques.

	β \ Tech.	Massaging		Reweighting		Unif. Sampling		Pref. Sampling	
		Over.	Under.	Over.	Under.	Over.	Under.	Over.	Under.
Adult	0.00	-0.030	-0.270	0.029	-0.090	0.024	-0.097	-0.055	-0.391
	0.25	-0.044	-0.289	0.032	-0.049	0.033	-0.037	-0.077	-0.393
	0.50	0.051	-0.158	0.137	0.119	0.139	0.142	-0.008	-0.274
	0.75	-0.003	-0.171	0.373	0.431	0.368	0.436	0.036	-0.205
	1.00	-0.066	-0.119	0.350	0.695	0.349	0.700	0.295	0.627
COMPAS	0.00	0.110	0.115	0.029	0.071	0.038	0.068	0.061	0.100
	0.25	0.097	0.108	0.027	0.057	0.039	0.062	0.045	0.086
	0.50	0.062	0.092	0.008	0.052	0.018	0.054	0.010	0.072
	0.75	0.047	0.074	0.016	0.042	0.017	0.044	-0.003	0.054
	1.00	0.053	0.099	0.014	0.051	0.016	0.050	0.007	0.079
Dutch	0.00	-0.045	-0.084	0.002	-0.093	0.009	-0.087	-0.107	-0.150
	0.25	-0.153	-0.167	0.006	-0.091	0.006	-0.091	-0.136	-0.166
	0.50	-0.203	-0.206	0.012	-0.087	0.012	-0.088	-0.150	-0.193
	0.75	-0.234	-0.266	0.011	-0.085	0.012	-0.084	-0.136	-0.244
	1.00	-0.265	-0.302	0.087	0.666	0.028	0.078	-0.074	-0.265

Table 6.7 shows results for overestimation and underestimation of Auditor tech-

Table 6.7: Overestimation and underestimation for Auditor technique.

Tech / β		Repair 0.20		Repair 0.50		Repair 0.80	
		Over.	Under.	Over.	Under.	Over.	Under.
Adult	0.00	0.062	0.079	0.028	0.004	-0.001	-0.045
	0.25	0.101	0.315	0.051	0.143	-0.001	-0.057
	0.50	0.413	0.656	0.158	0.386	-0.002	-0.061
	0.75	0.482	0.703	0.482	0.706	-0.002	-0.072
	1.00	0.482	0.803	0.484	0.783	0.168	0.265
COMPAS	0.00	0.160	0.132	0.160	0.130	0.157	0.121
	0.25	0.162	0.131	0.165	0.133	0.181	0.130
	0.50	0.147	0.126	0.180	0.139	0.165	0.121
	0.75	0.152	0.124	0.172	0.136	0.163	0.127
	1.00	0.142	0.131	0.172	0.142	0.155	0.127
Dutch	0.00	0.251	0.093	0.176	0.037	0.142	0.007
	0.25	0.238	0.090	0.146	0.023	0.136	0.011
	0.50	0.231	0.117	0.135	0.029	0.116	0.008
	0.75	0.263	0.323	0.163	0.217	0.148	0.188
	1.00	0.289	0.802	0.198	0.511	0.179	0.395

nique for each dataset on repair levels at 0.20, 0.50 and 0.80. For **Adult**, every technique performed similarly to Re-weighting and Uniform Sampling except Auditor at 0.80 repairing level, which kept almost every value close to zero for both overestimation and underestimation. For **COMPAS**, Auditor did not quite improve the result if compared to no technique used. In general, this technique could not keep up with the other preprocessing techniques for this scenario. For **Dutch**, Auditor improved the consistency of its results when we increase the repairing level to 0.80 as it has nearly 50% less overestimation compared to a repair level of 0.20. It is not Pareto-dominated (which means that it is not outperformed on both metrics) by any technique and provides a competitive result in most cases.

6.4 Overestimation vs. Underestimation graphs

Figure 6.11, Figure 6.12, Figure 6.13 and Figure 6.14 represent the overestimation versus underestimation of each dataset when no technique is used. Just like Figure 6.1, gray-scale dots depict how the method performs on a specific bias level according to a scale on the right. We measure the overestimation (x-axis) versus underestimation (y-axis). Finally, Figure 6.11, Figure 6.12, Figure 6.13 and Figure 6.14 provide results about the overestimation and underestimation metrics for each dataset when no technique is

applied.

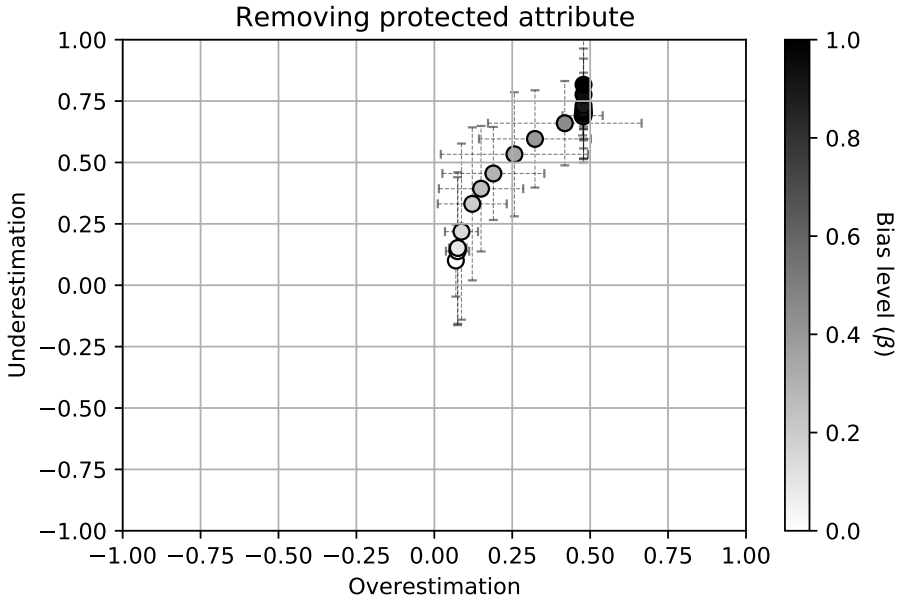


Figure 6.11: Overestimation and underestimation on artificially generated datasets learned from adult census (the classifier does not use the protected attribute).

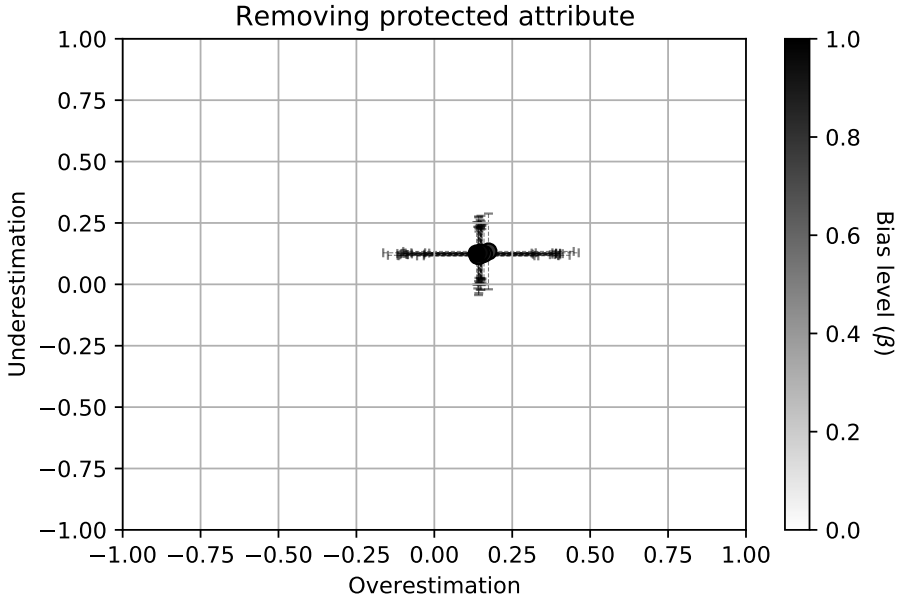


Figure 6.12: Overestimation and underestimation on artificially generated datasets learned from COMPAS recidivism data (the classifier does not use the protected attribute).

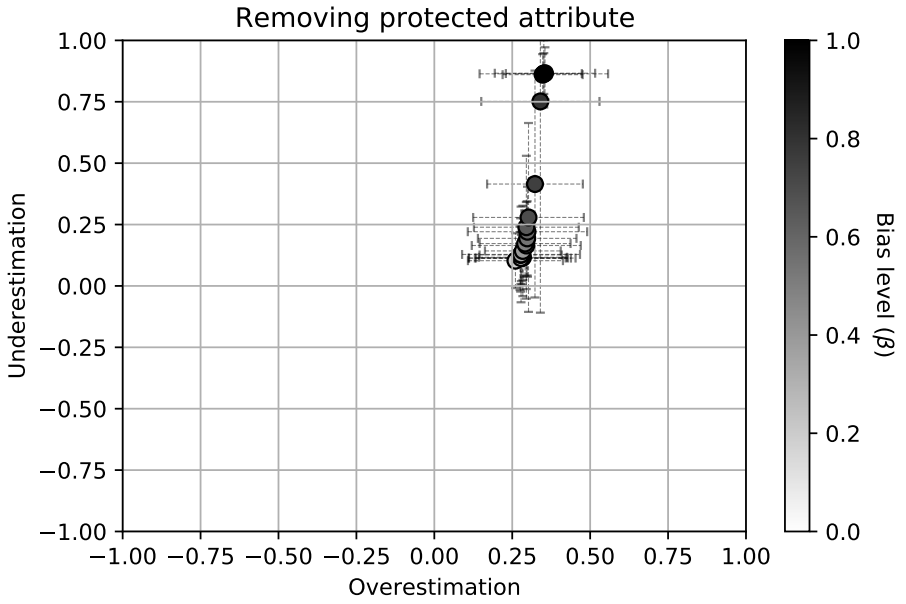


Figure 6.13: Overestimation and underestimation on artificially generated datasets learned from Dutch census (the classifier does not use the protected attribute).

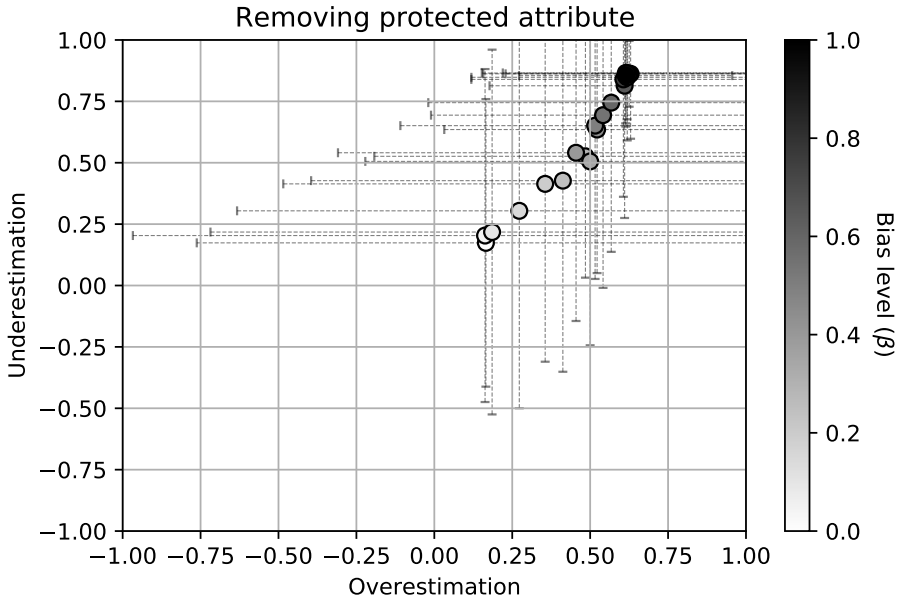


Figure 6.14: Overestimation and underestimation on artificially generated datasets learned from German credit data (the classifier does not use the protected attribute).

6.5 Compact Representation

One may find difficult to make sense of this amount of results, thus, we provide a compact yet informative representation as described in Section 4.6. Instead of having different points for each result we represent a whole curve as two points, or an arrow, in a

chart, as it can be seen in Figure 6.15 and Figure 6.16. The solid black arrow represents the result when no technique is applied.

Figure 6.15 depicts our methodology of result summarizing for Adult dataset. Notice that, regarding the behavior of Audit technique, as we increase the repair level its arrow goes further towards fairness prioritization although it degrades the quality. Massaging technique solely relies on improving fairness. COMPAS, in Figure 6.16, has a curious behavior for the Audit techniques as they somehow mimic the classifier in black; the results in Table 6.3 suggest that this is indeed the case.

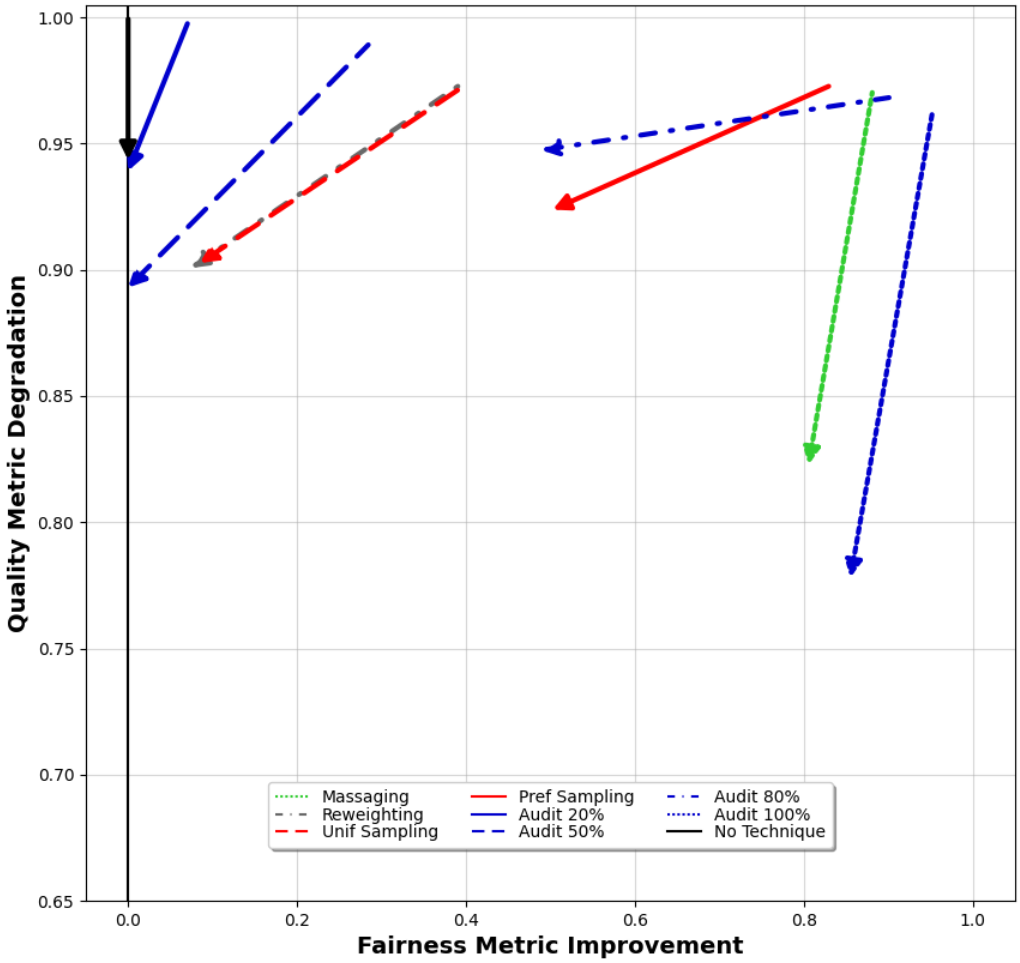


Figure 6.15: Summarized representation quality vs fairness: Adult census.

As for Dutch census results on Figure 6.17, we have the same behavior of Auditor technique observed in Adult census. However, in this case Preferential Sampling is the technique, among Kamiran et. al. [19] pre-processing techniques, that focuses on fairness improvement the most. German Dataset, as expected from a dataset with too few samples,

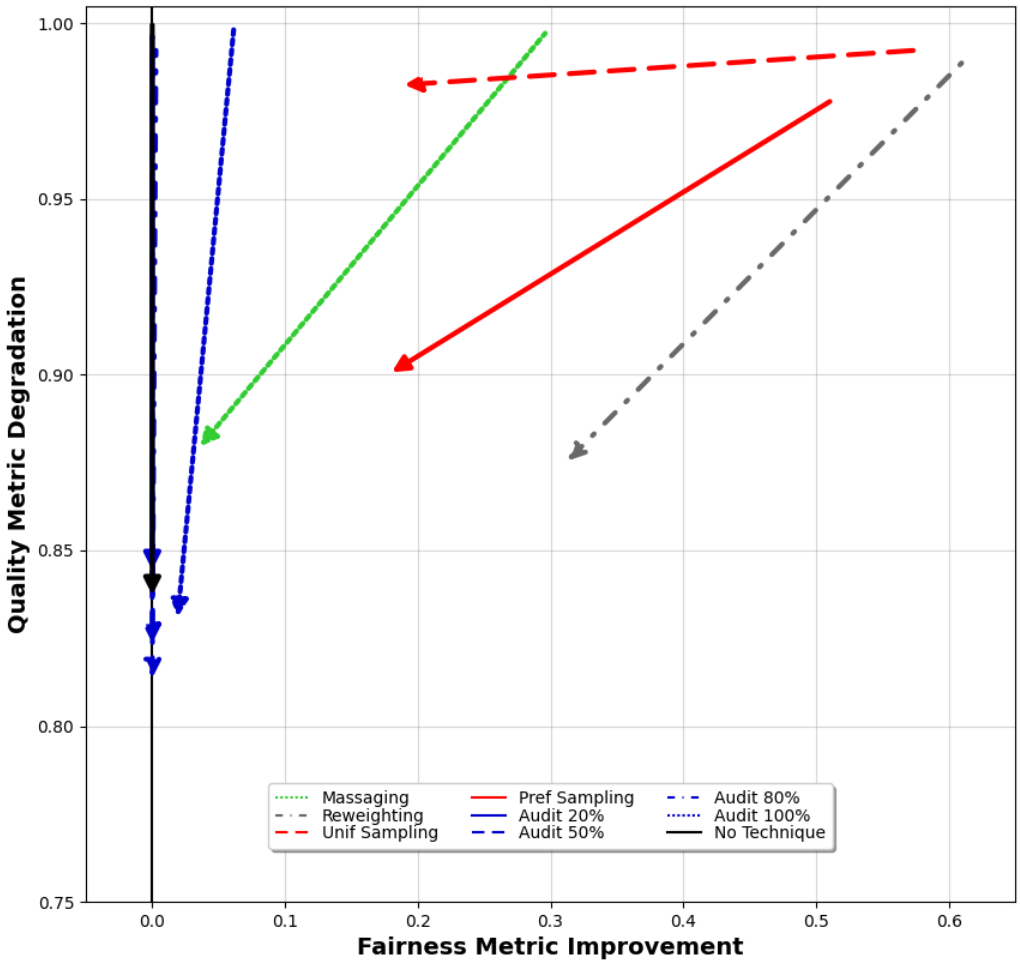


Figure 6.16: Summarized representation quality vs fairness: COMPAS.

presented the lowest reliability, thus, exhibiting higher norms for the arrows. Also, the techniques seem to prioritize quality rather than fairness except maybe for massaging, uniform sampling and higher Auditor's repairing levels. We can also notice that Auditor's behavior looks similar on Adult and Dutch.

6.6 Summary

As the main objective of this work is to provide a benchmark framework, we can argue that our results provide evidence that it is useful for comparing bias mitigation

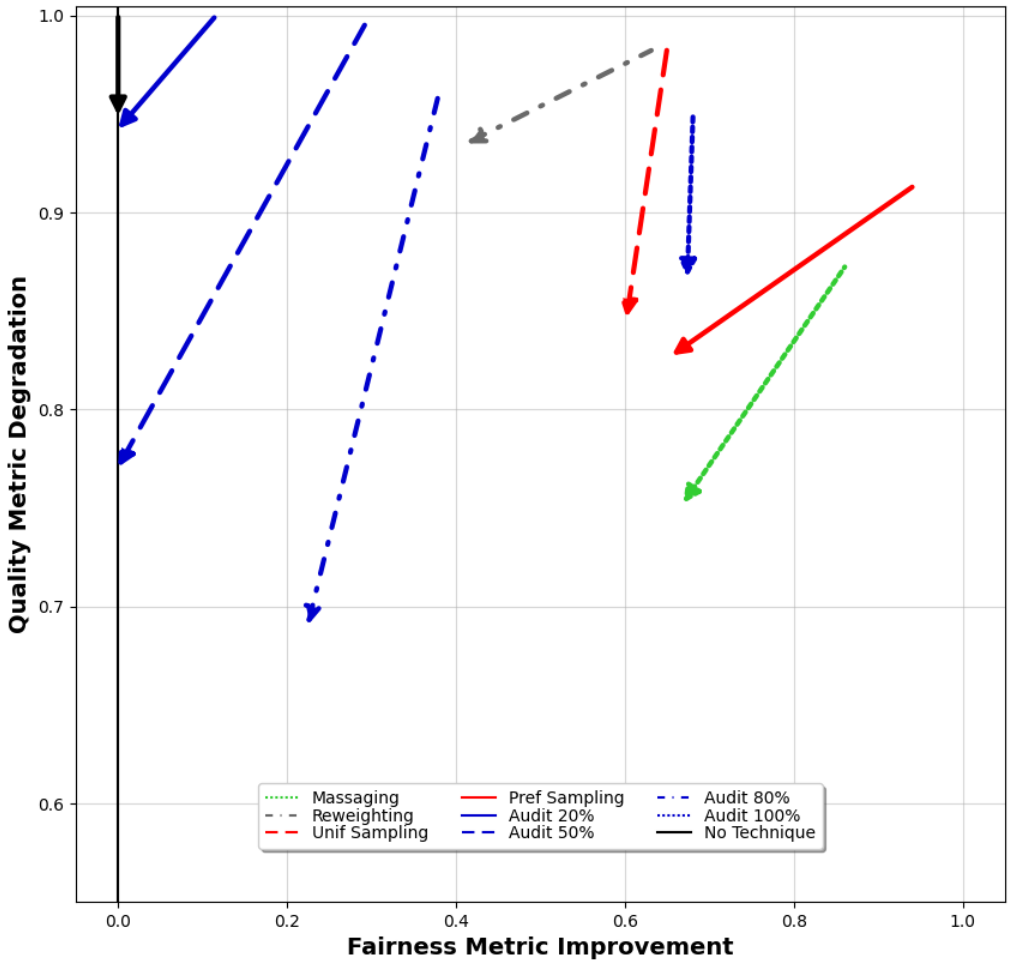


Figure 6.17: Summarized representation quality vs fairness: Dutch census.

techniques, since the decision maker may take misleading directions if they blindly select a technique without having a broader view of what the technique is capable of and what it is not. An example of that would be a technique that behaves poorly on very imbalanced scenarios. Another important point is the variety of applicable scenario. Our benchmark can be extended with low effort to different datasets, and also allow different classification quality or fair ML metrics, thus, enhancing the quality of its comparisons.

Despite the necessity of injecting bias in the data generation process in order to simulate real-world data, the framework becomes convenient if the decision maker has strong knowledge of the dataset that will be analyzed in the framework. On the Adult Census dataset, for example, we needed to remove some of its features so that we are able to increase the methodology's trustworthiness.

The process of choosing the arrow parameters in the summarized visualization,

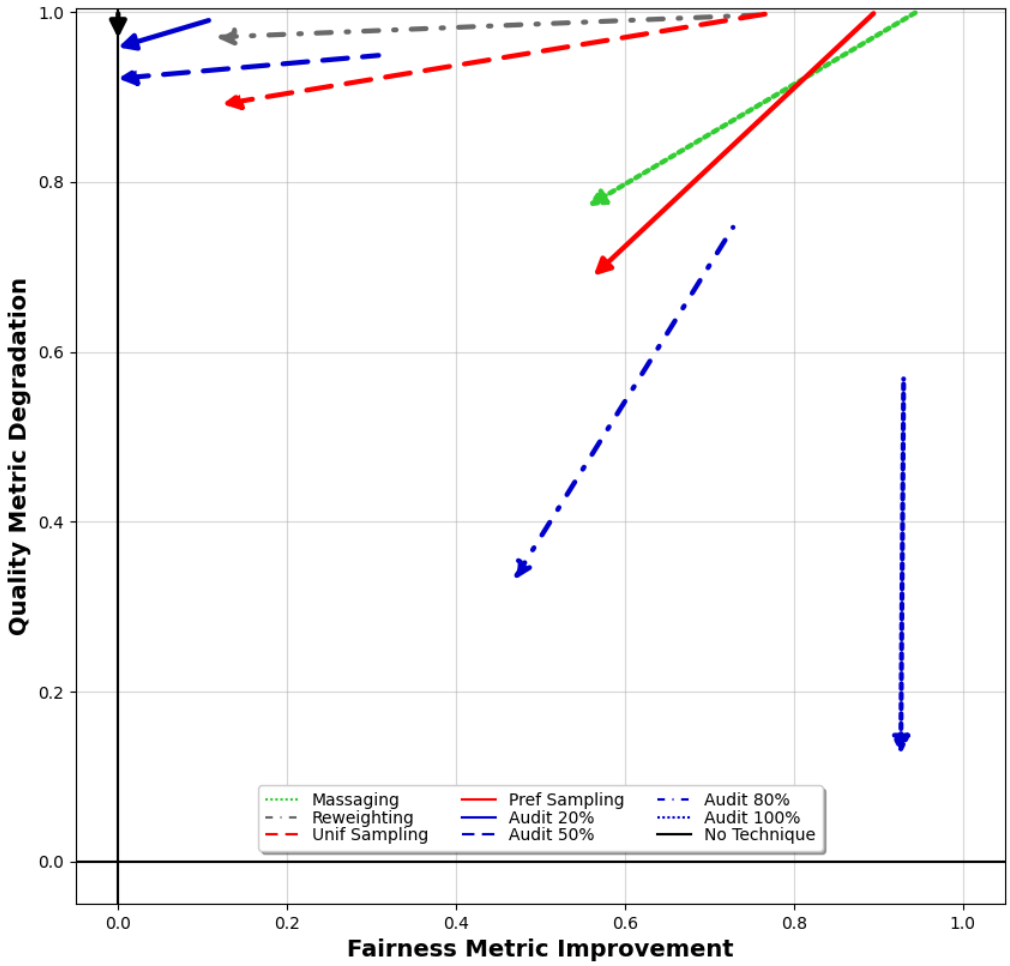


Figure 6.18: Summarized representation quality vs fairness: German credit.

that is θ and ω values, can also be customized to different approaches. Our summarized approach focuses primarily in bringing out the discussion on how to represent the results from Section 6.2 in a concise way.

We argue that our methodology can help researches on easily capture the nuances of their techniques that could only be capture on a daunting task of analyzing their results one by one. Although doing that is important, such methodology can help pointing out potential flaws directly instead of demanding an in-depth investigation on each result.

Chapter 7

Conclusions

This work introduces a novel benchmark framework for validating discrimination-aware data mining and machine learning models using systematically biased datasets generated from real-world data. The need for a benchmark is crucial due to the lack of common ground for the evaluation of techniques. We demonstrated the applicability and effectiveness of the proposed benchmark through a comparative assessment among several models on three relevant datasets.

7.1 Final Remarks

The value of our benchmark approach is apparent when we observe that it is hard to define the best technique for a given scenario. It is important is to decide which constraint is more relevant under a given scenario and then interpret the accuracy versus discrimination or over/under-estimation results in order to exercise the trade-off between metrics. Our framework makes this type of decision easier.

One must pay attention to the performance of the technique on a metric and the desired threshold of discrimination. One may want to decrease the bias metric due to law implications. Others may desire to reduce the misclassification rates in order to reduce the imbalance between classification errors between classes. As it has been argued in Zafar et al. [38], in the task of risk assessment, the false positives difference between classes (Underestimation) must be close to 0, since there are scenarios where it is more acceptable to classify as positive than wrongly classify as negative.

It is also important to explore different notions of fairness in the process of generating artificial datasets. Many studies on fairness metrics have been pursued in the past few years. It is important to increase the coverage of our methodology.

For now, we are not interested in just pointing out which technique is better than the other since it would require more parameter tuning. The main goal here is to propose a framework to verify the consistency of the technique and thus developing a model that

helps us to chose which one is more suitable for a specific scenario.

Finally, we have also introduced a methodology to provide a compact representation of the result which objective is to deliver the main intuitions of a result without compromising much of its information. The use cases listed in Section 4.7 point out some of the advantages of such methodology.

7.2 Future Directions

We intend to extend our work to provide full coverage of techniques of disparate impact and disparate mistreatment removal, as well as other fairness metrics. We know that other datasets fit discrimination-aware analysis. It would be interesting to perform similar studies on more diversified scenarios. We also managed to run an experiment for two classes of sensitive attributes. Using the adult census dataset, we generated artificial datasets with different levels of bias for gender and race. We can argue that it is hard to define which technique should be used in a given specific scenario. The idea of proposing a benchmark works by extensively comparing such techniques on different scenarios so that decision-makers may decide which technique presents the best trade-off according to their needs.

Then, we must define, of course, a metric that describes the performance of a given technique in this benchmark. Not only a simple raw score needs to be shown to a decision-maker, but also a diversity of scores must take place, so it should be possible to calculate a trade-off between the accuracy and the metrics then decide which technique is more suitable for the scenario. Different notions of fairness would also be useful for our analysis since decision-makers might have different fairness objectives.

References

[1] Philip Adler, Casey Falk, Sorelle A. Friedler, Gabriel Rybeck, Carlos Scheidegger, Brandon Smith, and Suresh Venkatasubramanian. Auditing black-box models for indirect influence. In *IEEE 16th International Conference on Data Mining, ICDM 2016, December 12-15, 2016, Barcelona, Spain*, pages 1–10, 2016.

[2] Julia Angwin, Jeff Larson, Lauren Kirchne, and Surya Mattu. Minority neighborhoods pay higher car insurance premiums than white areas with the same risk. propublica, april 5, 2017, 2017.

[3] Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. Machine bias: there's software used across the country to predict future criminals. and it's biased against blacks. propublica 2016, 2016.

[4] Rachel KE Bellamy, Kuntal Dey, Michael Hind, Samuel C Hoffman, Stephanie Houde, Kalapriya Kannan, Pranay Lohia, Jacquelyn Martino, Sameep Mehta, Aleksandra Mojsilović, et al. Ai fairness 360: An extensible toolkit for detecting and mitigating algorithmic bias. *IBM Journal of Research and Development*, 63(4/5):4–1, 2019.

[5] Irad Ben-Gal. Bayesian networks. *Encyclopedia of statistics in quality and reliability*, 1, 2008.

[6] Francesco Bonchi, Sara Hajian, Bud Mishra, and Daniele Ramazzotti. Exposing the probabilistic causal structure of discrimination. *I. J. Data Science and Analytics*, 3(1):1–21, 2017.

[7] Toon Calders and Sicco Verwer. Three naive bayes approaches for discrimination-free classification. *Data Min. Knowl. Discov.*, 21(2):277–292, 2010.

[8] Flavio Calmon, Dennis Wei, Bhanukiran Vinzamuri, Karthikeyan Natesan Ramamurthy, and Kush R Varshney. Optimized pre-processing for discrimination prevention. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 3992–4001. Curran Associates, Inc., 2017.

[9] Rodrigo L. Cardoso, Wagner Meira Jr., Virgílio A. F. Almeida, and Mohammed J. Zaki. A framework for benchmarking discrimination-aware models in machine learning. In Vincent Conitzer, Gillian K. Hadfield, and Shannon Vallor, editors, *Proceed-*

ings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society, AIES 2019, Honolulu, HI, USA, January 27-28, 2019, pages 437–444. ACM, 2019.

[10] Simon Caton and Christian Haas. Fairness in machine learning: A survey. 2020.

[11] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*, pages 214–226. ACM, 2012.

[12] Brasil Supremo Tribunal Federal et al. *Constituição da república federativa do Brasil*. Supremo Tribunal Federal, 1988.

[13] 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, Sydney, NSW, Australia, August 10-13, 2015*, pages 259–268, 2015.

[14] Sorelle A Friedler, Sonam Choudhary, Carlos Eduardo Scheidegger, Evan P Hamilton, Suresh Venkatasubramanian, and Derek Roth. A comparative study of fairness-enhancing interventions in machine learning. In *2019 ACM Conference on Fairness, Accountability, and Transparency, FAT* 2019*, pages 329–338. Association for Computing Machinery, Inc, 2019.

[15] Sara Hajian, Francesco Bonchi, and Carlos Castillo. Algorithmic bias: From discrimination discovery to fairness-aware data mining. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 2125–2126, 2016.

[16] Sara Hajian, Anna Monreale, Dino Pedreschi, Josep Domingo-Ferrer, and Fosca Giannotti. Fair pattern discovery. In *Symposium on Applied Computing, SAC 2014, Gyeongju, Republic of Korea - March 24 - 28, 2014*, pages 113–120, 2014.

[17] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The WEKA data mining software: an update. *SIGKDD Explorations*, 11(1):10–18, 2009.

[18] Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 3315–3323, 2016.

[19] Faisal Kamiran and Toon Calders. Data preprocessing techniques for classification without discrimination. *Knowl. Inf. Syst.*, 33(1):1–33, 2011.

[20] Faisal Kamiran, Toon Calders, and Mykola Pechenizkiy. Discrimination aware decision tree learning. In *ICDM 2010, The 10th IEEE International Conference on Data Mining, Sydney, Australia, 14-17 December 2010*, pages 869–874, 2010.

[21] Jon Kleinberg, Sendhil Mullainathan, and Manish Raghavan. Inherent trade-offs in the fair determination of risk scores. In Christos H. Papadimitriou, editor, *8th Innovations in Theoretical Computer Science Conference (ITCS 2017)*, volume 67 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 43:1–43:23. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2017.

[22] Jeff Larson, Surya Mattu, Lauren Kirchner, and Julia Angwin. How we analyzed the compas recidivism algorithm. *ProPublica (5 2016)*, 9, 2016.

[23] M. Lichman. UCI machine learning repository, 2013.

[24] David Liu, Zohair Shafi, William Fleisher, Tina Eliassi-Rad, and Scott Alfeld. Rawlsnet: Altering bayesian networks to encode rawlsian fair equality of opportunity. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, pages 745–755, 2021.

[25] Pranay K Lohia, Karthikeyan Natesan Ramamurthy, Manish Bhide, Diptikalyan Saha, Kush R Varshney, and Ruchir Puri. Bias mitigation post-processing for individual and group fairness. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2847–2851. IEEE, 2019.

[26] Karima Makhlouf, Sami Zhioua, and Catuscia Palamidessi. Survey on causal-based machine learning fairness notions. 2020.

[27] Koray Mancuhan and Chris Clifton. Combating discrimination using bayesian networks. *Artif. Intell. Law*, 22(2):211–238, 2014.

[28] Dimitris Margaritis. Learning bayesian network model structure from data, 2003.

[29] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *ACM Comput. Surv.*, 54(6), jul 2021.

[30] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[31] Radhakrishnan Nagarajan, Marco Scutari, and Sophie Lèbre. Bayesian networks in r. *Springer*, 122:125–127, 2013.

[32] Dino Pedreschi, Salvatore Ruggieri, and Franco Turini. Discrimination-aware data mining. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, Nevada, USA, August 24-27, 2008*, pages 560–568, 2008.

[33] Dino Pedreschi, Salvatore Ruggieri, and Franco Turini. Measuring discrimination in socially-sensitive decision records. In *Proceedings of the SIAM International Conference on Data Mining, SDM 2009, April 30 - May 2, 2009, Sparks, Nevada, USA*, pages 581–592, 2009.

[34] Dino Pedreschi, Salvatore Ruggieri, and Franco Turini. A study of top-k measures for discrimination discovery. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing*, pages 126–131. ACM, 2012.

[35] J Ross Quinlan. *C4. 5: programs for machine learning*. Elsevier, 2014.

[36] Salvatore Ruggieri, Dino Pedreschi, and Franco Turini. Data mining for discrimination discovery. *TKDD*, 4(2):9:1–9:40, 2010.

[37] Till Speicher, Hoda Heidari, Nina Grgic-Hlaca, Krishna P. Gummadi, Adish Singla, Adrian Weller, and Muhammad Bilal Zafar. A unified approach to quantifying algorithmic unfairness: Measuring individual &group unfairness via inequality indices. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '18, page 2239–2248, New York, NY, USA, 2018. Association for Computing Machinery.

[38] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez-Rodriguez, and Krishna P. Gummadi. Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment. In *Proceedings of the 26th International Conference on World Wide Web, WWW 2017, Perth, Australia, April 3-7, 2017*, pages 1171–1180, 2017.

[39] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rogriguez, and Krishna P Gummadi. Fairness constraints: Mechanisms for fair classification. In *Artificial Intelligence and Statistics*, pages 962–970. PMLR, 2017.

[40] Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. Mitigating unwanted biases with adversarial learning. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pages 335–340. ACM, 2018.

[41] Indre Zliobaite. Measuring discrimination in algorithmic decision making. *Data Min. Knowl. Discov.*, 31(4):1060–1089, 2017.

[42] Indre Zliobaite, Faisal Kamiran, and Toon Calders. Handling conditional discrimination. In *11th IEEE International Conference on Data Mining, ICDM 2011, Vancouver, BC, Canada, December 11-14, 2011*, pages 992–1001, 2011.