ASHESI UNIVERSITY

REFLECT: TEACHING MACHINE LEARNING MODELS TO RECONSIDER THEIR BIASES

UNDERGRADUATE THESIS

B.Sc. Computer Science

Maame Yaa Osei

2020

ASHESI UNIVERSITY

REFLECT: TEACHING MACHINE LEARNING MODELS TO RECONSIDER THEIR BIASES

UNDERGRADUATE THESIS

Thesis submitted to the Department of Computer Science, Ashesi University in partial fulfilment of the requirements for the award of Bachelor of Science degree in Computer Science.

Maame Yaa Osei

2020

DECLARATION

I hereby declare that this Undergraduate Thesis is the result of my own original work and that no part of it has been presented for another degree in this university or elsewhere.

Candidate's Signature:

.....

Candidate's Name:

.....

Date:

.....

I hereby declare that preparation and presentation of this Undergraduate Thesis were supervised in accordance with the guidelines on supervision of Undergraduate Thesis laid down by Ashesi University.

Supervisor's Signature:

.....

Supervisor's Name:

.....

Date:

.....

Acknowledgements

Many individuals have dedicated great amounts of time, thought and effort to see the work of this thesis come to fruition. As such it is almost impossible for me to acknowledge them all here. However, I would firstly like to thank my supervisor, Mr. Dennis Asamoah-Owusu for his great help in refining my ideas and his motivation. Though we found ourselves in different time zones and faced some roadblocks along the way, Mr. Asamoah-Owusu's good knowledge on the subject matter and genuine interest went a long way to contribute to the success of this work.

A very big thank you goes out to my biological and selected family that made sure I was comfortable physically, mentally and emotionally to embark on this academic journey. I appreciate every thought, prayer and word of encouragement.

My sincere gratitude also goes out to Sam Moorhouse for buying me the book that inspired me to embark on this project. To my friends Jean Sebastien Dovonon and Ariel Woode, thank you for being great peer tutors and pushing me to be creative and resilient. To Maxwell Aladago and Mawuli Adjei, thank you for the dialogues that helped me flesh out the ideas of this paper's work.

Abstract

Machine learning models have gained prevalence in the world we live in today. Currently, they are employed in different fields to automate the decision-making process of various institutions. With the increasing availability of data to train these models, machine learning models are adopted to solve complex classification problems that have the potential to affect people's lives both positively and negatively. Unfortunately, certain patterns of bias that are tied to the presence of socio-economic characteristics of individuals such as race, gender and income levels may exist in the data used in training these models. When such data is employed in creating predictive machine learning models, they go on to make decisions that go against certain classes of individuals in society and favour others. This paper elaborates on a method that improves the fairness of machine learning models by closing the disparity between the misclassification rates of predictions made for classes within a sensitive group under consideration. It achieves this by modifying the loss function of a classifier such that it considers the disparate mistreatment of people, based on their membership of a particular sensitive class. This is done by calculating for the gradient of the error between predictions made and the ground truth for each sensitive group (based on the sensitive feature taken under consideration that may contribute to unfair decision making). This gradient is then added to the gradient of the Cross-Entropy loss function of a Logistic Regression classifier. By including this modification to models' loss functions, it learns parameters that not only correctly predict recidivism, but also minimize disparate mistreatment. This experiment was able to close the disparity between the false positive rates of recidivism risk score predictions made on African Americans and individuals of other racial origin in a subset of the COMPAS Recidivism Dataset by 52% (from a difference of 0.25 to 0.12). It was able to do so with a test accuracy of 70.4%.

Keywords:

Fairness; bias; predictive machine learning; Logistic Regression

Table of Contents

DECLARATION . . . . . **i**

Acknowledgements . . . . . **ii**

Abstract . . . . . **iii**

1 Chapter 1: Introduction **1**

2 Chapter 2: Related Work **4**

2.1 Bias Testing in Machine Learning 4

2.2 Formalizations of Fairness 4

2.3 Auditing Machine Learning Models 5

2.4 Algorithm Transparency 6

2.5 Measures for Bias Correction in Machine Learning 7

2.6 Legal Work on Fairness in Machine Learning 7

3 Chapter 3: Methodology **8**

3.1 Requirement Analysis on Data 8

3.2 Dataset 9

3.3 Logistic Regression for Binary Classification 11

3.4 Approach 12

3.5 Technologies Employed 14

3.5.1 Google Colaboratory 14

3.5.2 Sci-kit Learn 15

3.5.3 NumPy 15

3.5.4 Altair Visualization 15

3.5.5 Matplotlib 15

4 Chapter 4: Methodology 2 - Implementation **17**

4.1 The Base Experiment 17

4.2 The Core Experiment 18

4.2.1 Gradient of Error Between Predictions and Ground Truth 19

4.2.1.1 Algorithm of Implementation of Core Experiment 21

4.2.2 Using Mean Squared Error of Misclassification Rates 21

5 Chapter 5: Results 23

5.1 Results of Base Experiment 23

5.2 Results of Core Experiment 26

5.2.1 $\beta_0 = 1$ and $\beta_1 = 0$ 26

5.2.2 $\beta_0 = 0.5$ and $\beta_1 = 0.5$ 28

5.2.3 $\beta_0 = 0.3$ and $\beta_1 = 0.7$ 29

6 Chapter 6: Conclusions and Further Work 32

6.1 Summary of Conclusions 32

6.2 Limitations 32

6.3 Suggestions for Further Work 33

List of Tables

3.1 Demonstration of Difference between Dependent Variable and Ground truth 9

4.1 Distribution of Training and Test data 17

4.2 Demonstration of Feature Table with Features (X_n) and Sensitive Feature (z) 19

4.3 Demonstration of Reconstructed Feature Table for $z = 0$ 20

4.4 Demonstration of Reconstructed Feature Table for $z = 0$ 20

5.1 Distribution of Classifier Predictions Across Sensitive Feature (z) 24

List of Figures

3.1 Flowchart of Base Experiment 13

3.2 Flowchart of Core Experiment 14

4.1 Results Achieved Using Mean Squared Error of Misclassification Rates . . 22

5.1 Results of Base Experiment 23

5.2 Confusion Matrices from ”Unfair” Classifier 25

5.3 Confusion Matrices from ”Fair” Classifier 26

5.4 Results of Core Experiment with $\beta_0 = 1$ and $\beta_1 = 0$ 27

5.5 Predictions of Core Experiment with $\beta_0 = 1$ and $\beta_1 = 0$ 27

5.6 Results of Core Experiment with $\beta_0 = 0.5$ and $\beta_1 = 0.5$ 28

5.7 Predictions of Core Experiment with $\beta_0 = 0.5$ and $\beta_1 = 0.5$ 29

5.8 Results of Core Experiment with $\beta_0 = 0.3$ and $\beta_1 = 0.7$ 30

5.9 Predictions of Core Experiment with $\beta_0 = 0.3$ and $\beta_1 = 0.7$ 31

Chapter 1: Introduction

Currently, 2.7 zettabytes of data exist in our world. And by the year 2020, each person is estimated to be responsible for the production of 1.7 megabytes of data every ticking second [24]. With such large amounts of data being generated comes the opportunity to create powerful Artificial Intelligence systems and Machine Learning models to automate and solve a variety of problems in the world today. These problems range from the classification of customers' sentiments behind reviews on e-commerce platforms to national intelligence surveillance of individuals for terrorist alerts. Here, grave danger arises in the possibilities these machine learning solutions present, because machine learning models learn exactly what they are taught.

In their attempt to make efficient generalizations of the data they learn, machine learning models encounter the problem of bias; the occurrence of prejudiced results due to wrong assumptions made in the process of learning [19]. Examples of such bias show up in the word embeddings of Natural Language Processing models which are more likely to associate gender neutral occupational titles such as "doctor" and "nurse" to the male and female gender respectively [6]. Such an error may come across as benign, but as we begin to trust these machine learning models to make more complex decisions, the consequences of their wrong classifications exhibit more gravity. In 2016, research by ProPublica on a commercial tool developed by Northpointe Inc. (now known as Equivalente) called COMPAS (Correctional Offender Management Profiling for Alternative Sanctions) which is used in some legal systems in the United States to predict recidivism (the likelihood of offenders committing crimes again) was nearly twice as likely to classify black defendants as higher risk when compared to its classification of their counterpart white defendants [12, 17]. And a tool developed by Amazon to automate the recruitment process of top companies was found to reject the applications of women to fields regarded as more technical; such as those within the areas of Science, Technology, Engineering and Mathematics (STEM)[8].

The biases that machine learning models demonstrate may arise for different reasons including but not limited to an insufficient scope of training data (sample bias), systematic distortion of values due to an error in a tool for data collection (measurement bias), bias in the algorithms behind these models (which have nothing to do with the data the models are trained on) and/or prejudice or learned bias, which is introduced as a result of stereotypes

that humans introduce into data used to train models. Regardless of the types of bias present within a given machine learning model, the need to address them presents itself in order to produce more accurate outcomes [7, 9, 22, 27].

As varied as sources of bias in data may be, they may be mitigated with due diligence. For example, one may simply ensure that tools of measurement are properly functional to mitigate measurement bias. With sample size bias, one may make a conscious effort to select a sample that represents its larger population well [22]. Regardless of the aforementioned efforts one may make, certain biases are ingrained in the very existence of features in data. These biases, referred to as learned biases or societal biases in some literature are those that are picked up by models as a result of passed on human biases. Although usually benign, learned biases target specific features that revolve around the socio-economic characteristics of human subjects represented in training data. Learned biases may occur through the presence of factors such as the age, gender, race and income levels of subjects represented in training data. When such bias exists in data, machine learning models begin to display and possibly perpetuate it. By its latent nature, learned biases are difficult to mitigate simply. However, it is necessary that they be evaluated because they may negatively affect human lives directly or indirectly.

This gives rise to the objective of this research paper; to mitigate learned biased in machine learning models. To achieve this, the misclassification of a model is used to penalize it in its training steps, to help it adjust for its biased outcomes and improve the general fairness of models.

Misclassifications refer to the errors that arise during a model's attempt to classify data. This paper is concerned with misclassifications that arise on the basis of a particular sensitive feature in training data. In the paper, misclassification is measured relative to additional data referred to as ground truth and included in the loss function of a Logistic Regression classifier such that the model begins to consider its unfair misclassifications as it learns during training. In detail, an error is computed as the difference between a model's predictions and the ground truth for each class within a specified sensitive feature. The gradient of this error is computed similarly to the gradient of the Logistic Regression Cross-Entropy loss function and added to the classifier's gradient descent function. This teaches the classifier to learn parameters that optimize for both correct and fair predictions. Weights are place on the classifier's correctness and fairness to analyze its behavior when they are

adjusted. The misclassification rates for each sensitive group are analyzed as the weights are adjusted to assess how well the model is able to address the disparity between them. This approach yielded positive results when an even weight was placed on correctness and fairness in the model's predictions as it was able to bring the misclassification rates of the different sensitive groups closer to each other. When more emphasis is placed on fairness than on correctness, the model is unable to learn appropriately to make neither correct nor fair predictions.

It is worth noting that the term bias (not to be confused with the statistical measure) and unfairness are used interchangeably in this paper.

Chapter 2: Related Work

2.1 Bias Testing in Machine Learning

Current efforts to test for the biases inherent within the data fed into training machine learning models are gaining ground. These efforts have approached the problem of testing and auditing bias in machine learning models from different perspectives. One of such approaches is the application of psycho-social metrics that are useful in assessing human bias. One such example of this is the Implicit Association Test (IAT) that is used in quantifying the prevalence of known biases (such as those of race and gender), inherent in human associations [3]. The IAT has been applied on corpora of text found on the web that have been used in Natural Language Processing models to give researchers a sense of the possible biases such models may perpetuate.

However, it is not enough to simply know the possibilities of bias existing within the data and the algorithms used in training these models. Also, such measures require qualitative research involving human interaction to be able to accurately define the nature of biases existing in the IAT. This requirement may not be easily fulfilled when dealing with raw data to be trained on.

2.2 Formalizations of Fairness

Seeing that the application of standard measures of assessing social bias to machine learning models may not prove feasible in application to real-word problems, researchers have coined formalizations of bias that may be checked for in machine learning, to allow the implementation of methods to mitigate them. Thus, machine learning models may be classified as exhibiting bias in the form of disparate treatment, disparate impact and/or disparate mistreatment [21, 22, 26, 27]. These measures are further explained as follows:

1. Disparate treatment: This type of bias is realized in a model if its prediction or classification of a given instance changes when a change is made to a sensitive feature. Take a model that scans through resumes to classify individuals as either qualified or unqualified for a given job. If for a given female individual it classifies as unqualified yet classifies as qualified when the gender of that individual is adjusted to male with all other features remaining equal, the model is said to exhibit disparate treatment.

2. Disparate impact: This type of bias is realized when there is a difference in the proportion of a model's various outcomes for the various sensitive groups under its consideration. Taking the example of the resume assessing model above, if it tends to predict black people as unqualified more than it does for white people it is said to exhibit disparate impact.

3. Disparate mistreatment: More aligned with the goal of this study is the idea of disparate mistreatment. Disparate mistreatment occurs when a model produces different proportions of accurate outcomes for the various sensitive groups under its consideration. To asses this kind of bias, the misclassification rates of models must be taken into consideration. Using the resume scanning model to demonstrate this, it would be said to exhibit disparate mistreatment if it is more likely to classify Hispanic individuals as unqualified when the ground truth of such individuals is qualified than it is to classify Caucasian individuals as unqualified when the ground truth of such individuals is qualified.

These definitions may be expressed mathematically depending on the use case at hand and enforced as constraints on predictive models.

2.3 Auditing Machine Learning Models

Researchers have taken different approaches to the task of evaluating bias and/or enforcing fairness in machine learning models. One such approach that is gaining recognition involves revealing the significance of the features fed into predictive models to assess the extent of models' discrimination. This is integrated into various machine learning libraries in the form of feature importance metrics. Using these metrics provide information on whether or not a potentially problematic sensitive feature has a high importance score (meaning, the model takes it into consideration in decision-making heavily). Another tool known as FairML developed in the Massachusetts Institute of Technology presents a solution for diagnosing this bias and how it affects the results these models present. The application of these tools on a given model generates a percentage that informs users on how much a given feature influences a model's decisions. In a nutshell, such machine learning auditing tools are able to inform its users on how much a model takes certain variables into consideration when making decisions [1]. These tools are said to be model-agnostic as they can be

applied on any machine learning model. In contrast to model-agnostic tools, certain models are able to explain their own outcomes. For example, decision rules can be extracted from decision trees for given outcomes [15].

Aequitas is another tool developed by researchers at the Center for Data Science and Public Policy in University of Chicago [21]. It enables users to audit data that they use to train models to generate reports on the types of bias (where bias indicates a breach in the aforementioned formalizations of fairness) that both the data and the model such data is fed exhibit. With Aequitas, users can upload data to a web toolkit, call on methods implemented in a Python library, or pass command line arguments to generate bias reports.

Tools used to audit machine learning models are beneficial in assessing the potential source of bias that may result from the models' outcomes. However, they are neither directly preventive nor corrective.

2.4 Algorithm Transparency

The complex nature of some machine learning models makes it hard for scientists to interpret the outcomes of such models. As a result, many models are considered to be black boxes. Another attempt to improve the fairness of machine learning models is to eliminate the black-box nature of these models' algorithms in what has been as described as algorithmic transparency. A solution within this area is a tool known as Local Interpretable Model-Agnostic Explanations (LIME) [14] which seeks to increase the transparency of classifications carried out by machine learning models, by giving an explanation to their outcomes. This helps humans take a step further into decisions on trusting the predictions these models make (regardless of how accurate they may be) as it provides more context on how the models come about the decisions they make.

Furthermore, researchers at Google have developed what is known as Google What-If [25], a tool that assists data scientists and individuals who create machine learning models to probe the behavior of such models. The What-If tool offers users an interactive web interface to explore the results of the models they build. It allows users to visualize data and examples from the models they build. It also displays the behavior and changes in the model's outcomes if any of the data given or examples they provide changes.

2.5 Measures for Bias Correction in Machine Learning

The concept of bias correction is not one that is novel in the field of machine learning. As machine learning gains ground in recent times and the effects of the bias it presents becomes more and more realistic and dire, researchers put in much more effort into exploring methods to curb the bias. However, as we have seen that bias stems from numerous areas, it is an area that presents great complexity. One of such methods focuses on the spreading of misclassification rates amongst the bias and weights that models learn as part of their features [7]. Work has also been conducted on the penalization of models by placing constraints on their loss functions based on the misclassifications they make [26]. Here, the constraints placed on the loss function are the aforementioned mathematical formalizations of fairness. These mathematical expressions are non-convex in nature and as such make it difficult to impose them on the convex nature of the model's optimization problems. To solve this, mathematical heuristics such as the Discipline Convex-Concave Program (DCOP) [26] are employed to enable them to be represented as convex functions. From here, the loss functions of predictive models are optimized, subject to these derived convex functions which act as mathematical constraints. In the attempt to correct bias in machine learning models, a tradeoff between model fairness and accuracy is presented [1, 13, 26].

2.6 Legal Work on Fairness in Machine Learning

In the legal sphere of algorithm governance, various regulations have been put in place to ensure the risk of bias in machine learning algorithms [11] is checked. In the United States of America, the Algorithm Accountability Act has been introduced to address the biases in decision making models that may affect sensitive groups of society negatively by denying them certain services or providing outcomes that deepen the data bias against them [2]. Though this bill is yet to be passed, it is also yet to be considered in other parts of the world to improve the effect of biased outcomes of machine learning models.

Chapter 3: Methodology

As has been discussed in this paper and in numerous works surrounding the ideas of fairness and bias correction in machine learning models, bias leading to unfair decisions may arise from the nature of the data employed in training said models. As a result, fairness in a model's outcomes is characterized in this paper as a closeness between selected misclassification rates (either false positive or false negative rates) for training examples belonging to various classes of a given sensitive feature.

$$p(\hat{y} \neq g \mid z = 0, g = 0) \approx p(\hat{y} \neq g \mid z = 1, g = 0)$$

where $\hat{y}$ = Model's predicted value [0,1],

g = Ground truth [0,1],

z = Sensitive feature [0,1]

The expression above demonstrates an attempt to neutralize disparate mistreatment (as explained in Section 2.2 of the paper) in models, by removing large disparities between the false positive rates between members of the sensitive group. Implementing these restrictions on the learning process of models implies that they may be forced to make a tradeoff between accuracy and fairness. For example, an attempt to address the disparity between the false positive rates of sensitive groups may result in an increase in the false negative rates of the groups. However, in some use cases, accuracy may not necessarily be an optimal measure of good performance in models. This is evident in some scenarios which require that models maintain either good precision or good recall [5]. As such this becomes a fair tradeoff for scientists to take into consideration [10].

3.1 Requirement Analysis on Data

Given that the goal of this paper is to implement and assess bias correction in machine learning models, the data to be employed must exhibit features that could give rise to learned bias. These may include but not be limited to race/ethnicity, gender, age and income levels of human subjects under consideration. For the purposes of this study, it is assumed that data under consideration is devoid of sample size and measurement bias (that is, the data

is representative of its larger population and tools of measurement are accurate). It is also imperative that the data contains the ground truth of predictions made within the dataset. Note that ground truth as referred to by this paper, differs from the dependent variable of a given set of data. Whereas the dependent variable (y) of the data may represent a classifier's output on a set of independent variables, the ground truth represents an assessment of correctness on the output of classification.

Assuming that the table below represents data employed in predicting whether or not an individual is carrying a gun (y). As demonstrated, Example 1 is predicted to be a carrier ($y = 1$). Say that after the prediction was made, police officers frisk the individual represented by Example 1 and find that indeed they carried a gun, ground truth (g) is also reported to be 1. This is referred to as a true positive. In the case of Example 2, $y = 0$ (meaning the classifier predicted the individual as carrying no gun) and $g = 1$ (after being frisked, the individual was found to be carrying a gun). This is referred to as a false negative. In Example 3, $y = 1$ (indicating that the individual was predicted to have a gun) but $g = 0$ (upon being searched, was found to carry no gun). Example 3 reflects the idea of a false positive. Finally, in Example 4, $y = 0$ (individual was predicting as not carrying a gun) and $g = 0$ (meaning that after being frisked was found to have no gun). This is referred to as a true negative.

Table 3.1: Demonstration of Difference between Dependent Variable and Ground truth

Example	Dependent Variable (y)	Ground Truth (g)
1	1	1
2	0	1
3	1	0
4	0	0

3.2 Dataset

The dataset under consideration to test this paper's approach is the ProPublica for FairML Dataset. This is a subset of the data sourced by ProPublica in its analysis of the COMPAS Recidivism Prediction Tool [17]. It contains 6172 examples of data about the demographics and conviction information of individuals convicted in Broward County, Florida in the United States. It also contains 10 rows of independent features including sensitive variables such as race and gender.

Below is a description of the dataset:

1. Two_yr_Recidivism: This variable represents the ground truth and informs on whether or not a defendant recidivated within two years. 1 for True, 0 for False.

2. Number_of_Priors: Number of past arrests for the given defendant.

3. Score_factor: Categorical variable encoding either a low or high COMPAS score. 1 implies high risk of recidivism and 0 implies low risk of recidivism.

4. Age_Above_FourtyFive: Boolean representing whether or not the defendant is older than forty-five years. 1 for True, 0 for False.

5. Age_Below_TwentyFive: Boolean representing whether or not the defendant is younger than twenty-five years. 1 for True, 0 for False.

6. African_American: Boolean representing whether or not the defendant is African American. 1 for True, 0 for False.

7. Asian: Boolean representing whether or not the defendant is Hispanic. 1 for True, 0 for False.

8. Hispanic: Boolean representing whether or not the defendant is Asian. 1 for True, 0 for False.

9. Native_American: Boolean representing whether or not the defendant is Native American. 1 for True, 0 for False.

10. Other: Boolean representing whether or not the defendant is either African American, Hispanic or Native American. 1 for True, 0 for False.

11. Female: Boolean representing whether the defendant is female. 1 for True, 0 for False.

12. Misdemeanor: Boolean representing whether or not the crime committed by the defendant was a misdemeanor or not. 1 for True, 0 for False.

3.3 Logistic Regression for Binary Classification

In choosing which machine learning model best suits the task of this paper (fairly predicting the risk scores of individuals in the COMPAS Recidivism Dataset), Logistic Regression was settled upon. Logistic Regression is suitable for classification problems as it is able to calculate the probability that a given example belongs to a given class based on its features. It is a simple classification model to employ in predicting binary categorical dependent variables. In this paper, Logistic Regression would be able to predict the probability of a given defendant being either high or low risk recidivism (risk score factor) based on their demographic and criminal information. Logistic Regression classifiers are able to calculate these probabilities by mapping the output of predictions with an activation function referred to as the sigmoid function.

$$S(z) = \frac{1}{1+e^{-z}}$$

where:

$S(z) =$ Probability value between 0 and 1

$z =$ Model's prediction

In the implementation of the work of this paper, a decision boundary is set such that a given probability estimate (p) maps to a risk score factor of either 0 (low risk recidivism) or 1 (high risk recidivism).

$$p \geq 0.5; y = 1 \text{ (high risk recidivism)}$$
$$p < 0.5; y = 0 \text{ (low risk recidivism)}$$

This was chosen to maintain an even split in the probability estimates of the model's decisions. Though this may not necessarily yield results that optimize the model's accuracy and/or fairness, this decision threshold is maintained as a control to ensure that desired optimizations result from the implementation of the work of this paper.

To ensure that Logistic Regression models learn to make accurate predictions, a loss function known as the Cross Entropy function is used to measure the error (cost) between

the model's predictions and the expected predictions at each step of is training process.

$$J(\Theta) = \frac{1}{m}\sum_{i=1}^{m} \text{cost}\left(h_\theta\left(x^i\right), y^i\right)$$

$$\text{cost}\left(h_\theta(x), y\right) = -\log\left(h_\theta(x)\right) \quad \text{if } y = 1$$

$$\text{cost}\left(h_\theta(x), y\right) = -\log\left(1 - h_\theta(x)\right) \quad \text{if } y = 0$$

where:

$m =$ length of training set

$h_\theta(x) =$ Model's prediction

$y =$ Expected prediction

This piecewise loss function can be compressed into one expressed as:

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m} \left[y^i \log\left(h_\theta\left(x^i\right) + \varepsilon\right) + \left(1 - y^i\right)\log\left(1 - h_\theta\left(x^i\right) + \varepsilon\right)\right]$$

Notice that by multiplying each of the components of the loss function by y^i and $1 - y^i$ respectively, the equation is able to solve for y = 0 and y = 1 cases separately. Also a very small constant ε is introduced to avoid running into log errors when computing the loss. The model's cost is minimized during gradient descent. In gradient descent, an optimal set of parameters that find the global minimum value of the loss function is learnt.
The gradient of the loss function is calculated as:

$$\theta_{t+1} = \theta_t - \alpha\sum_{i=1}^{m}\left(h_\theta\left(x^i\right) - y^i\right)x^T$$

where :

$\theta_t =$ Parameters at a given iteration,t

$\alpha =$ Learning rate

3.4 Approach

The work to be done in this paper may be thought of in two segments:

1. The Base Experiment: This involves demonstrating that inherent bias or unfair behavior stems from the data and sensitive features used in the training of the classifiers. This is done as a confirmatory procedure to ensure that it is worthwhile pursuing the

core work of this paper, as it would be needless to ensure fair decision making in a model if it does not make unfair decisions, or if its unfair decisions do not arise from the presence of sensitive features in its data.

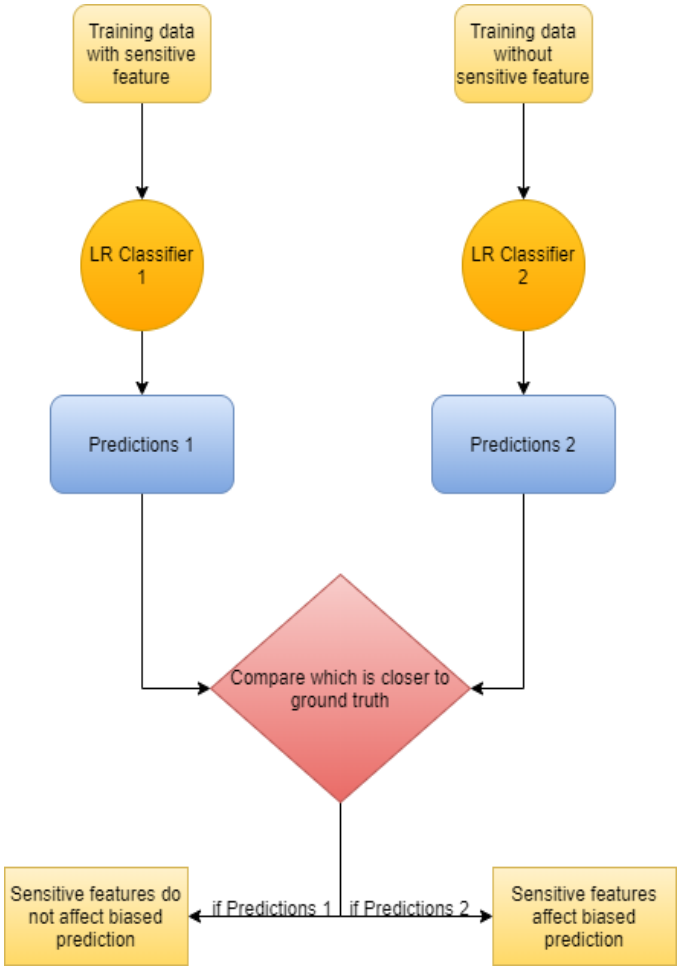


Figure 3.1: Flowchart of Base Experiment

2. The Core Experiment: This aspect of the paper's work has to do with the mechanisms put in place to abate the biased decision-making process of machine learning models. This is achieved by taking into account the gradient of the error between predictions made by the classifier and the ground truth when $z = 0$ and $z = 1$. The introduction of the gradient of this error forces the model to learn weights that bridge the disparity between the misclassification rates of the sensitive groups under consideration.

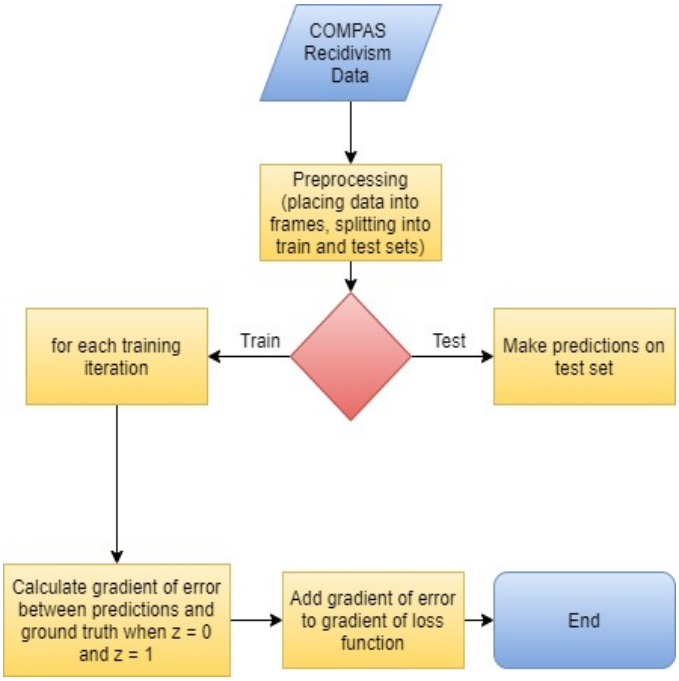


Figure 3.2: Flowchart of Core Experiment

3.5 Technologies Employed

3.5.1 Google Colaboratory

Google Colaboratory is a cloud based Python development environment developed by Google. It allows users to create machine learning and deep learning programs and experiments by utilizing the hardware capabilities of Google's cloud severs. On Google Colaboratory, users can run code on Graphical Processing Units (GPUs) or Tensor Processing Units (TPUs) to accelerate the time intensive process of deep learning computations for free.

Google Colaboratory comes ready with several useful packages and libraries to assist users to create machine learning programs and visualize their outputs and results. Its environment is similar to that of Jupyter Notebook, such that users write programs in modular bits referred to as cells. It also gives provision for users to provide some form of documentation to their software in the form of Markdown, HTML and LaTeXsyntax.

It was chosen as the preferred means of creating the experiments outlined in this paper as it provided all significant packages at hand. Since it is cloud based, it provides a form of security and ease of accessibility to the software.

3.5.2 Sci-kit Learn

Sci-Kit Learn is a machine learning library that includes tools for creating supervised and unsupervised machine learning models. It provides users with an abstraction of these machine learning models such that users need not know the ins and outs of them to apply them to their work. It also comes with various tools for preprocessing of data and evaluation of the model's performance. Sci-Kit learn was chosen as the preferred library to create the Logistic Regression classifier and evaluate its accuracy in the Base Experiment for its simplicity. It was also used in splitting the training data into test and train sets in a manner that ensures the same examples are used throughout the conducting of both of the experiments.

Sci-Kit Learn is built of the NumPy scientific computation library which is elaborated on below.

3.5.3 NumPy

NumPy serves as the basic package for scientific computations in Python. It allows users to represent information in multidimensional containers and perform mathematical operations such as linear algebraic calculations and Fourier transformations. It also has good implementations for the randomization of data. In the core experiment, data is contained and manipulated with NumPy.

3.5.4 Altair Visualization

The Altair Visualization Library is a powerful data visualization tool based on the Vega and Vega-lite visualization frameworks. It is useful in creating clear visualizations of the dataset being used and the output of the classifier. One major advantage it presents is the simplicity of its commands. It is used in visualizing the data and predictions made in both the Core and Base experiments.

3.5.5 Matplotlib

Matplotlib is a more commonly used visualization tool employed in Python applications. Though more verbose than the Altair visualization library, Matplotlib is useful in creating simple but powerful displays of the outcomes of the experiment. It is used in both

the Base and Core experiments to visualize the models predictions and performance. Other libraries employed such as the Sci-Kit Plot visualization package are built from Matplotlib.

Chapter 4: Methodology 2 - Implementation

4.1 The Base Experiment

As has been previously established, bias may not only arise from data. It may stem from the implementation of algorithms that perform classifications on the given data. For the work of the experiment to be conducted in this paper, it must firstly be established that a model's exhibition of bias stems from its data and specifically the presence of sensitive features in the data. Since previous studies suggest that COMPAS has the tendency to make unfair predictions towards people who possess certain racial features [12,17], this aspect of the paper's work is conducted to assess the nature of predictions made by a machine learning classifier with regards to individuals belonging to specific racial groups within the dataset under consideration.

In short, the sensitive feature under consideration here is race and more specifically, whether or not a person is African American or not (Other). This separation into two fundamental racial groups also aids in creating an almost even split in the proportions of different races in the dataset, as African Americans make up a larger number of examples in the data.

From here, a Logistic Regression (LR) classifier is built to emulate the COMPAS prediction algorithm such that, the created classifier should be able to predict the risk score factors of individuals based on given independent variables including racial information. The classifier is trained on 80% of the examples in the data described earlier, to predict the score factor of individuals within the remaining 20% of the data set which is set aside for testing. It's accuracy of prediction is then measured to show how well it is able to emulate the predictions of the COMPAS algorithm.

Table 4.1: Distribution of Training and Test data

Dataset	Number of Examples	African American	Other
Training	4937	2536	2401
Testing	1235	639	596
Total	6172	3175	2997

Subsequently, a second classifier that trains without a given sensitive feature of the dataset (which is racial information in this case) is implemented and the risk score factors

of individuals within the test dataset are predicted from it. Its accuracy of prediction will be assessed relative to the aforementioned classifier to determine how well it is able to accurately predict recidivism without the data's racial information. Though this aspect of the experiment takes a naïve approach by simply eliminating the sensitive feature under consideration, it demonstrates the impact of race on the classifier's unfair decision making.

It must however be noted that the removal of a given sensitive feature from the data that a given machine learning classifier trains on may not necessarily give rise to fairer classifications. This is because many times, features taken into consideration are not mutually exclusive and as such may be telling of which sensitive features training examples may possess or not. For example, names of individuals may be indicative of their gender or race and addresses may serve as proxies to their income levels.

4.2 The Core Experiment

After it has been established that bias arises from the data employed in training, it becomes feasible to implement measures to deal with the biased outcomes of machine learning models. In the core experiment, disparities between the misclassifications across the sensitive feature that are based on the data's ground truth are employed to inform the machine learning models on unfairness in its learning process. Misclassifications may be quantified as either False Positive (FP) rate or a False Negative (FN) rate based on the use case at hand. However, for the purpose of this experiment and with regards to the data being employed (the COMPAS Recidivism data), the FP rate is taken as the misclassification rate to be considered. This is because in the case of predicting risk scores with the COMPAS Recidivism dataset, it is considered unfair to rate a defendant as being high risk recidivism when in fact the defendant in question does not recommit the offence. As such what one hopes to achieve in terms of fairness in such a model, would be to narrow the big gap between the FP rates for both African Americans and individuals of other racial origins (Other). This is essentially telling the model to ensure that the probability that a person of Other racial origins being predicted as high risk when in fact, they do not recidivate should be as equal as possible to that of an African American being predicted as high risk when in

fact they do not recidivate. The FP rate is calculated as:

$$FP \text{ rate } = \frac{FP}{FP+TN}$$

where:

$FP =$ Number of false positives

$TN =$ Number of true negatives

As with the Base Experiment, race is the sensitive feature (z) under consideration here as well. Race is constructed as it was in the Base Experiment with z being either African American (to indicate that a defendant is African American) or Other (to indicate that a defendant is of another race). For simplicity, the two racial groups are expressed as $z = 0 \mid 1$ where 0 represents Other and 1 represents African American.

The goal of this aspect of the experiment is to enforce parity in the probability of misclassification between the given racial groups. To achieve this, an approach based on the modification of the loss function of a Logistic Regression classifier to allow it to consider, the gradient of the error between predictions made and the ground truth for $z = 1$ and $z = 0$.

4.2.1 Gradient of Error Between Predictions and Ground Truth

In the implementation of this approach, the gradients of the error of prediction must be separated according to z and calculated accordingly. This is done by reconstructing the feature array and ground truth such that we now have two different feature arrays for when $z = 0$ and $z = 1$ respectively. This is demonstrated in the tables below:
Assuming a feature table of independent variables:

Table 4.2: Demonstration of Feature Table with Features (X_n) and Sensitive Feature (z)

Example	X_1	...	z
1	1	0	0
2	10	1	1
3	7	1	0
4	0	0	0

When $z = 0$, Table 4.2 is reconstructed such that all rows where $z = 1$ are filled with zeros:

Table 4.3: Demonstration of Reconstructed Feature Table for $z = 0$

Example	X_1	...	z
1	1	0	0
2	0	0	0
3	7	1	0
4	0	0	0

And when $z = 1$, Table 4.2 is reconstructed such that all rows where $z = 0$ are filled with zeros:

Table 4.4: Demonstration of Reconstructed Feature Table for $z = 0$

Example	X_1	...	z
1	0	0	0
2	10	1	1
3	0	0	0
4	0	0	0

The same approach outlined above is taken to reconstruct the ground truth (g). This is done to ensure that for each X, g where $z = 0$ and $z = 1$ respectively, predictions are made without taking into account the value of the opposite z's features. From here, the gradient of the error between the prediction and ground truth for each sensitive group $z = 0, 1$ is computed. This gradient is introduced into the loss function as follows:

$$\theta_{t+1} = \theta_t - \frac{\alpha}{m}\left(\sum_{i=1}^{m}\left(\beta_0\left(\left(h_\theta\left(x^i\right) - y^i\right)x^T\right) + \beta_1\left(\left(h_\theta\left(x^i\right) - g_0^i\right)\left(x_0^i\right)^T\right) - \left(\left(h_\theta\left(x^i\right) - g_1^i\right)\left(x_1^i\right)^T\right)\right)\right)$$

where:

$g_o =$ Ground truth of training exampled for $z = 0$

$x_0 =$ Feature array of training examples for $z = 0$

$g_1 =$ Ground truth of training exampled for $z = 1$

$x_1 =$ Feature array of training examples for $z = 1$

Note that the variable β_0 is introduced to place weight on the classifier's correct predictions according to the dependent variable y. And β_1 is a weight placed on the classifier's "fair" predictions according to the ground truth. Both values of β must sum up to 1

4.2.1.1 Algorithm of Implementation of Core Experiment

Algorithm 1 below demonstrates the implementation of this methodology in software.

Algorithm 1 Batch Cross Entropy Gradient Descent including the Gradient of Error Between Predictions and Ground Truth

```
Input:Parameters (θ), Learning rate (α), β0, β1, iterations
Output:  Optimal Parameters
Data:  Testing set (X), Dependent Variable (y), Ground truth (g)

procedure GradientDescent
    x0 ← X where X_{r,z=1} = 0                ▷ r represents row of feature table
    x1 ← X where X_{r,z=0} = 0
    g0 ← g where g_{r,z=1} = 0
    g1 ← g where g_{r,z=0} = 0
    for i ← 0 to iterations − 1 do
        hθ(x) ← σ(X · θ)
        θ ← θ − (α/m) * (β0 * (hθ(x^i) − y^i) · x^T +
            β1 * ((hθ(x^i) − g0^i) · (x0^i)^T − ((hθ(x^i) − g1^i) · (x1^i)^T)))
                                                  ▷ m represents the length of X
    return θ                                      ▷ The optimal parameters
```

4.2.2 Using Mean Squared Error of Misclassification Rates

Before implementing the approach stated above, one based on penalizing the loss function of a Logistic Regression classifier with a constant metric (penalty) was implemented. The calculation of this penalty is computed as the mean squared error between the FP rates for $z = 1$ and $z = 0$. To achieve this, at every iteration of learning, the false positive (FP) rates of that iteration are calculated for groups $z = 0$ and $z = 1$ respectively. The Mean Squared Error (MSE) between the aforementioned FP rates is calculated to serve as a penalty to the classifier's loss function. The Mean Squared Error is calculated as:

$$MSE_{FPrate} = \frac{(FP\ rate_0 - FP\ rate_1)^2}{2}$$

where:

$FP\ rate_0 =$ FP rate when z=0

$FP\ rate_1 =$ FP rate when z=1

In this experiment, the classifier's gradient descent was carried out as Batch Gradient Descent where the batch size was the entire training dataset. This means that for each

iteration of the training step, predictions are made on the entire training set and the gradient of the error between predictions and the expected prediction is calculated from it. This implies that for each training step, the FP rates and MSE are calculated based on predictions and expected output of the entire training dataset.

After calculating the MSE of the FP rates, it is added to the gradient descent function as follows:

$$\theta_{t+1} = \theta_t - \frac{\alpha}{m}\left(\sum_{i=1}^{m}\left(\beta_0\left(h_\theta\left(x^i\right) - y^i\right)x^T + \beta_1 MSE_{FPrate}\right)\right)$$

where:

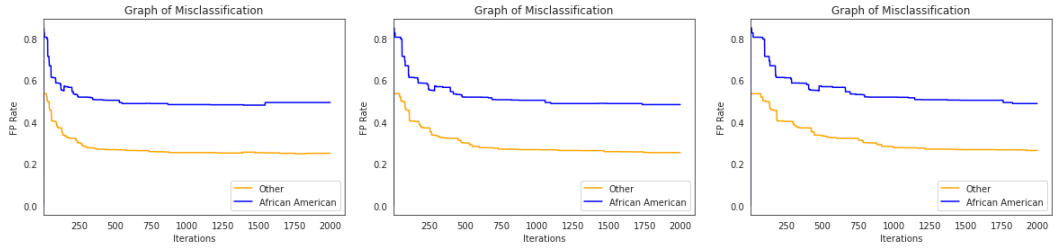
$\theta_t =$ Parameters at a given iteration

$\alpha =$ Learning rate

$MSE_{FP_{rate}} =$ Mean Squared Error of False Positive rates

$\beta_0, \beta_1 =$ Number between [0,1]

This approach did not yield desired results as for each β value pair, the classifier was unable to address the issue of disparity between the FP rates. This is demonstrated in Figure 4.1 below that shows the classifiers FP rates per iteration for β value pairs $(1,0)$, $(0.5,0.5)$ and $(0.3,0.7)$ respectively.



(a) $\beta_0 = 1, \beta_1 = 0$ (b) $\beta_0 = 0.5, \beta_1 = 0.5$ (c) $\beta_0 = 0.3, \beta_1 = 0.7$

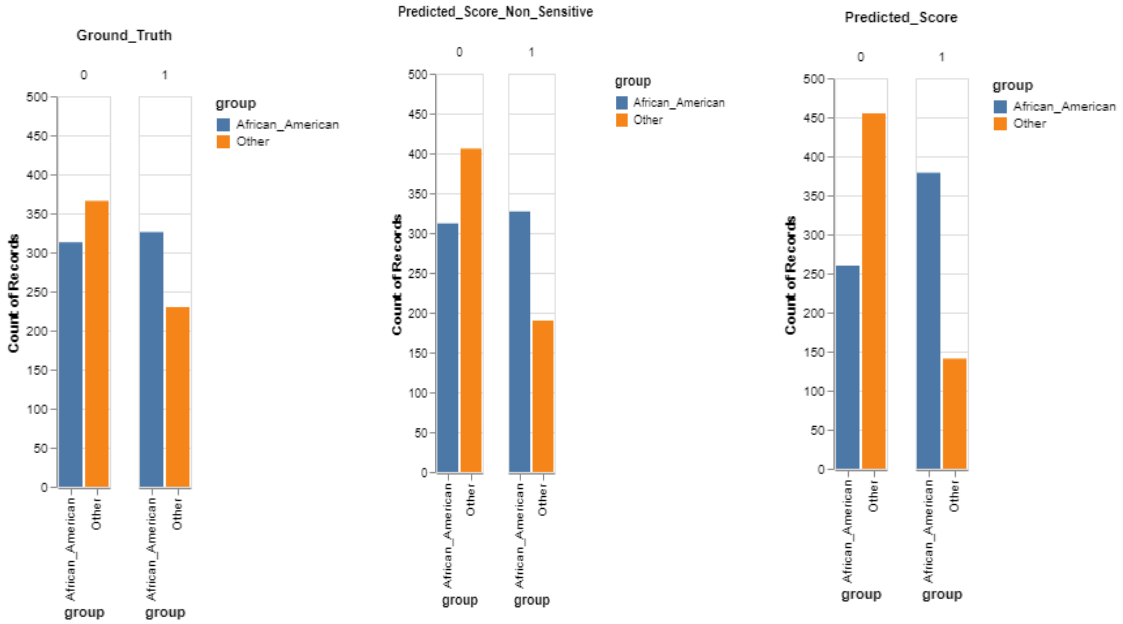
Figure 4.1: Results Achieved Using Mean Squared Error of Misclassification Rates

Chapter 5: Results

The results of the experiments conducted to achieve the goal of this paper are outlined in the subsections below.

5.1 Results of Base Experiment

The Base Experiment buttressed the findings of previous literature [12, 17] on the presence of racial unfairness in the COMPAS prediction algorithm.



(a) Ground Truth of Recidivism across z

(b) Predictions of Recidivism across z from "Fair" Classsifier

(c) Predictions of Recidivism across z from "Unfair" Classsifier

Figure 5.1: Results of Base Experiment

The graphs above are discussed in more detail in the table below:

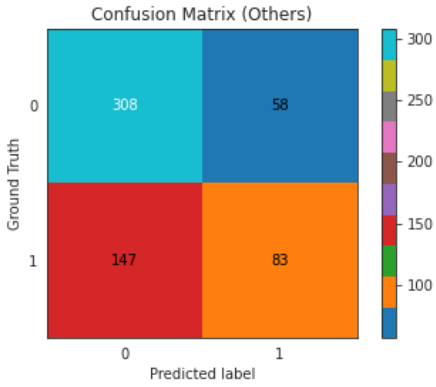
Table 5.1: Distribution of Classifier Predictions Across Sensitive Feature (z)

	Ground Truth		"Fair" Classifier (approx. count of y across z based on X with no racial information)		"Unfair" Classifier (approx. count of y across z based on X with racial information)	
Race ($z = 0, 1$)	Other	African American	Other	African American	Other	African American
High Risk (1)	230 (18.4%)	330 (26.4%)	190 (15.2%)	330 (26.4%)	140 (11.2%)	370 (29.6%)
Low Risk (0)	370 (29.6%)	320(25.6%)	410 (32.8%)	320 (25.6%)	450 (36%)	260 (20.8%)

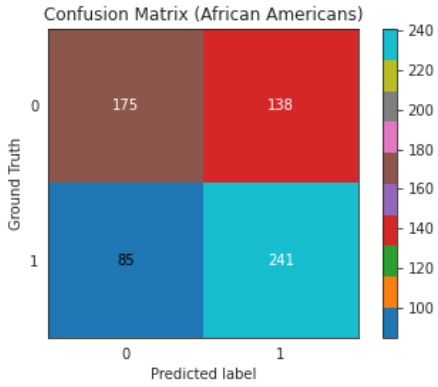
From Figure 5.1a above, we can tell that the data in the ground truth shows that about 320 African Americans did not recidivate within two years while around 370 individuals of other racial origin did not recidivate as well. However approximately 330 African Americans and around 230 people of other racial origin recidivated within two years.

Figure 5.1c demonstrates the result of predictions made by the "unfair" Logistic Regression classifier that has racial information as part of the features it learns. From it we can see that around 260 African Americans were predicted as low risk recidivism. A significantly lower number when compared to the ground truth. Also, around 450 people of other racial origin were predicted as low risk recidivism which is significantly higher than the ground truth depicts. It is also evident that the classifier predicts more African Americans as high-risk recidivism and significantly less people of other origin as high risk recidivism when compared to the ground truth (approx. 370 vs. 330 and 140 vs 230 respectively).

The confusion matrices of this classifier separated for $z = 0$ and $z = 1$ are depicted below.



(a) Confusion matrix demonstrating misclassifcations according to ground truth for $z = 0$. The number of false positives are found in the upper right quadrant.

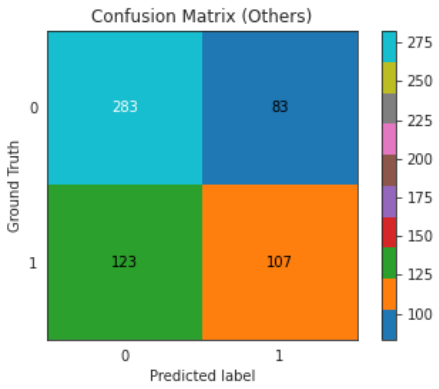


(b) Confusion matrix demonstrating misclassifcations according to ground truth for $z = 1$. The number of false positives are found in the upper right quadrant.

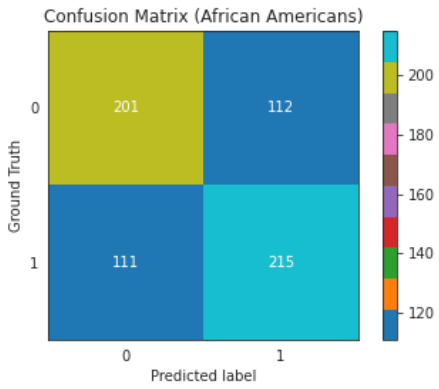
Figure 5.2: Confusion Matrices from "Unfair" Classifier

Here, one can tell that the classifier has a low number of false positives and a high number of false negatives when $z = 0$. On the other hand, when $z = 1$ it has a high number of false positives and a low number of false negatives. This tells us that it classifies less individuals of other races as high risk when they do not recidivate than it does African American individuals as high risk when they do not recidivate. The classifier also wrongly classifies African Americans as low risk (when according to ground truth they are not), less often that it does for other races.

Figure 5.1b shows results of predictions made by the "fair" Logistic Regression classifier that does not have racial information as part of its training features. Just by taking a glance at it, it becomes obvious that removing racial features from the training dataset results in predictions that are closer to the ground truth. Here, approximately 320 African Americans and 410 people of other racial origin are predicted as low risk recidivism. Around 330 African Americans and about 190 people of other origin are predicted as high risk recidivism. The confusion matrices of this classifier for $z = 0$ and $z = 1$ are depicted below.



(a) Confusion matrix demonstrating misclassifcations according to ground truth for $z = 0$. The number of false positives are found in the upper right quadrant.



(b) Confusion matrix demonstrating misclassifcations according to ground truth for $z = 1$. The number of false positives are found in the upper right quadrant.

Figure 5.3: Confusion Matrices from "Fair" Classifier

From the matrices, it depicts that the number of false positives increases when $z = 0$ and decreases when $z = 1$. The number of false negatives, however, decrease when $z = 0$ and increase when $z = 1$. This demonstrates the tradeoff between model accuracy and fairness because though this classifier makes fairer decisions they may not necessarily be the accurate decisions regarding the data.

5.2 Results of Core Experiment

The Core Experiment was carried out using three different sets of weights on the model's correctness and fairness (β_0 and β_1 respectively). In detail, three sets of sub-experiments were conducted using:

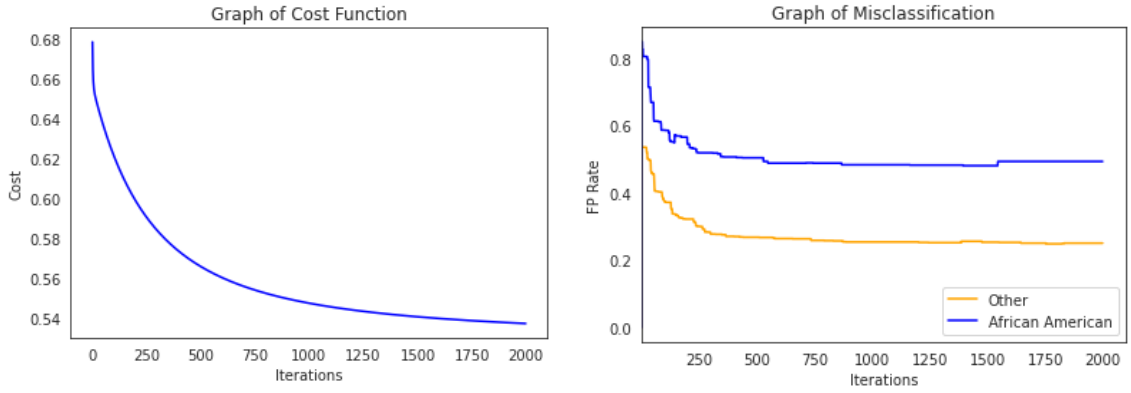
1. $\beta_0 = 1$ and $\beta_1 = 0$
2. $\beta_0 = 0.5$ and $\beta_1 = 0.5$
3. $\beta_0 = 0.3$ and $\beta_1 = 0.7$

The results obtained by these in the experiment are elaborated on below:

5.2.1 $\beta_0 = 1$ and $\beta_1 = 0$

Using these values of β, the model behaves as a regular Logistic Regression classifier. It also reflects the behavior of the "unfair" classifier in experiment one. Here, training is

completed in 120.11 seconds with a validation accuracy of 70.4% The cost graph and graph of FP rate per iteration are shown below:

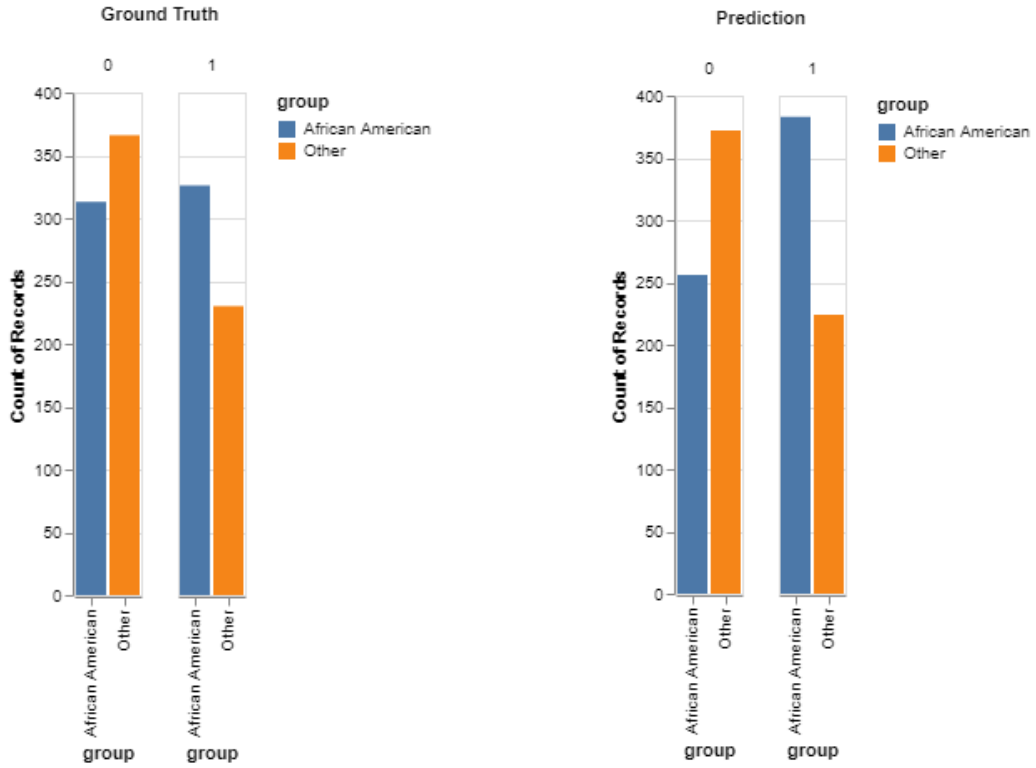


(a) Cost per Iteration of Classifier

(b) FP Rates per Iteration for Both $z = 0$ and $z = 1$

Figure 5.4: Results of Core Experiment with $\beta_0 = 1$ and $\beta_1 = 0$

The graph of predictions side by side with the ground truth are also identical to that of the "unfair" classifier in the Base Experiment.



(a) Ground Truth of Recidivism across z

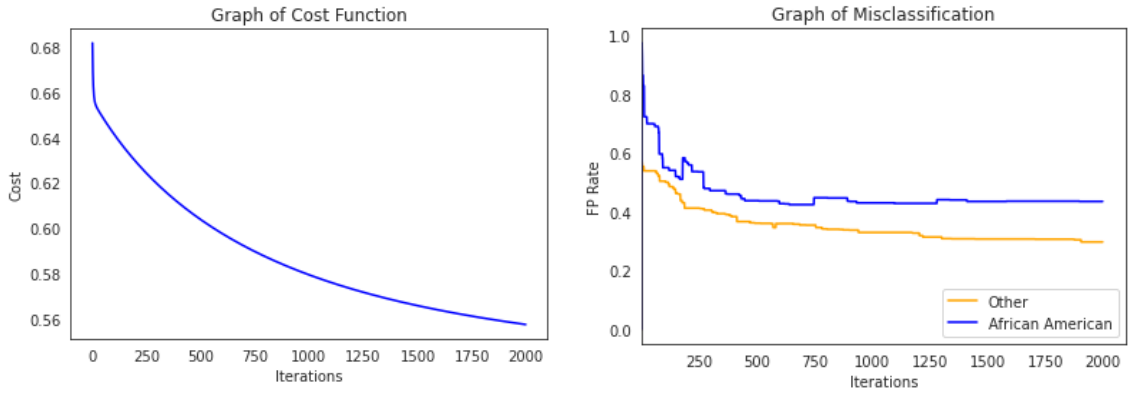
(b) Predictions of Recidvism across z

Figure 5.5: Predictions of Core Experiment with $\beta_0 = 1$ and $\beta_1 = 0$

From Fig 5.5, one may observe that using these weight values, the classifier is able to make predictions that are closer to the ground truth than when the classifier does not take fairness into consideration.

5.2.2 $\beta_0 = 0.5$ and $\beta_1 = 0.5$

Introducing equal weights to the values of β, the classifier was able to complete training in 120.59 seconds with a validation accuracy of 70.4%



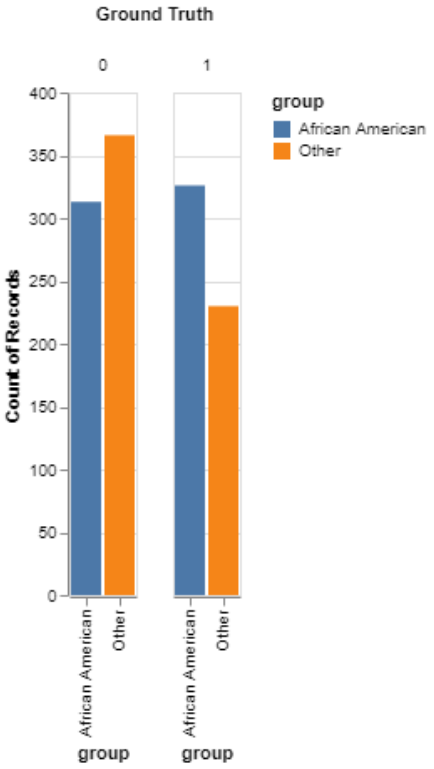
(a) Cost per Iteration of Classifier

(b) FP Rates per Iteration for Both $z = 0$ and $z = 1$

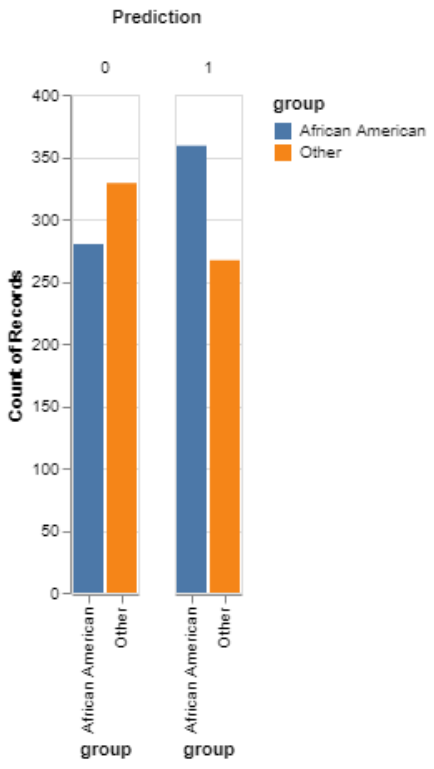
Figure 5.6: Results of Core Experiment with $\beta_0 = 0.5$ and $\beta_1 = 0.5$

The cost function is able to reach a minimum after the number of iterations. However, it does not level out smoothly in the end and this can be resolved by introducing more training iterations. Interestingly, the FP rates for both values of z decrease as the training iterations increase. However, the disparity between them is addressed such that the FP rate when $z = 1$ gets closer to the FP rate when $z = 0$. This does well to address the issue of unfairness as set out by this paper because the classifier is able to ensure that members of each racial groups are misclassified similarly.

The ground truth and predictions made with these weight values are placed side by side below



(a) Ground Truth of Recidivism across z



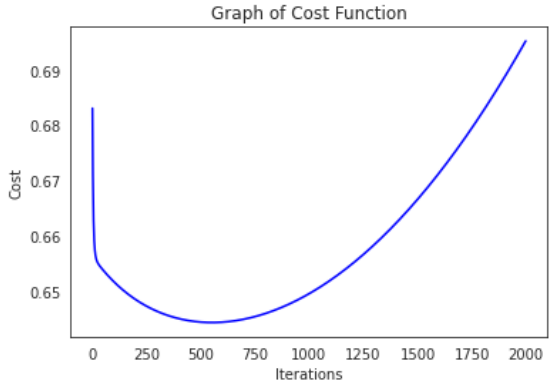
(b) Predictions of Recidvism across z

Figure 5.7: Predictions of Core Experiment with $\beta_0 = 0.5$ and $\beta_1 = 0.5$

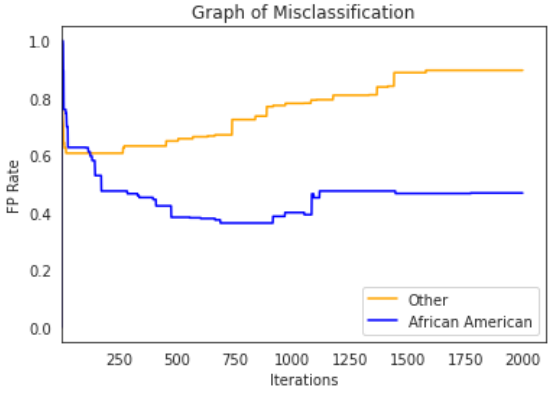
From Figure 5.7, one may observe that using these weight values, the classifier is able to make predictions that are closer to the ground truth than when the classifier does not take fairness into consideration.

5.2.3 $\beta_0 = 0.3$ and $\beta_1 = 0.7$

Here, more emphasis is placed on fairness than correctness of prediction. The classifier was able to complete training in 119.47 seconds with a validation accuracy of 53.5%. This is poor performance as it does not perform any better than a random guess when predicting, assuming that each example had equal chances of being predicted as high or low risk recidivism.



(a) Cost per Iteration of Classifier

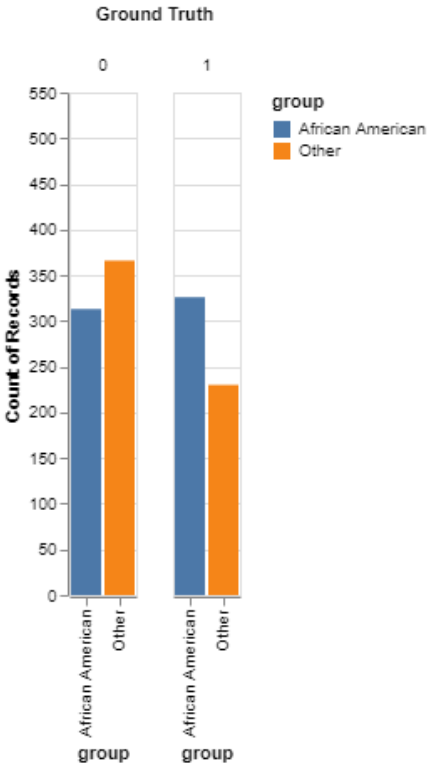


(b) FP Rates per Iteration for Both $z = 0$ and $z = 1$

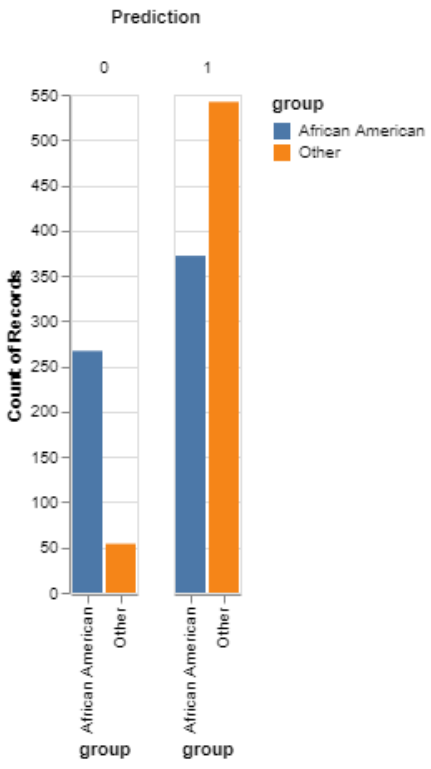
Figure 5.8: Results of Core Experiment with $\beta_0 = 0.3$ and $\beta_1 = 0.7$

From Figure 5.8a, one may observe that the cost eventually begins to rise as the iterations increases. This phenomenon indicates that the model is unable to accurately learn a good set of parameters for prediction. From Figure 5.8b, the FP rate when $z = 1$ takes a significant drop only to increase again after a thousand iterations. The FP rate when $z = 0$ however, increases gradually and levels out at the end. This is not desirable behavior as it implies that the classifier makes predictions on the group $z = 0$ unfairly.

The graphs of predictions and ground truth demonstrate the further below:



(a) Ground Truth of Recidivism across z



(b) Predictions of Recidvism across z

Figure 5.9: Predictions of Core Experiment with $\beta_0 = 0.3$ and $\beta_1 = 0.7$

This shows that placing a large emphasis on fairness and a small one on correctness results in neither correct nor fair predictions.

Chapter 6: Conclusions and Further Work

6.1 Summary of Conclusions

The paper explored a method of enforcing fairer outcomes on machine learning models. The method includes the gradient of the error of prediction for members of a given sensitive group in the loss function. Both methods used a subset of the COMPAS Recidivism Dataset used in predicting individuals' risk scores. For the purposes of this research, the sensitive feature under consideration was race and the classifier of choice was a Logistic Regression.

This attempt performs well in achieving the goal of the paper in terms of fairness when an even emphasis was placed on the model's fairness and correctness. In this case it is able to address the disparity between the false positive rates of the sensitive groups by reducing the difference between them by 52%, while lowering them ultimately. It also mirrors the ground truth of prediction closely and maintains an accuracy of 70.4%. This is very similar to the accuracy of a regular Logistic regression (70.2%) as demonstrated in the paper and does not present the tradeoff between accuracy and fairness discussed in literature [1, 13, 26]. However, placing more emphasis on fairness than correctness results in an inability of the model to learn either correctly or fairly. On average, the experiment takes 120.06 seconds to return results.

A previous experiment conducted involving the addition of the mean squared error between the false positive rates of prediction did not produce significant results regardless of the weight placed on fairer prediction, such that the behavior of penalized models did not differ from that of regular Logistic Regression. This is because the calculated mean squared error is introduced as a constant and added to the gradient in the gradient descent. For it to have an effect on the model's learning, it should be a function of the model's parameters

6.2 Limitations

The following are the limitations on the work of this paper. They are the factors that affect the performance of development and results produced by them:

1. The Core Experiment can only be carried out on sets of data for which ground truth is present. This limits their applications as many current machine learning problems do

not collect ground truth as part of the data they utilize in making decisions.

2. The current implementation of the Core Experiment is inefficient due to the reconstruction of the features to enable them to be separated according to racial group. This causes the learning process of the classifier to slow down significantly.

3. Finally, the subset of the data that was used to conduct this study was significantly smaller than the original COMPAS dataset. This was because several examples were discarded as some of their features were missing or not considered useful to the study. This limited the performance of the classifiers that were built as more data could have improved their accuracies.

6.3 Suggestions for Further Work

To improve the work done in this paper, the following may be considered:

1. A refinement of prior work done int the Core Experiment to express the Mean Squared Error as a function of parameters to allow for the gradient of it to be applied to the loss function. This improves the functionality of the experiment and has the potential to yield meaningful results.

2. The problem of this paper may be framed as a Reinforcement Learning question such that the state of the model's training informs it on what actions to take to maximize a reward expressed in terms of how well the model is doing with regards to fairness.

3. The speed of the experiments conducted may be improved by looking into more efficient ways to reconstruct training fields and separate them according to the sensitive feature under consideration.

References

[1] Julius A. Adebayo. *FairML : ToolBox for diagnosing bias in predictive modeling*. Thesis, Massachusetts Institute of Technology, 2016. Accepted: 2017-04-18T16:37:35Z Journal Abbreviation: ToolBox for diagnosing bias in predictive modeling.

[2] Vaidyanathan Balasubramanian. Algorithmic Accountability Act of 2019 – Challenges & Opportunities - Wipro, June 2019. Library Catalog: www.wipro.com.

[3] Jeremy Berg. Measuring and managing bias. *Science*, 357(6354):849–849, September 2017. Publisher: American Association for the Advancement of Science.

[4] Ekaba Bisong. Google Colaboratory. In Ekaba Bisong, editor, *Building Machine Learning and Deep Learning Models on Google Cloud Platform: A Comprehensive Guide for Beginners*, pages 59–64. Apress, Berkeley, CA, 2019.

[5] Jason Brownlee. Classification Accuracy is Not Enough: More Performance Measures You Can Use, March 2014. Library Catalog: machinelearningmastery.com Section: Machine Learning Process.

[6] Aylin Caliskan, Joanna J. Bryson, and Arvind Narayanan. Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334):183–186, April 2017. Publisher: American Association for the Advancement of Science Section: Reports.

[7] Irene Chen, Fredrik D. Johansson, and David Sontag. Why Is My Classifier Discriminatory? *arXiv:1805.12002 [cs, stat]*, December 2018. arXiv: 1805.12002.

[8] Jeffrey Dastin. Amazon scraps secret AI recruiting tool that showed bias against women. *Reuters*, October 2018.

[9] Chris DeBrusk. The Risk of Machine-Learning Bias (and How to Prevent It), March 2018. Library Catalog: sloanreview.mit.edu.

[10] Richard Dinga, Brenda W. J. H. Penninx, Dick J. Veltman, Lianne Schmaal, and Andre F. Marquand. Beyond accuracy: Measures for assessing machine learning models, pitfalls and guidelines. *bioRxiv*, page 743138, August 2019. Publisher: Cold Spring Harbor Laboratory Section: New Results.

[11] Danilo Doneda and Virgilio A.F. Almeida. What Is Algorithm Governance? *IEEE Internet Computing*, 20(4):60–63, July 2016. Conference Name: IEEE Internet Computing.

[12] Julia Dressel and Hany Farid. The accuracy, fairness, and limits of predicting recidivism. *Science Advances*, 4(1):eaao5580, January 2018. Publisher: American Association for the Advancement of Science Section: Research Article.

[13] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Rich Zemel. Fairness Through Awareness. *arXiv:1104.3913 [cs]*, November 2011. arXiv: 1104.3913.

[14] Carlos Guestrin, Marco Tulio Ribeiro, and Sameer Singh. Local Interpretable Model-Agnostic Explanations (LIME): An Introduction, August 2016. Library Catalog: www.oreilly.com.

[15] Hennie de Harder. Model-Agnostic Methods for Interpreting any Machine Learning Model, January 2020. Library Catalog: towardsdatascience.com.

[16] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.

[17] Jeff Larson, Surya Mattu, Lauren Kirchner, and Julia Angwin. How We Analyzed the COMPAS Recidivism Algorithm, May 2016. Library Catalog: www.propublica.org.

[18] Travis Oliphant. NumPy: A guide to NumPy. USA: Trelgol Publishing, 2006–. [Online; accessed May 28, 2020].

[19] Osonde A. Osoba and William Welser IV. *An Intelligence in Our Image: The Risks of Bias and Errors in Artificial Intelligence*. Rand Corporation, April 2017.

[20] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[21] Pedro Saleiro, Benedict Kuester, Loren Hinkson, Jesse London, Abby Stevens, Ari Anisfeld, Kit T. Rodolfa, and Rayid Ghani. Aequitas: A Bias and Fairness Audit Toolkit. *arXiv:1811.05577 [cs]*, April 2019. arXiv: 1811.05577.

[22] Jaspreet Sandu. Understanding and Reducing Bias in Machine Learning, April 2019. Library Catalog: towardsdatascience.com.

[23] Jacob VanderPlas, Brian Granger, Jeffrey Heer, Dominik Moritz, Kanit Wongsuphasawat, Arvind Satyanarayan, Eitan Lees, Ilia Timofeev, Ben Welsh, and Scott Sievert. Altair: Interactive Statistical Visualizations for Python. *Journal of Open Source Software*, 3(32):1057, December 2018.

[24] Olivia Wassén. Big Data facts - How much data is out there?, October 2018. Library Catalog: www.nodegraph.se Section: Guides.

[25] James Wexler. The What-If Tool: Code-Free Probing of Machine Learning Models, September 2018. Library Catalog: ai.googleblog.com.

[26] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rodriguez, and Krishna P. Gummadi. Fairness Beyond Disparate Treatment & Disparate Impact: Learning Classification without Disparate Mistreatment. *Proceedings of the 26th International Conference on World Wide Web*, pages 1171–1180, April 2017. arXiv: 1610.08452.

[27] Ziyuan Zhong. A Tutorial on Fairness in Machine Learning, July 2019. Library Catalog: towardsdatascience.com.