Improve Algorithmic Fairness in Recidivism Prediction

Akshit Nanda

A dissertation submitted for the partial fulfillment of BS-MS dual degree in Science

Indian Institute of Science Education and Research, Mohali

May, 2023

Certificate of Examination

This is to certify that the dissertation titled **Improve Algorithmic Fairness in Recidivism Prediction** submitted by **Akshit Nanda** (Reg. No. MS18216) for the partial fulfillment of BS-MS Dual Degree programme of the institute, has been examined by the thesis committee duly appointed by the institute. The committee finds the work done by the candidate satisfactory and recommends that the report be accepted.

Dr. Neeraja Sahasrabudhe
(Committee Member)

Dr. Shane D'Mello
(Committee Member)

Dr. Vaibhav Vaish
(Supervisor)

Dr. Sarabjot Singh Anand
(External Supervisor)

Dated: May, 2023

Declaration

The work presented in this dissertation has been carried out by me under the guidance of Dr. Sarabjot Singh Anand at the Indian Institute of Science Education and Research, Mohali.

This work has not been submitted in part or in full for a degree, a diploma, or a fellowship to any other university or institute. Whenever contributions of others are involved, every effort is made to indicate this clearly, with due acknowledgment of collaborative research and discussions. This thesis is a bonafide record of my original work done by me and all sources listed within have been detailed in the bibliography.

Akshit Nanda
(Candidate)
Dated: May, 2023

In my capacity as the supervisor of the candidate's project work, I certify that the above statements by the candidate are true to the best of my knowledge.

Dr. Sarabjot Singh Anand
(Supervisor)
Dated: May, 2023

Acknowledgements

I want to express my most profound appreciation and gratitude to all those who have supported me throughout my journey to completing my Master of Science (MS) thesis. I am sincerely grateful for your invaluable contributions and unwavering support. First and foremost, I express my heartfelt gratitude to my thesis advisors, Dr.Sarabjot Singh Anand, and Dr. Vaibhav Vaish, for their constant guidance, mentorship, and encouragement. I am truly grateful for their unwavering support and invaluable feedback.

I sincerely thank my family for their unwavering love, encouragement, and support throughout my academic journey. Their belief in my abilities and constant encouragement has been a driving force behind my success, and I am deeply grateful for their presence in my life.

I am also grateful to my friends for their support and camaraderie. Your friendship, motivation, and encouragement have made this journey memorable and enjoyable. Lastly, I thank all the researchers, scholars, and authors whose works have been referenced in my thesis. Their research and contributions to the field have been invaluable in shaping my understanding and analysis of the topic.

In conclusion, I am deeply grateful to all those who have supported me in various ways, directly or indirectly, in completing my MS thesis. Your guidance, encouragement, and support have been instrumental in my academic and research journey, and I sincerely appreciate your contributions.

Abstract

Algorithmic fairness has become an essential research topic due to the potential for machine learning algorithms to perpetuate existing biases and discrimination in society. Many algorithms have been shown to produce unfair outcomes for certain groups, particularly those who have historically faced discrimination. Therefore, it is important that the outcomes of Machine Learning (ML) models are fair and do not lead to discrimination. Improving algorithmic fairness involves developing methods to ensure that algorithms are fair and unbiased in their decision-making processes.

One approach to improving algorithmic fairness is using data preprocessing techniques such as data cleaning, sampling, and feature engineering. These techniques aim to remove or mitigate any biases that may be present in the data used to train the algorithm. Another approach involves modifying the algorithm to incorporate fairness constraints, such as requiring the algorithm to treat all groups equally.

In this thesis, we analyzed five different bias mitigation methods belonging to different fairness-enhancing techniques and decide which method best suits our dataset. The results show to great extent, the pre-processing technique works best for our dataset.

List of Figures

2.1 Sigmoid function 12
2.2 Hyperplane in 2D and 3D feature space 15
2.3 Possible hyperplanes and optimal hyperplane 15

4.1 Stability 25
4.2 Accuracy 26
4.3 F1 score 27
4.4 TPR difference 27
4.5 Equalized Odds 28
4.6 Disparate Impact 28

List of Tables

2.1 Confusion matrix 8

4.1 Characteristics of recidivism dataset 22
4.2 Base rate for sex attribute 22
4.3 Base rate for race attribute 22
4.4 Base rate for all possible sex-race groups 23
4.5 Results of the experiments on the recidivism dataset 29

List of Abbreviations

Measures

BR	Base rate
TPR	True Positive Rate
TNR	True Negative Rate
FPR	False Positive Rate
FNR	False Negative Rate

Fairness measures

EO	Equalized Odds
DI	Disparate Impact

Accuracy measures

BAS	Balanced Accuracy Score

Fairness-enhancing techniques

DIR	Disparate Impact Remover
CND	Classification with No Discrimination
IGD	Individual and Group Debiasing
TO	Threshold Optimizer
PR	Prejudice Remover

Contents

Acknowledgements **V**

Abstract **VII**

List of Figures **IX**

List of Tables **XI**

List of Abbreviations **XIII**

1 Introduction **1**
1.1 Motivation 2
1.2 Relevance of Fairness in AI 3
1.3 Problem Statement 3
1.4 Definitions 3
1.5 Outline 4

2 Theoretical Background **5**
2.1 Bias 5
2.1.1 Causes of Biasness 6
2.2 Fairness measures 7
2.2.1 Parity-based fairness measures 7
2.2.2 Confusion matrix-based fairness measures 8
2.3 Trade-off 9
2.4 Bias mitigation techniques 10
2.4.1 Pre-processing 10
2.4.2 In-processing 11
2.4.3 Post-processing 11
2.5 Binary Classification Algorithms 11
2.5.1 Logistic Regression 11

2.5.2 Gaussian Naive Bayes 13
2.5.3 Decision tree 14
2.5.4 Support vector machine 14

3 Fairness Enhancing Methods **17**
3.1 Classification with No Discrimination 17
3.2 Disparate Impact Remover 18
3.3 Prejudice Remover 18
3.4 Individual Group Debiasing 19
3.5 Threshold Optimizer 20

4 Experiment, Results and Conclusion **21**
4.1 Data 21
4.1.1 Data cleaning 21
4.1.2 Data profiling 22
4.1.3 Data Pre-processing 23
4.2 Measures 24
4.3 Choice of baseline algorithm 25
4.4 Evaluating the algorithms 26
4.5 Results 26
4.5.1 Performance of methods 26
4.6 Which processing technique to use? 29

A Definitions **31**
A.1 One-hot encoding 31
A.2 Precision 31
A.3 Recall 32

Chapter 1

Introduction

Nowadays, an increasing number of decisions are taken by machine learning algorithms. They are used in many real-life applications, such as decision-making systems in business and social applications, recommendation systems, face recognition, search engines, and many more fields. The motivation for using artificial intelligence (AI) for decision-making is that algorithms are expected to handle more data more efficiently than humans. Second, algorithms can perform complex calculations. Third, decisions made by algorithms are not subjectively driven, unlike decisions by humans. The decision-making an aspect of machine learning algorithms dramatically impacts people's lives. For example, algorithms are used to predict whether a person will get a job, a credit card, or get a loan, and many more; these decisions play an important role in one's life. Since these machine learning algorithms can significantly affect an individual's life, there is great importance in evaluating and improving the ethics of the decisions made by the algorithms. Recently, a lot of machine learning algorithms have been discriminatory against a particular group of the population. One of the most common examples is from the criminal justice field, where recent revelations from ProPublica have shown that the United States judicial system algorithm had falsely predicted future criminality among blacks at twice the rate it predicted for white people [JLA23]. Some other infamous cases are: algorithms used to hire people by Amazon, which showed that it discriminates against women [Kod19], and the algorithm used by Google's ad-targeting proposed higher-paying executive jobs more for men than women [AD15]. We consider the above examples biased/unfair because they discriminate against people based on "sensitive attributes," like sex or race. Legally, no one can be discriminated against based on these sensitive attributes [Uni18].

For this reason, bias-mitigation algorithms have been developed to improve algorithmic fairness. The algorithms are supposed to be unbiased, but unfortunately, the data used to train the machine learning models contains historical biases. At many points, like data collection due to human interference, the data gets biased, resulting in biased machine learning algorithms. Over the last decade, many papers have proposed investigating how an algorithm came to a decision and whether the decisions are fair [MMS+21].
This is, however, a difficult research area because the concept of a 'fair algorithm' is ambiguous. Should the outcomes be the same for all groups of people? Or should similar people get a similar outcome? A different interpretation of what is fair has led to numerous fairness measures like Disparate Impact, Equal opportunity, etc. [VR18]
Based on different definitions of fairness, different machine-learning algorithms have been proposed. These algorithms tend to mitigate biases by either transforming the data or by changing/modifying the algorithm or changing the predictions from an algorithm to improve the fairness of the classifier concerning the chosen fairness measure [HAÅ+20].
In this thesis, we are investing the possibility of finding which bias mitigation process is more suitable for the Recidivism dataset.

1.1 Motivation

There have been many papers that have proposed many bias mitigation techniques to improve the fairness of a classifier. There are three types of bias-mitigation methods: First, pre-processing methods transform the data before feeding it into the algorithm. Second, in-processing techniques, by modifying the algorithm. Third, post-processing techniques, by changing the predictions. However, finding which bias-mitigation algorithm will give the best fair outcome takes work for a given dataset. Various issues complicate this decision.
First, different techniques tend to achieve a different notion of fairness. Therefore, knowing how other methods perform on other fairness measures is essential.
Secondly, various techniques were tested on different pre-processed datasets make it difficult to compare the performances of the bias-mitigation techniques, as it is observed that the results are susceptible to small changes in the data or its pre-processing methods. [FSV+19]

In this thesis, we will discuss the performances of some of the bias mitigation algorithms on recidivism data for different fairness measures.

1.2 Relevance of Fairness in AI

The relevance of fairness in Artificial Intelligence (AI) algorithms is significant because AI algorithms are increasingly being used to make decisions that impact people's lives, such as hiring, loan approvals, and criminal justice. If these algorithms are biased or unfair, they can perpetuate and even exacerbate existing social inequalities and discrimination.
In addition to being ethically important, fairness in AI algorithms are also crucial for building trust in these systems. If people perceive AI as biased or unfair, they are less likely to accept its decisions and recommendations, which could ultimately lead to the failure of AI initiatives.
Therefore, it is essential to prioritize fairness in AI algorithms to ensure that they are beneficial to society as a whole, rather than perpetuating or exacerbating existing inequalities.

1.3 Problem Statement

This thesis focuses on comparing the different fairness-enhancing techniques and find the best method that maximizes the fairness and minimizes the trade-off between accuracy and fairness for the recidivism dataset.

1.4 Definitions

Below are some definitions and notations that will be often used in the following chapters.

1. Bias: refers to the systematic error or tendency of favoritism towards a particular group [MMS+21].
2. Fairness: absence of bias based on personal characteristics [MMS+21].
3. Favourable label: desired outcome, receiving this label will benefit the individual [HAÅ+20].
4. Privileged group: the group that is advantaged by receiving the favorable outcome more often than others [HAÅ+20].

5. Group membership: whether an individual belongs to a privileged or unprivileged group based on protected attributes. G = 0 is the unprivileged group, and G = 1 is the privileged group.

6. Protected/Sensitive attribute: = features that are not allowed to discriminate on, like sex, race, age [CH20].

7. S: Set of protected attributes, where for every s ε S, $s = 1/0$ denotes the privileged/unprivileged groups according to s.

8. X: The non-sensitive feature that is used for classification.

9. Y: Actual label/outcome, according to the data. Where Y = 1 is the favorable label, and Y = 0 is the unfavorable label.

10. $\hat{Y}$: Predicted label by the classifier.

11. $\hat{y}$: Predicted probability score. The value lies between 0 and 1.

1.5 Outline

The subsequent chapters of this thesis are structured as follows. Chapter 2 covers the theoretical background, explains the biases, fairness measures, the trade-off between accuracy and fairness, different techniques used for bias mitigation and the different machine learning algorithms used in this thesis. Chapter 3 describes the fairness-enhancing methods in detail. The set-up of these experiments, and the results and conclusion are discussed in Chapter 4.

Chapter 2

Theoretical Background

2.1 Bias

Bias in algorithms refers to a systematic error or deviation that occurs in the results or predictions produced by an algorithm resulting in unfair treatment or discrimination of specific individuals or groups [FN96]. Every machine learning classifier algorithm has to decide whether an instance will get a favorable or unfavorable outcome. Still, these decisions should be made on reasonable grounds without giving an unfair advantage to any specific individual or group. People should not be discriminated against on sensitive characteristics such as sex, race, age, religion, etc.
A simple way is to remove the sensitive attributes before feeding data into the algorithm. This will work, but it is not always sufficient to eliminate bias.
Firstly, in some cases, sensitive attributes may be correlated with other non-sensitive attributes in the dataset. Removing the sensitive attributes may also remove important information the algorithm needs to make accurate predictions.
Secondly, bias can also arise from how the data is collected or labeled, and removing sensitive attributes does not address these underlying issues. For example, if a dataset is biased towards a certain group of people, removing sensitive attributes will not address this bias.
Instead of simply removing sensitive attributes, it is important to consider the broader context and potential sources of bias in the data and to implement a range of techniques to mitigate these biases. These might include techniques such as data augmentation, algorithmic transparency, and fairness constraints. It is also important to regularly monitor and evaluate the

algorithm's performance to ensure that it is not perpetuating bias.
For example, men and women in health care might require different treatment due to biological differences. Treating men and women similarly for a certain disease may have negative impacts but treating them differently can show positive results.

2.1.1 Causes of Biasness

Biases can exist or arise at different places in the machine learning pipeline through many ways:[MMS+21].

- Historical Bias is caused due to existing historically biased human decisions, erroneous reports, inaccurate measurements, or other reasons. Machine learning algorithms are designed to replicate the trends already present to predict results.
- Biases brought on by missing data, such as missing values or sample/selection biases, leads to a class imbalance in the datasets.
- Biases resulting from algorithmic goals that favor majority groups over minority by minimizing overall aggregated prediction errors.
- Biases brought by 'proxy' attributes. Sensitive attributes, such as color, gender, and age, distinguishing between privileged and underprivileged groups, are often inappropriate for decision-making. Non-sensitive attributes called proxy attributes can be used to obtain sensitive attributes. If the dataset includes proxy attributes, the machine learning system may implicitly make decisions based on the sensitive qualities while disguising its actions by using the seemingly legitimate proxy attributes [BS16].

For example, a bank uses a machine learning algorithm to decide whether to approve or deny loan applications. The bank uses the applicant's zip code as a proxy for their race, assuming applicants from specific zip codes are more likely to be of a certain race. However, this assumption may not always be accurate, and using zip code as a proxy for race could lead to discrimination against certain groups of people.
The data collected now using algorithms will affect the world, making decisions later on using the collected data. This is called a feedback loop [MMS+21].

2.2 Fairness measures

Fairness is a tricky term to use as an intuition of what is fair might differ from one instance to another and also from person to person. On the baseline, it is the need to treat individuals or group without any discrimination. There are several approaches to measure fairness in machine learning, including demographic parity [FFM+15], equal opportunity [HLGK19], equalized odds [HLGK19] and individual fairness [DHP+12].
It is not possible to achieve all kinds of fairness simultaneously; thus, it becomes important to consider the underlying moral assumptions of each fairness measure to decide which measure is most appropriate in a given situation. However, it is important to note that achieving fairness in machine learning is not always straightforward, and there may be trade-offs between fairness and other desirable properties of the algorithm, such as accuracy or efficiency.

2.2.1 Parity-based fairness measures

Parity-based fairness measures are a set of metrics used to evaluate fairness in machine learning models. These measures assess whether a model treats different groups of people similarly without discriminating against any particular group. According to these measures, the predictions should be independent of the group membership.
Let us denote independence by $\perp$:

$$Independence = \hat{Y} \perp G$$

Demographic parity: This measure requires that the proportion of positive outcomes (such as being approved for a loan or being hired for a job) is the same for all groups, regardless of their demographic characteristics such as race, gender, or age.

$$P(\hat{Y} = 1|G = 0) = P(\hat{Y} = 1|G = 1)$$

Demographic parity is measured by calculating the Disparate Impact (DI) ratio. The base rate is defined as the percentage of favorable outcomes. DI ratio is calculated by dividing the base rate of the underprivileged group by the privileged group.

$$DI(ratio) = \frac{P(\hat{Y} = 1|G = 0)}{P(\hat{Y} = 1|G = 1)}$$

DI tending to 1 represents more similar rates across groups and, therefore, more fairness. According to US laws, the ratio should be at least 0.8 of the rate for the group with the highest base rate, called the ”80 percent rule” [FFM$^+$15]. If any dataset has a DI ratio of less than 0.8 is said to have a disparate impact. The assumption is that any group's chances of a favorable outcome are the same. Disparities between the base rate are thus considered a sign of biases.
However, it is also important to note that if demographic parity is not met, it does not always mean that there is bias. As different datasets need not satisfy a particular fairness measure. For example, consider the adult dataset. This dataset predicts whether an individual's annual income is above or below 50,000$. Women are often predicted to receive less than the threshold salary. This result shows clear biases towards fellow men colleagues. But, looking further into the dataset reveals that the women have fewer working hours. So, in this case, the different income levels for the two groups can be explained by an actual difference in the features of these groups.
This example shows that demographic parity is not always the desired measure of fairness.

2.2.2 Confusion matrix-based fairness measures

Confusion matrix: A confusion matrix is a table that lists how well a machine learning model performed when applied to a binary classification issue. For a certain set of data, the matrix displays the number of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) that the model has predicted.
The confusion matrix is usually represented as a 2x2 table, with the predicted labels of the model on one axis and the actual labels of the data on the other axis.

Output	Predicted Positive	Predicted Negative
Actual Positive	TP	FN
Actual Negative	FP	TN

Table 2.1: Confusion matrix

Measures based on the confusion matrix follow the ”separation” crite-

rion, according to which the predictions should be independent of the group membership conditioned on the actual labels [CH20].

$$separation = \hat{Y} \perp G|Y$$

True positive rate (TPR), also known as sensitivity or recall, is a performance metric used to evaluate the effectiveness of a binary classification model. It is defined as the ratio of the true positive (TP) predictions to the actual positive instances in the dataset:

$$TPR = \frac{TP}{TP+FN}$$

False positive rate (FPR), is a performance metric used to evaluate the effectiveness of a binary classification model. It is defined as the ratio of the false positive (FP) predictions to the actual negative instances in the dataset:

$$FPR = \frac{FP}{FP+TN}$$

Equalized Odds: For EO, both the true and false positive rates should be equal across the groups. Mathematically formulated as [HPS16]:

$$P(\hat{Y} = 1|Y = 1, G = 0) = P(\hat{Y} = 1|Y = 1, G = 1)$$

$$P(\hat{Y} = 1|Y = 0, G = 0) = P(\hat{Y} = 1|Y = 0, G = 1)$$

Equal Opportunity: This requires a true positive rate to be equal across groups. Mathematically formulated as [HPS16]:

$$P(\hat{Y} = 1|Y = 1, G = 0) = P(\hat{Y} = 1|Y = 1, G = 1)$$

Individual Fairness: This aims to treat similar individuals similarly. Mathematically formulated as [DHP+12]:

$$P(\hat{Y}^{(i)} = y|X^{(i)}G^{(i)}) = P(\hat{Y}^{(j)} = y|X^{(j)}G^{(j)}); \quad if \quad d(i,j) \approx 0$$

where i and j are two different individuals, $X^{(.)} and G^{(.)}$ denotes the non-sensitive attributes and sensitive attributes of the referred individual. d(i,j) is the distance metric.

2.3 Trade-off

Fairness and accuracy are often considered desirable properties of machine learning models, but in many cases, it isn't easy to achieve both at the same

time. This is because fairness and the accuracy is often in tension with each other, meaning that optimizing for one can come at the cost of the other.
One reason for this tension is that fairness often requires treating different groups or individuals differently while accuracy requires treating everyone the same. For example, consider a credit scoring model that aims to predict the likelihood of a borrower defaulting on a loan. If the model is trained using historical data that reflects biased lending practices, it may learn to discriminate against certain groups, such as minorities or low-income individuals, even if those groups are just as creditworthy as others. To address this, the model may need to be adjusted to give more weight to certain features or groups, which could reduce its overall accuracy.
Another reason for the tension between fairness and accuracy is that fairness is often subjective and context-dependent. Different people may have different ideas of what constitutes fairness, and what is considered fair in one context may not be fair in another. This can make it challenging to optimize for fairness in a precise and objective way, which can, in turn, affect the model's accuracy.
In summary, fairness and accuracy can be challenging to achieve simultaneously because they often require different approaches and can be in tension with each other. Machine learning practitioners need to consider the trade-offs between carefully these two goals and make informed decisions based on the specific context and requirements of the problem at hand.

2.4 Bias mitigation techniques

Recently, many methods have been proposed to increase the fairness of machine learning algorithms. These methods are categorized into three categories pre-processing, in-processing, and post-processing.

2.4.1 Pre-processing

Pre-processing techniques include the methods that involve changing the data for training before feeding into the machine learning algorithm to ensure that the model is trained on unbiased data. There are different methods included in this technique: As disparate Impact Remover, Learning Fair Representation, Optimized Pre-processing, Reweighing, etc. Some methods involve changing labels close to the decision boundary [KC12]. The values

close to the boundary are most prone to be wrongly classified. Some methods involve changing the attribute representations and distribution to make the training data fairer before using the data for training [CWV+17].

2.4.2 In-processing

In-process techniques include methods that involve changing the machine learning algorithm itself. There are different methods included in these techniques like Prejudice Remover, Adversarial Debiasing, Grid Search Reduction, etc. Some methods suggest adding constraints to the machine learning algorithm that call for satisfying a proxy for EO [WGOS17] or DI [ZVRG17]. Some suggest adding a penalty term into the objective function that enforce matching proxies of FPR and FNR [BL17].

2.4.3 Post-processing

Post-processing techniques involve post-processing of the output scores of the machine learning algorithm to produce more fair results. There are different methods included in this technique like Threshold optimizer, Reject Option Classifier, Equality of Opportunity, etc. Some methods suggest selecting different thresholds for other groups, such that the accuracy and fairness are improved [CDPF+17]. [DIKL18] propose a decoupling technique to learn a different classifier for each group.

2.5 Binary Classification Algorithms

This section will discuss the four most common machine-learning algorithms used for binary classification.

2.5.1 Logistic Regression

It is one of the most popular supervised machine-learning algorithms. The first thing to note is that logistic regression is not a regression but a classification learning algorithm. It is called a regression as its fundamental Mathematical formulation is similar to linear regression.
Problem: We want to model y_i as a linear function of x_i . The linear combination of attributes such as $wx_i + b$ is a function whose range is from $-\infty$ to $+\infty$, whereas y_i takes only two values. A function that gives probabilistic

values between $(0,1)$ is the sigmoid function.

$$f(x) = \frac{1}{1+e^{-x}}$$

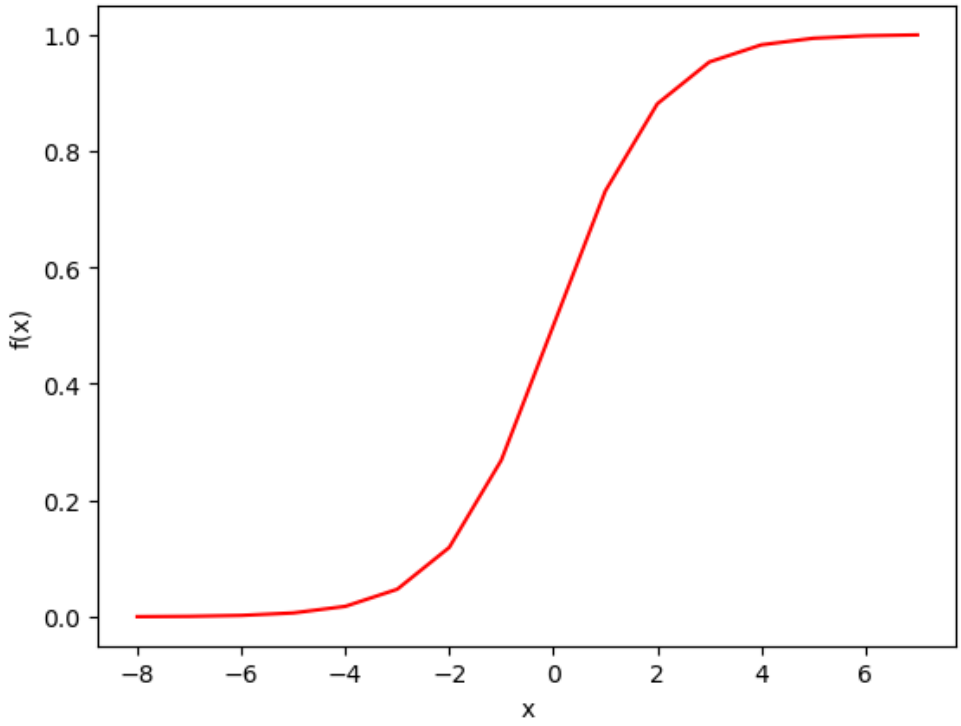

Figure 2.1: Sigmoid function

And applying a threshold to this function, we can classify the output as 0 or 1. Therefore the logistic regression model is represented as:

$$f_{w,b}(x) = \frac{1}{1+e^{-(wx+b)}}$$

Solution: We need to find the best value for w and b. For this, we use the likelihood function and try to maximize it. For example, assume that we have a labeled dataset (x_i, y_i). Start with randomly chosen w and b. Now applying the model $f_{w,b}$ on x. We will get a value between 0 and 1. If y_i is a positive class, the likelihood of y_i being a positive class, according to our model, is given by p. Similarly, if y_i is the negative class, the likelihood of it being the negative class is given by $1-p$. The likelihood function is defined as [Bur19]:

$$L_{w,b} = \prod_{i=1,2...,N} f_{w,b}(x_i)^{y_i}(1-f_{w,b}(x_i))^{1-y_i}$$

The log-likelihood is defined as:

$$log(L_{w,b}) = ln(L_{w,b}(x)) = \sum_{i=1,2,..,N} y_i ln(f_{w,b}(x)) + (1-y_i) ln(1-f_{w,b}(x)) \tag{2.1}$$

We need to maximize the log-likelihood because the log is a strictly increasing function; maximizing this function is the same as maximizing its argument and the solution to this new optimization problem are the same as the solution to the original problem.

2.5.2 Gaussian Naive Bayes

The Naive Bayes algorithm is based on Bayes' theorem, which states that the probability of a hypothesis (such as the target label) given some observed evidence (such as the attributes) is proportional to the probability of that evidence, given the hypothesis, multiplied by the prior probability of the hypothesis. Mathematically, Bayes' theorem can be written as:

$$P(hypothesis|evidence) = \frac{P(evidence|hypothesis)P(hypothesis)}{P(evidence)}$$

where, $P(hypothesis|evidence)$ is the posterior probability of the hypothesis given the evidence, $P(evidence|hypothesis)$ is the likelihood of the evidence given the hypothesis, P(hypothesis) is the prior probability of the hypothesis, and P(evidence) is the probability of the evidence.
In the context of the Naive Bayes algorithm, we assume that the attributes (i.e., the evidence) are conditionally independent given the target label (i.e., the hypothesis). This means that the probability of observing a particular combination of features given a target label can be calculated as the product of the probabilities of each attribute given that target label. Using this assumption, we can calculate the likelihood of the evidence given the hypothesis as:

$$P(A_1, A_2, ..., A_n|hypothesis) = P(A_1|hypothesis)P(A_2|hypothesis)...P(A_n|hypothesis)$$

where, A_i is the i-th attribute.
Once we have calculated the likelihood of the evidence given each possible hypothesis (i.e., each class label) and the prior probability of each hypothesis, we can use Bayes' theorem to calculate the posterior probability of each

hypothesis given the observed evidence. The hypothesis with the highest posterior probability is then selected as the predicted class label for the input data.
In the case of the Gaussian Naive Bayes algorithm, we assume that the likelihood of each feature given the class label follows a Gaussian (normal) distribution. This means that we can estimate the mean and standard deviation of each feature for each class label based on the training data, and use these parameters to calculate the probability density function of each feature given each class label. We can then use these probability density functions to calculate the likelihood of the evidence is given for each hypothesis.

2.5.3 Decision tree

A decision tree classifier is a machine learning algorithm that creates a tree-like model of decisions and their possible consequences. It is a supervised learning method that is commonly used for classification problems. It is an acyclic graph, wherein for each branching node, a specific attribute of the attribute vector (x) is examined. If the attribute's value is below a threshold, the left branch is chosen; otherwise, the right branch is chosen.
Problem: Let us assume that we have a labeled dataset (x, y) , where labels belong to the set $\{0,1\}$. We want to build a model that takes an attribute vector and predicts the label of the given vector.
Solution: In Decision Trees, for predicting a class label for a record, we start from the root of the tree. We compare the values of the root attribute with the record's attribute. Based on the comparison, we follow the branch corresponding to that value and jump to the next node.

2.5.4 Support vector machine

Support vector machine is a machine learning algorithm that aims to find a hyperplane in an N-dimensional space, where N is the number of features in the dataset that distinctly categorizes the data points.

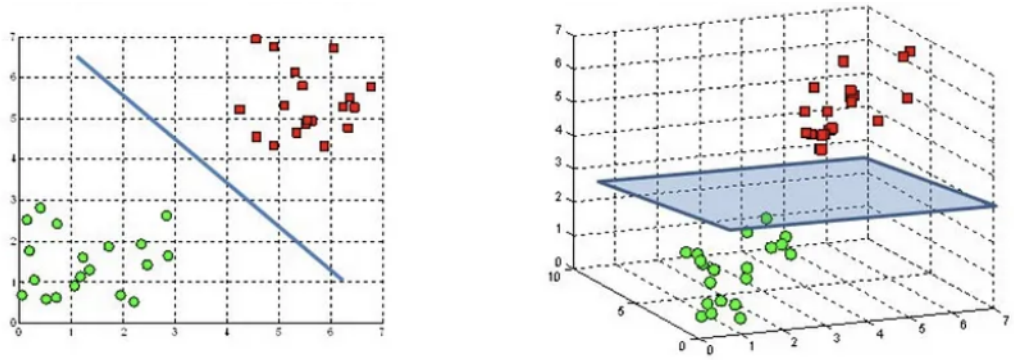

Figure 2.2: Hyperplane in 2D and 3D feature space
[Gan18]

There are a variety of hyperplanes that might be used to split the two groups of data points. Finding a plane with the greatest margin—the greatest separation between data points from both classes—is our goal. Maximizing the margin distance adds some support, increasing the confidence with which future data points may be categorized.

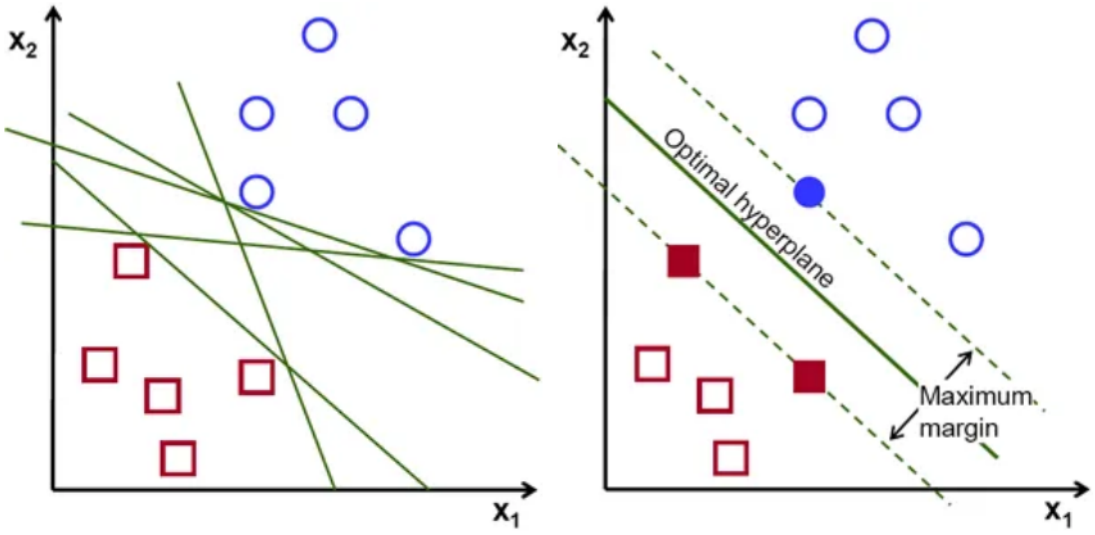

Figure 2.3: Possible hyperplanes and optimal hyperplane
[Gan18]

The data points closer to the decision boundary are called support vectors, as we maximize the margin of the classifier using these. They influence the position and orientation of the hyperplane.

Chapter 3

Fairness Enhancing Methods

In this chapter, different fairness-enhancing methods are explained. The first two methods, CND and DIR, belong to pre-processing techniques, PR to in-processing techniques, and IGD and TO to post-processing techniques. The methods CND and IGD are explained in detail, while DIR, PR, and TO are briefly described as these methods are established procedures and were used directly as packages from AIF360 and FairLearn. AI Fairness 360 (AIF360) is an open-source Python a toolkit developed by IBM provides a comprehensive set of tools and algorithms for addressing bias and promoting fairness in machine learning models. Fairlearn is an open-source Python package developed by Microsoft provides tools and algorithms for measuring and mitigating bias in machine learning models.

3.1 Classification with No Discrimination

Classification with No Discrimination is a pre-processing method of bias mitigation. This method treats a biased dataset with a single sensitive attribute as input for bias treatment.
Bias measure: Bias measure is defined as the difference between the conditional probabilities and denoted by CND_bias. Mathematically it is defined as [Fel15]:

$$CND\_bias = P(Y = 1|G = 1) - P(Y = 1|G = 0)$$

If CND_bias is 0.5, the instances with $G = 1$ have a 50% higher chance of a favorable outcome. This method aims to create a new dataset that minimizes CND_bias. A dataset with CND_bias equal to 0 implies the target column is independent of the sensitive attribute, meaning the probability of

an instance getting a favorable outcome is independent of its group membership. A binary classifier trained on a dataset with $CND\_bias = 0$ will make predictions with no bias.
Algorithm: Consider D as the dataset with biases. Train any binary classification algorithm on D to obtain the probability a score of each instance receiving the favorable outcome. Then divide the dataset, D, into two parts with the following conditions:

- Instances for Promotion (IP): these are the instances with $G = 0$ and $Y = 0$
- Instances for Demotion (ID): these are the instances with $G = 1$ and $Y = 1$

The dataset, IP, and ID are sorted decreasingly and increasingly by probability scores. The first instance of IP is from the underprivileged group, which had the highest probability of being categorized into a favorable class but didn't. Similarly, the first instance of ID is from the privileged group, which had the lowest probability of being classified into a favorable class but didn't. Starting with the first instances from both the dataset IP and ID, iterate through the instances and change each instance's label, until the value of CND_bias is zero. After a finite number of swaps the value of CND_bias approaches 0, and we get a new dataset D' with $CND\_bias \approx 0$.

3.2 Disparate Impact Remover

Disparate Impact Remover (DIR) is used from the AIF360 library. This mitigation algorithm hopes to eliminate the disparate impact by achieving demographic parity. It accomplishes this by altering the feature values to increase equity between the privileged and underprivileged groups. The in-group order is maintained; thus, even if the feature values vary, the ranking of the instances in the data will remain constant for every single feature [FFM$^+$15].

3.3 Prejudice Remover

Prejudice Remover (PR) is used from the AIF360 library. Prejudice means statistical dependence between the protected attribute (S) and target variable

(Y). This method aims to remove prejudice in the dataset by adding a regularization term into the logistic regression (LR) cost function. Consider a dataset, $D = \{(x,s,y)\}$. The conditional probability of Y given a particular group is modeled by $LR[Y|X,S;W]$. We have seen in section 2.6.1 that to improve the logistic fit, the parameter W is optimized by maximizing the log-likelihood function. In this method, we add two additional regularization terms to it. The first regularization term is the L2 regularizer, and the second is the prejudice remover regularizer. The prejudice remover regularizer is defined as [KAAS12]:

$$R_{PR}(D,W) = \sum_{(x_i,s_i)\varepsilon D} \sum_{y=\{0,1\}} LR[y|x_i,s_i;W] ln \frac{P(y|s_i)}{P(y)} \tag{3.1}$$

Therefore, now the new objective function to minimize is given by:

$$-log(L_{w,b}) + \eta R_{PR}(D,W) + \frac{\lambda}{2} \sum_{s\varepsilon S} ||w_s||_2^2$$

where $log(L_{w,b})$ is equation (2.1) and $R_{PR}(D,W)$ is equation (3.1).

3.4 Individual Group Debiasing

Individual Group Debiasing is a post-processing method of increasing individual and group fairness. In its broadest sense, individual fairness means treating individuals with similar features similarly. In contrast, group fairness in its broadest sense means treating each group in a population similarly over a fairness measure. In this method, the instances most prone to individual bias are chosen for a change of predicted label.

Individual bias: Individual bias occurs when the same individual receives a different outcome by changing the sensitive attribute [LRB+19].

$$\hat{Y}(X_i, S=0) \neq \hat{Y}(X_i, S=1)$$

Individual bias score: Individual bias score is the difference between the probability score of an individual when the sensitive feature is changed, keeping all other features the same. It is denoted by b_i, where i denotes the ith individual.

$$b_i = \hat{y}(X_i, S=0) - \hat{y}(X_i, S=1)$$

Average overall instances give the bias score of the dataset, denoted by $\bar{B}$.
Algorithm: The dataset is divided into the train, validation, and test datasets. For this, we first consider a binary classifier, C, and train it on the training dataset. Then using the definition of individual bias score, the bias scores of each individual of the validation dataset with $S = 0$ are calculated. A new dataset is constructed with the bias scores $\{(X_1, B_1), (X_2, B_2)..., (X_n, B_n)\}$, where B_i is calculated by thresholding the bias scores, $b_i's$.

$$B_i = \begin{cases} 1, & \text{if } b_i \geq \bar{B} \\ 0, & \text{otherwise} \end{cases}$$

A new binary classifier, $\hat{C}$, is trained on this new dataset. The new model, $\hat{C}$ is used on the test dataset. The individuals with $S = 0$ from the training dataset are considered. If the predicted value is 1, then the individual is labeled the outcome it would receive if $S = 1$ by model C.

3.5 Threshold Optimizer

Threshold Optimizer (TO) is used from the FairLearn library. This method is used. This approach aims to identify the ideal thresholds for each group to get specified fairness and accuracy. This is based on the equality of opportunity algorithm [HPS16].

Chapter 4

Experiment, Results and Conclusion

In this chapter, we will evaluate the accuracy and fairness measures of fairness-enhancing techniques and determine which algorithm best suits the recidivism dataset. In the following sections, the characteristics of the datasets, the cleaning, profiling, data pre-processing methods, the measures used for evaluating the performances of different methods, the process to choose the baseline machine learning classifier, results, and finally conclusion is discussed.

4.1 Data

All the experiments were performed on one real-world dataset, i.e. the recidivism dataset [JLA23]. The dataset is considered as it is highly impacted by the questions of its validation based on accuracy and fairness in machine learning.

Recidivism dataset: This dataset predicts whether a criminal defendant is likely to re-offend within two years based on different features like demographic features, past criminal record, race, sex, etc. This dataset is widely used in fairness research as the dataset shows disparities over race and sex. The dataset used, ”compas-scores-two-years.csv,” is downloaded from ProPublica. The dataset has 7214 instances and 53 different features.

4.1.1 Data cleaning

The dataset was cleaned by removing the unnecessary columns and rows containing at least one missing value. After this, the dataset obtained had

7214 instances, nine features, and one label.
Features: juvenile felonies count, juvenile misdemeanors count, juvenile other count, sex, race, priors count, age, age category, decile score.
Label: two-year recidivism, describes whether a person did or did not recidivate within two years, where the latter is the favorable label
Summary of the dataset:

	Recidivism dataset
Domain	justice
Instances	7214
Favourable	didn't recidivate
Unfavorable	recidivate
Class balance	1.22
Sensitive attributes	sex and race
Disparate Impact	0.74

Table 4.1: Characteristics of recidivism dataset

4.1.2 Data profiling

Data profiling is a way to understand the data better and identify any potential issues or anomalies that could affect the accuracy and reliability of the machine learning models built on it. It involves examining the data's basic properties, such as its size, distribution, and statistics, as well as identifying missing values, outliers, and anomalies.
The dataset consists of 5819 male instances and 1395 female instances. Base rates of sex and race attributes are presented in Table 4.2.

Sex	Base rate
Male	0.53
Female	0.64

Table 4.2: Base rate for sex attribute

Race	Base rate
African-American	0.49
Asian	0.72
Caucasian	0.61
Hispanic	0.64
Native American	0.44
Other	0.65

Table 4.3: Base rate for race attribute

Combining the sex and race attributes and analyzing the base rates, it shows that African-American males have a much lower base rate than all other groups (disregarding the Asian and Native American females, since these groups are too small with respectively 2 and 4 instances)

Race	Sex	Base rate
African-American	Female	0.62
	Male	0.46
Asian	Female	0.50
	Male	0.74
Caucasian	Female	0.65
	Male	0.59
Hispanic	Female	0.68
	Male	0.63
Native American	Female	0.25
	Male	0.54
Other	Female	0.79
	Male	0.62

Table 4.4: Base rate for all possible sex-race groups

There are two sensitive attributes, sex, and race, in our dataset. A feature called sex-race is added by combining the two sensitive attributes, sex and race. After adding the newly derived attribute, the original sensitive attributes are removed from the dataset. The sensitive attribute is then binarized by singling out the group with the lowest base rate as the underprivileged group and all others as the privileged group. In this case, the unprivileged group consists of African-American men, with a base rate of 0.46. All other groups (race-sex combinations) are privileged, with a base rate of 0.62.

4.1.3 Data Pre-processing

All categorical features were transformed into numerical data using the one-hot encoder technique. The numerical values are scaled between 0 and 1 by using the MinMaxScaler from Sklearn. The favorable label is 0 (i.e., not recidivated).

4.2 Measures

The performances of different algorithms will be evaluated based on fairness and accuracy measures. The accuracy measures include Balanced accuracy score and f1-score. The fairness measures include DI ratio, TPR difference, EO.

1. Balanced accuracy score is a metric to evaluate a classification model's performance. It is calculated as the arithmetic mean of sensitivity (TPR) and specificity (TNR).

$$\frac{1}{2}(\frac{TP}{TP+FN}+\frac{TN}{TN+FP})$$

   The balanced accuracy score ranges from 0 to 1, where a score of 0 indicates that the model is performing poorly, and a score of 1 indicates that the model is performing perfectly. A score of 0.5 indicates that the model performs no better than random chance.

2. F1-score is a metric used to evaluate the accuracy of the classification model. It is a measure of the balance between precision and recall. The F1 score is the harmonic mean of precision and recall, and it ranges from 0 to 1, with 1 being the best possible score. It is calculated as:

$$F1\ \ score = 2[\frac{Precision * Recall}{Precision + Recall}]$$

3. DI ratio: DI ratio is the percentage of favorable outcomes of the unprivileged group divided by the percentage of favorable outcomes of the privileged group. A score between 0.8 and 1.25 is usually already considered to be fair.

4. TPR diff: The TPR difference shows the difference in the True Positive Rates between the privileged and the unprivileged group.

$$TPR_{diff} = |TPR_{priv} - TPR_{unpriv}|$$

   The smaller the difference, the more equal the TPR is for both groups. Thus, a score close to 0 resembles more fairness.

5. EO: The TPR and the FPR differences should be equal for the privileged and the underprivileged group.

$$EO = \frac{1}{2}(|TPR_{priv} - TPR_{unpriv}| + |FPR_{priv} - FPR_{unpriv}|)$$

4.3 Choice of baseline algorithm

For evaluating the performances of different algorithms after implementation of the different fairness techniques, we start with a baseline algorithm that shows the classifications being made without interventions for fairness.
Stability: The stability of an algorithm is defined as the algorithm's performance, each tested on ten random train/test splits. A rectangle is drawn centered on the mean, and a width and height equal to the standard deviation along that measure are plotted.
For this four most common binary classification algorithms were chosen: Logistic Regression, Naive Bayes, Decision tree, and Support vector machine. The result obtained is shown below:

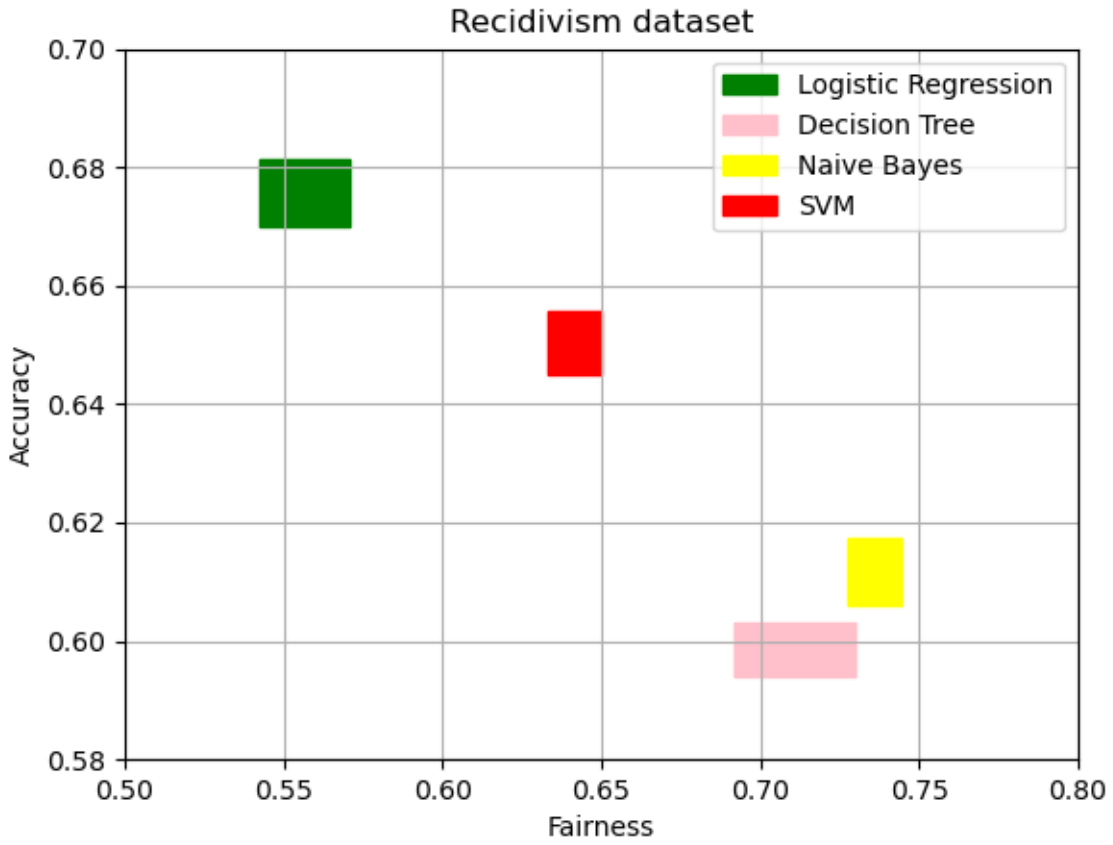

Figure 4.1: Stability

From Figure 4.1, we conclude that SVM and Decision Tree have high standard deviations and thus are not desirable choices. From Logistic Regression and Naive Bayes. Logistic regression is chosen as it has the highest

accuracy, and after the implementation of the fairness algorithm, we can try to achieve higher fairness with Logistic Regression.

4.4 Evaluating the algorithms

For the baseline algorithm, we choose Sklearn's Logistic Regression Classifier. This baseline algorithm is used to see how much the bias mitigation algorithms improve fairness relative to a regular classifier. The logistic regression classifier is also used for making predictions in the pre-processing and post-processing methods.

4.5 Results

This section presents and discusses the result of all the algorithms. We will discuss how the different methods affect different accuracy and fairness measures.

4.5.1 Performance of methods

The results of the bias mitigation techniques on the recidivism dataset are shown below. The result of logistic regression is used as the base and shown via red dotted lines, and solid red lines show the ideal value for the measure on hand. The blue bars represent the Pre-processing The orange bar shows the In-processing technique, and the green bars show the Post-processing technique.

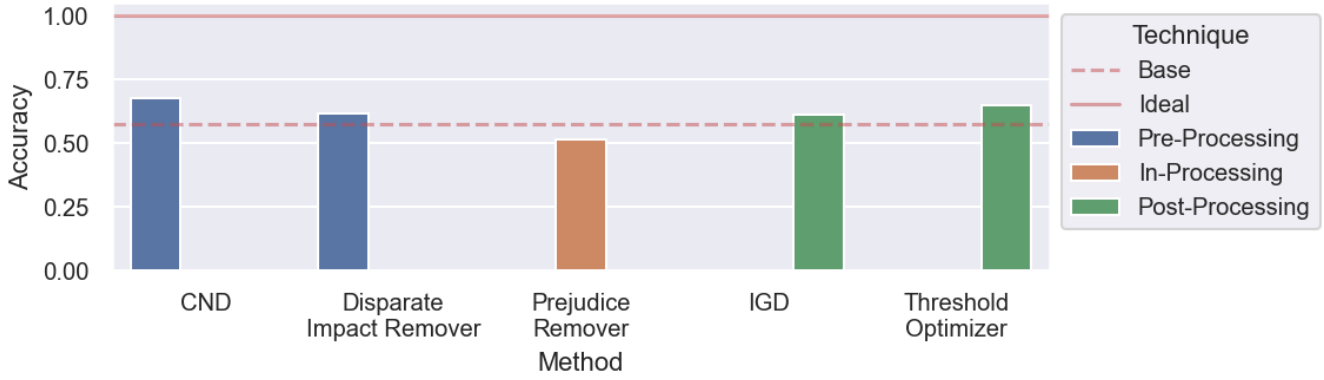

Figure 4.2: Accuracy

In Figure 4.2, the accuracy is observed to be increased from the base in CND, DIR, IGD, and TO, with the highest in CND.

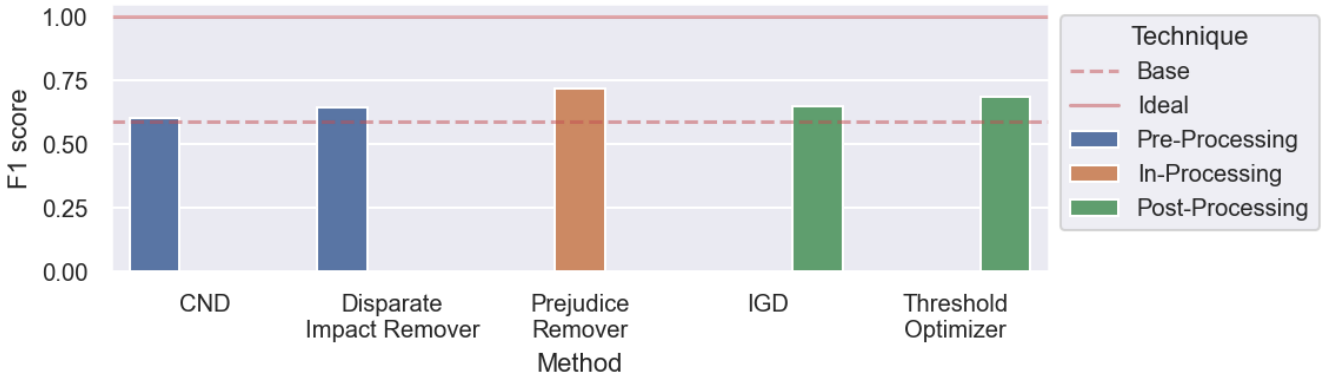

Figure 4.3: F1 score

In Figure 4.3, the F1 score is observed to be increased from the base in all the methods, with the highest in PR.

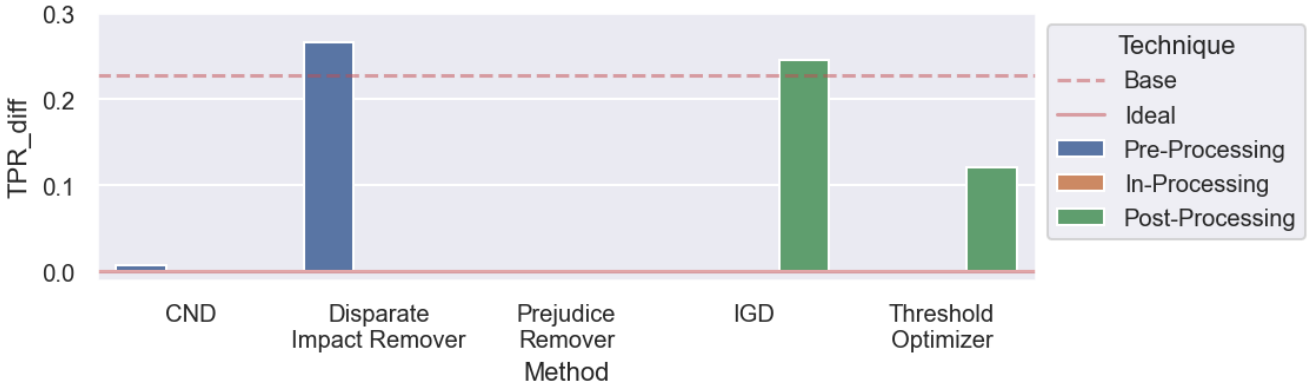

Figure 4.4: TPR difference

In Figure 4.4, the difference between TPR for privileged and unprivileged is observed to be increased from the base and away from the ideal value 0 for DIR and IGD. For CND, PR, and TO, the difference is decreased from the base with CND and PR were performing the best.

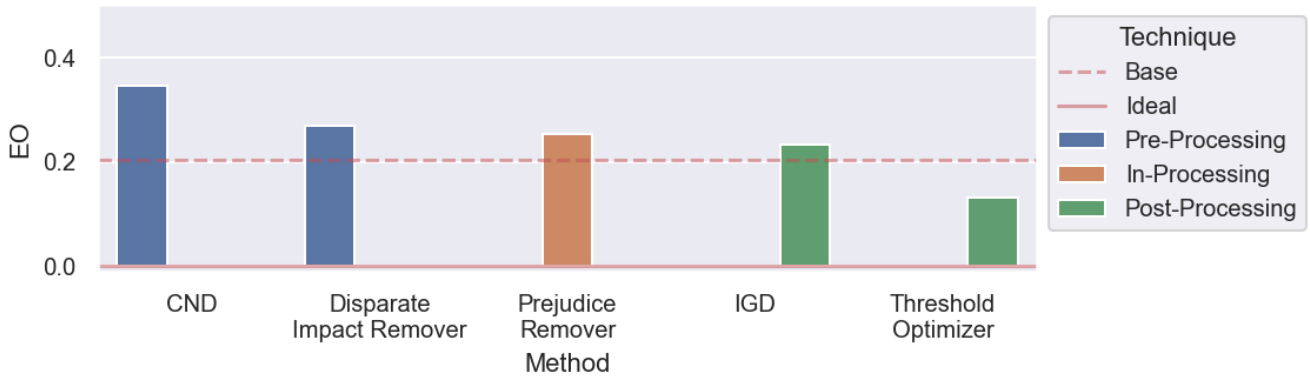

Figure 4.5: Equalized Odds

In Figure 4.5, the EO is observed to be increased and moving away from the ideal value is 0 for all the methods except TO. To is observed to perform the best.

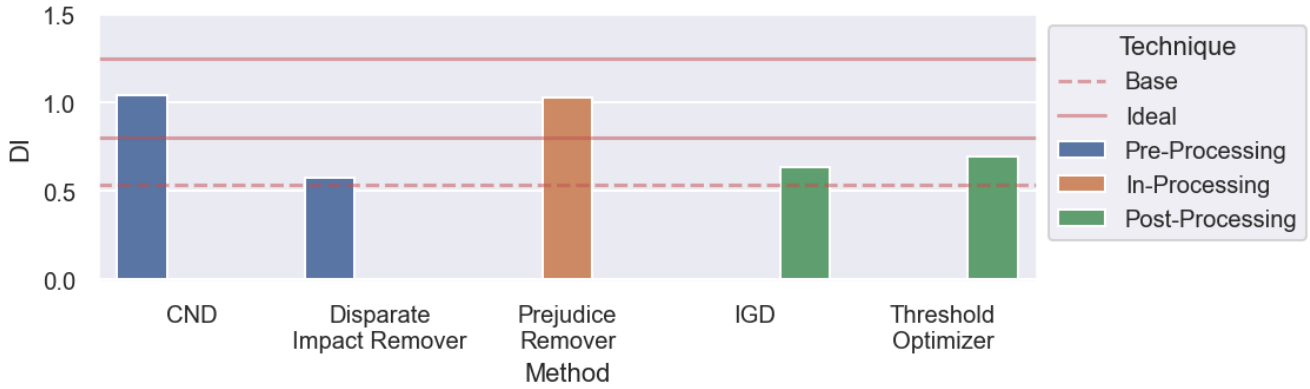

Figure 4.6: Disparate Impact

In Figure 4.6, the DI is increased in every method and towards the ideal range (0.8 to 1.25) for all the methods. Only CND and PR have the value of DI in the ideal range.

Method		Accuracy	F1 score	TPR_diff	EO	DI
Logistic Regression	Mean	0.572	0.586	0.228	0.203	0.533
	std	0.010	0.014	0.041	0.028	0.029
CND	Mean	0.674	0.603	0.007	0.346	1.044
	std	0.295	0.053	0.135	0.201	2.814
DIR	Mean	0.614	0.644	0.266	0.268	0.575
	std	0.010	0.012	0.032	0.026	0.020
PR	Mean	0.514	0.72	0	0.253	1.030
	std	0.003	0.157	0.258	0.140	0.014
IGD	Mean	0.613	0.647	0.246	0.233	0.637
	std	0.011	0.015	0.034	0.029	0.031
TO	Mean	0.648	0.685	0.122	0.132	0.692
	std	0.015	0.015	0.035	0.026	0.035

Table 4.5: Results of the experiments on the recidivism dataset

4.6 Which processing technique to use?

Every technique has its advantages and disadvantages.
Considering pre-processing techniques can be advantageous as it can be incorporated with any machine learning algorithm, and at the same time due to its flexibility of use with any the algorithm makes it difficult to increase accuracy.
Considering in-processing techniques, these are advantageous as they provide a regularization term for the trade-off between accuracy and fairness, and at the same time, due to its complex integration with the machine learning algorithm makes it difficult to mold.
Considering post-processing techniques, these can also be used with all kinds of classification algorithms. Still, inferior results are obtained due to its implication at the very last stage.
[Ham17] in his paper showed that the performance of different techniques varied for different datasets and showed no conclusive evidence of any technique dominating others.
In this thesis, pre-processing technique (CND) performed the best to get admissible fairness and minimize the trade-off between accuracy and fairness.

Appendix A

Definitions

A.1 One-hot encoding

One hot encoding is a technique used to represent categorical variables as numerical data. In this technique, each category is represented as a binary vector where each element in the vector corresponds to a possible category value.
For example, if we have a variable 'color.' with three possible categories: 'red,' 'green' and 'blue,' we can represent each category as a binary vector of length three:

1. 'red': [1,0,0]
2. 'green': [0,1,0]
3. 'blue': [0,0,1]

This technique is called one-hot encoding because only one element in the vector is 'hot' or 'on' (set to 1) for each category, while all other elements are 'cold' or 'off' (set to 0).

A.2 Precision

Precision is the ability of the model to correctly identify positive samples (true positives) out of all samples that it predicted as positive (true positives + false positives).

$$Precision = \frac{TP}{TP + FP}$$

A.3 Recall

Recall is the ability of the model to correctly identify all positive samples (true positives) out of all samples that are actually positive (true positives + false negatives).

$$Recall = \frac{TP}{TP + FN}$$

Bibliography

[AD15] Anupam Datta Amit Datta, Michael Carl Tschantz, *Automated experiments on ad privacy settings.proceedings on privacy enhancing technologies*, 2015.

[BL17] Yahav Bechavod and Katrina Ligett, *Learning fair classifiers: A regularization-inspired approach*, arXiv preprint arXiv:1707.00044 (2017), 1733–1782.

[BS16] Solon Barocas and Andrew D Selbst, *Big data's disparate impact*, California law review (2016), 671–732.

[Bur19] Andriy Burkov, *The hundred-page machine learning book*, 2019.

[CDPF+17] Sam Corbett-Davies, Emma Pierson, Avi Feller, Sharad Goel, and Aziz Huq, *Algorithmic decision making and the cost of fairness*, Proceedings of the 23rd acm sigkdd international conference on knowledge discovery and data mining, 2017, pp. 797–806.

[CH20] Simon Caton and Christian Haas, *Fairness in machine learning: A survey*, arXiv preprint arXiv:2010.04053 (2020).

[CWV+17] Flavio Calmon, Dennis Wei, Bhanukiran Vinzamuri, Karthikeyan Natesan Ramamurthy, and Kush R Varshney, *Optimized pre-processing for discrimination prevention*, Advances in neural information processing systems **30** (2017).

[DHP+12] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel, *Fairness through awareness*, Proceedings of the 3rd innovations in theoretical computer science conference, 2012, pp. 214–226.

[DIKL18] Cynthia Dwork, Nicole Immorlica, Adam Tauman Kalai, and Max Leiserson, *Decoupled classifiers for group-fair and efficient machine learning*, Conference on fairness, accountability and transparency, PMLR, 2018, pp. 119–133.

[Fel15] Michael Feldman, *Computational fairness: Preventing machine-learned discrimination*, Ph.D. thesis, 2015.

[FFM+15] Michael Feldman, Sorelle A Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian, *Certifying and removing disparate impact*, proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining, 2015, pp. 259–268.

[FN96] Batya Friedman and Helen Nissenbaum, *Bias in computer systems*, ACM Transactions on Information Systems (TOIS) **14** (1996), no. 3, 330–347.

[FSV+19] Sorelle A Friedler, Carlos Scheidegger, Suresh Venkatasubramanian, Sonam Choudhary, Evan P Hamilton, and Derek Roth, *A comparative study of fairness-enhancing interventions in machine learning*, Proceedings of the conference on fairness, accountability, and transparency, 2019, pp. 329–338.

[Gan18] Rohith Gandhi, *Support vector machine — introduction to machine learning algorithms*, 2018.

[HAÅ+20] Knut T Hufthammer, Tor H Aasheim, Sølve Ånneland, Håvard Brynjulfsen, and Marija Slavkovik, *Bias mitigation with aif360: A comparative study*, Norsk IKT-konferanse for forskning og utdanning, no. 1, 2020.

[Ham17] Evan Hamilton, *Benchmarking four approaches to fairness-aware machine learning*, Ph.D. thesis, 2017.

[HLGK19] Hoda Heidari, Michele Loi, Krishna P Gummadi, and Andreas Krause, *A moral framework for understanding fair ml through economic models of equality of opportunity*, Proceedings of the conference on fairness, accountability, and transparency, 2019, pp. 181–190.

[HPS16] Moritz Hardt, Eric Price, and Nati Srebro, *Equality of opportunity in supervised learning*, Advances in neural information processing systems **29** (2016).

[JLA23] Lauren Kirchner Jeff Larson, Surya Mattu and Julia Angwin, *Compas recidivism risk score data and analysis*, 2023.

[KAAS12] Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, and Jun Sakuma, *Fairness-aware classifier with prejudice remover regularizer*, Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2012, Bristol, UK, September 24-28, 2012. Proceedings, Part II 23, Springer, 2012, pp. 35–50.

[KC12] Faisal Kamiran and Toon Calders, *Data preprocessing techniques for classification without discrimination*, Knowledge and information systems **33** (2012), no. 1, 1–33.

[Kod19] Akhil Alfons Kodiyan, *An overview of ethical issues in using ai systems in hiring with a case study of amazon's ai based hiring tool*, Researchgate Preprint (2019), 1–19.

[LRB+19] 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*, Icassp 2019-2019 ieee international conference on acoustics, speech and signal processing (icassp), IEEE, 2019, pp. 2847–2851.

[MMS+21] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan, *A survey on bias and fairness in machine learning*, ACM Computing Surveys (CSUR) **54** (2021), no. 6, 1–35.

[Uni18] European Union, *General data protection regulation. the european parliament and the council of the european union*, 2018.

[VR18] Sahil Verma and Julia Rubin, *Fairness definitions explained*, Proceedings of the international workshop on software fairness, 2018, pp. 1–7.

[WGOS17] Blake Woodworth, Suriya Gunasekar, Mesrob I Ohannessian, and Nathan Srebro, *Learning non-discriminatory predictors*, Conference on Learning Theory, PMLR, 2017, pp. 1920–1953.

[ZVRG17] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rogriguez, and Krishna P Gummadi, *Fairness constraints: Mechanisms for fair classification*, Artificial intelligence and statistics, PMLR, 2017, pp. 962–970.