

Fairness in Machine Learning: Ensuring Fairness in Datasets for Classification Problems

Business Informatics Masters Thesis

Begüm Hattatoğlu

6395341

b.hattatoglu@students.uu.nl

First supervisor

Dr. A.A.A. (Hakim) Qahtan

Second supervisor

Dr. Heysem Kaya



Universiteit Utrecht

Department of Information and Computing Sciences

Utrecht University

Netherlands

July 2021

Abstract

Machine Learning (ML) algorithms are used in a wide range of applications, which affected societies either directly or indirectly in daily life. ML algorithms are preferred for many tasks that require complex computations with a massive volume of data due to the better performance compared to humans. Moreover, people have subjective opinions and points of view, which can lead to bias in their decisions. Unfortunately, ML algorithms are not always objective either. Using ML algorithms in several decision-making systems and other services may cause serious discrimination among some groups of people in society. One of the most significant reasons behind the biased predictions of the algorithms for different demographic groups is the imbalanced representation of each demographic subgroup in the population. In this master's thesis, COSCFair (Clustering and OverSampling for Fair Classification) is proposed, which is a framework to ensure fairness among all the subgroups that exist in a dataset without changing the original class labels of the samples. COSCFair consists of clustering, oversampling, and classification components, where the classification component considers the outcomes of the clustering algorithm. The classification component contains an ensemble technique of class label prediction. The experimental results over different datasets that are widely used as benchmarks to evaluate algorithmic fairness show that the COSCFair framework yields consistent improvements in fairness while causing a minimal loss in predictive performance compared to a set of baseline methods.

Keywords— Machine Learning, Fairness, Algorithmic Fairness, Clustering, Oversampling, Classification

Acknowledgements

I would like to spare this section to thank the people who helped me and consistently supported me to complete this work. Without such continuous support and guidance, it would not be possible to work on such a topic and to achieve the resulted quality of work. First of all, I would like to thank and express my gratitude to my first supervisor Hakim Qahatan for his continuous attention at every step of this project, for following up and guiding my progress every week, and for always being open to all of my questions and discussions regarding this project to help me. Without his guidance and help, I could not complete such a scientific project that might have a significant value to help the society and machine learning community. I also would like to thank my second supervisor, Heysem Kaya, for his support for this topic, his interest, and all the valuable feedback he provided for this project. I would like to express my sincere gratitude to Christiaan Bol for his endless support and courage during these two years of my study by motivating me every day to keep doing my best. During the COVID-19 pandemic, he has always been there to encourage me whenever I needed it. Finally, my dear parents deserve endless gratitude: they are the ones who have always supported me at every step of my education throughout my life in every way possible and gave me courage and confidence during my masters with their love and affection. My accomplishments and success are because they have always believed in me.

Contents

Abstract	1
Acknowledgements	2
1 Introduction	5
1.1 Motivation	6
1.2 Research Questions	7
1.3 Glossary	8
1.4 Research Methodology	8
1.5 Outline	10
2 Related Work	11
2.1 Fairness Measures	11
2.1.1 Statistical Fairness Measures	12
2.1.2 Individual Fairness Measures	14
2.1.3 Sub-group Fairness Measures	16
2.1.4 Causal Reasoning Measures	17
2.2 Mitigation Algorithms	18
2.2.1 Pre-processing Algorithms	18
2.2.2 In-Processing Algorithms	21
2.2.3 Post-Processing Algorithms	23
2.3 The Goals in Fairness-Aware Machine Learning	25
2.3.1 Discrimination Detection	25
2.3.2 Discrimination Elimination	27
2.4 Classifiers Used in Algorithmic Fairness	28
3 Theoretical background	30
3.1 Problem Statement	30
3.2 Improving Fairness Theoretically	32
3.2.1 Examples of Improving Fairness and the Effects on Performance Measures	34
3.3 Clustering	35
3.4 Oversampling	37
3.5 Classification	37
4 The COSCFair Framework	42
4.1 Data Preparation	43
4.2 Fuzzy c-means	44
4.3 SMOTE Oversampler	46
4.4 Classifiers for Prediction	47
4.5 Recommended Strategies	48

4.5.1	Using Single Classification Model	48
4.5.2	Using Cluster Membership for Prediction	48
4.5.3	Using Weighted Cluster Memberships for Prediction	48
5	Evaluation	51
5.1	Datasets	51
5.2	Measures	53
5.2.1	Fairness Measures	53
5.2.2	Prediction Performance Measures	53
5.3	Baseline Mitigation Algorithms	54
5.4	Experimental Setup	54
5.5	Analysis and Results	55
5.5.1	Comparing Classification Strategies for the Optimal Framework Construction	55
5.5.2	Effect of COSCFair on All Subgroups	57
5.5.3	Comparison of COSCFair with Baseline Methods	59
6	Conclusion	63
6.1	Summary	63
6.2	Answers Found for the Research Questions	64
6.3	Limitations	64
6.4	Ethical Considerations	65
6.5	Future Work	66
A	Appendix: More Tables From Experimental Results	72
A.1	Ratio Tables of Strategy 3	72
A.2	Other Tables from the Experiment 2	73

Chapter 1

Introduction

The usage of machine learning in a wide diversity of domains has affected everyone’s daily life to varying extent. Every day, more and more machine learning algorithms are used for decision-making in business and government systems or applications [59]. They are used in recommender systems, advertisements, hiring systems, and many other fields that provide services to people or companies. There are several reasons behind the widespread usage of machine learning algorithms. One of these reasons is that the algorithms perform better than humans in such decision-making tasks. For example, machine learning algorithms can handle big volumes of data for complex computational tasks in a significantly shorter time compared to humans. Besides, while people have subjective opinions which can lead to bias in their decisions while performing a task, machine learning algorithms do not have any “opinions”, thus they are supposed to be objective in decision making. Since many of these machine learning systems can affect people’s lives with the decisions they make regarding a job application, loan approval, healthcare-related risk approval, or legal decisions about people, it is very substantial that the algorithms make fair decisions.

Unfortunately, machine learning algorithms are not always objective. Using these algorithms in several decision-making systems and other services may cause serious discrimination among some groups of people in society. For example, it was revealed that Amazon’s algorithm for free same-day delivery made racially biased decisions while choosing which neighborhoods to provide this service [65], [48]. In another investigation, it was found that one of the job search portals that is called Xing in China uses an algorithm to rank the job applicants had a significant gender bias against women [47]. Furthermore, other studies revealed existing gender and race bias while showing advertisements in Google’s search engine results, such as showing less high-paying job ads for females when they use the search engine, or showing arrest records as a recommendation when a name with African-American origin is searched [21, 66]. It has been also found that bias exists in word embeddings in natural language processing (NLP), namely, gender bias that associates the word “she” more with words like homemaker and nanny, while it matches the “he” word with architect or computer programmer [10]. Such biased stereotyping in word embeddings can cause the systems using NLP to become unfair with the services they provide.

Another domain in which several bias cases were revealed was facial recognition. For example, Google Photos had tagged two African-American people as “gorillas” with its visual recognition algorithm [5], and Nikon’s smart cameras with facial recognition algorithm could not recognize the blinks of Asian people [58]. Furthermore, it was found that Microsoft and IBM’s facial recognition softwares that are used for gender classification were performing better for

lighter faces than darker faces, male faces than females faces, and finally, they performed worst at the African-American women faces [11]. Lastly, probably one of the most famous and influential findings in this domain was the identification of racial bias in the COMPAS recidivism estimation tool, which is used in many courts of the United States. It was found that the tool was discriminating African-American males more by predicting a higher recidivism risk for them compared to white male offenders [3]. According to the risk level of the defendants, courts can keep the defendants in custody until the trial. Also, the judges considered these risk scores while deciding on the severity of the sentence of defendants. Given the example cases above, we see that existence of bias in machine learning algorithms can significantly affect people’s lives economically (job applicants ranking tool) and socially (COMPAS tool).

There are several reasons why bias exists in machine learning. It could emerge due to the historical bias or prejudice reflected in the decision variable (class labels) of a dataset. Another reason could be due to limited features in a dataset that could be less informative about the population, or due to the existence of an attribute that is directly related to the sensitive attributes, such as race and gender, even when these sensitive attributes are not used to train the algorithms. Furthermore, an imbalanced representation of different demographic groups in a dataset can also cause bias. If a demographic group is represented with many more negative outcomes than positive while another group is represented the other way around, this will cause having a biased dataset. These potential problems in a dataset cause machine learning algorithms to keep the existing bias and reflect it, or even sometimes exacerbate the existing bias in their predictions.

In order to prevent bias in machine learning, researchers have come up with several different fairness measures around fairness-aware machine learning. Not only the measures to quantify fairness but also different algorithmic approaches have been developed by researchers to eliminate the existing bias or mitigate it under a certain level. Unfortunately, there is no consensus on how to measure bias and mitigate or eliminate it to ensure fairness yet in the literature. However, most of the approaches are mitigating the bias based on only a single sensitive attribute even though there are multiple sensitive attributes in a dataset. This can cause the calculated measures to indicate less bias than what actually exists. For example, in the whole dataset of COMPAS, there are both race and gender as sensitive attributes. If only race attribute is considered, the African-American females will make existing bias look less than it actually is due to the fact that females have more positive outcomes than males, which will cause the dataset to have a disparate impact ratio of 0.78, which is very close to the acceptable ratio thresholds (0.8-1.25) to deem a dataset or a classifier fair. If only gender is considered as the sensitive attribute, then Caucasian males will decrease bias between groups and have a disparate impact ratio of 0.79 which is also very close to the fairness threshold. Finally, if both sensitive attributes are considered, then the disparate impact ratio becomes 0.68 which is much further from the minimum fairness threshold. Thus, being able to investigate multiple sensitive attributes in datasets simultaneously is very substantial to identify and mitigate bias adequately.

1.1 Motivation

An increase in the usage of machine learning algorithms in different domains has increased the importance of the decisions or predictions of these algorithms to be free from any form of bias since they affect the lives of individuals significantly. Even though numerous bias mitigation algorithms are proposed to ensure fairness, there is no consensus on which approach is the best and robust one to follow since the performance of these algorithms can fluctuate across datasets. In addition, the chosen fairness measures to quantify fairness and measure the performance of these algorithms can affect the outcomes significantly. For example, while a couple of fairness measures deem the predictions of a classifier satisfactorily fair, the others

might deem them unfair. Furthermore, most of the mitigation algorithms proposed in the literature can handle only a single binary attribute, which means that they can handle only a limited type of dataset. However, not all the datasets consist of only a single sensitive attribute with binary values. There are only a few mitigation algorithms that can process multiple binary sensitive attributes.

Another overlooked topic in the domain is the imbalance in the number of samples that each group has in datasets. Datasets might have an imbalanced distribution over the groups defined by the sensitive attributes as well as an imbalanced distribution over the samples with positive and negative class labels per group of people. The imbalance in datasets can emerge from biased data collection procedures or not having ample sources to collect a sufficient amount of data from every demographic group. The imbalance between the number of positive and negative class labels and imbalance between the number of samples per demographic group in datasets, and especially under-representation of certain groups can create bias in the predictions of classification algorithms. However, there are only a few studies that focus on solving these problems in fairness.

The problems described above indicate the need to design a pre-processing framework that can handle datasets with multiple sensitive attributes. The framework should eliminate both class and group imbalance in datasets simultaneously to mitigate the bias before training a classifier. This way, the classifier will not carry on or exacerbate existing bias in a dataset. By eliminating any imbalance in the dataset, the framework will be able to satisfy multiple fairness measures in the literature. In this research, pre-processing is the preferred step to mitigate bias since it is classifier-agnostic, which means that after pre-processing, any classifier can be trained on the fair dataset. Furthermore, it is not limited to any specific group of classifiers, unlike the in-processing approaches that are based on altering the training procedure of classifiers to force them to satisfy some fairness measures. The pre-processing approach is not prone to sub-optimal classifier performance either since there is no need to alter the predicted outcomes of a classifier like post-processing methods do.

1.2 Research Questions

The main research goal of this thesis project is to develop a framework to eliminate discrimination in datasets by applying a re-sampling technique in combination with a clustering technique in pre-processing step of the machine learning pipeline and ensure fairness in imbalanced classification.

Thus, the main research question of this thesis is going to be the following:

Research Question: How can we develop a framework to eliminate or mitigate bias in datasets?

To answer the main research question, the following set of research sub-questions will be answered:

On at least 2 out of 3 benchmark datasets:

1. Can the proposed mitigation framework improve at least 3 out of 5 chosen fairness measures calculated on a classifier's predictions by at least 20%?
2. Can the proposed mitigation framework significantly outperform at least two of the baseline bias mitigation techniques in terms of at least three out of five chosen fairness measures without a significant difference in the performance measures?

1.3 Glossary

In this section, the most important and widely used terms in the domain of fairness are defined, which are relevant to this thesis. These terms are going to be used frequently in the upcoming chapters.

- **Fairness:** The absence of any prejudice or favoritism towards an individual or a group based on their intrinsic or acquired traits in the decision-making context [63].
- **Discrimination:** An unjustified distinction of individuals based on their membership in a certain category or group [57].
- **Protected Group:** A group or category of people (identified by the sensitive attributes they have) that are subject of discrimination analysis, who are explicitly protected from discrimination [57]. It is also known as *unprivileged group* in the literature.
- **Privileged Group:** A group or category of people who have the dominant and favorable position or outcomes in discrimination analysis.
- **Protected Attribute:** The attributes that are potentially used to treat individuals unfairly and prohibited to be used by the human rights laws, such as gender, sexual orientation, race, ethnicity, skin color, language, social origin, religion or belief, disability, marital status, or age [57]. It is also named as *sensitive attribute* in the literature.
- **Disparate Treatment:** The application of different rules or practices to comparable situations to treat one person less favorably than another based on the subject's sensitive attributes [57]. It is also named as *direct discrimination* or *systematic discrimination* in the literature.
- **Disparate Impact:** A neutral verdict, criterion, or practice that takes into account personal attributes in relation to race, gender, and other discriminatory grounds and results in unfair treatment of a protected group [57]. It is also named as *indirect discrimination* in the literature.
- **Favorable Label:** The desired decision outcome value that provides an advantage for individuals.
- **Bias Mitigation Algorithm:** The algorithms which aim to eliminate or partially eliminate (mitigate) the existing discrimination in the dataset or whole system.

1.4 Research Methodology

The main goal of this research is to introduce a new framework to the current body of knowledge that ensures fairness in machine learning. The machine learning pipeline consists of data collection, data cleaning or pre-processing, model training, testing, and finally the visualization of the results. Thus, machine learning can be considered as a special sub-category under data mining since the steps of data mining are including the steps of a machine learning pipeline. For successful data mining projects, Cross-Industry Standard Process for Data Mining (CRISP-DM) [70] is the most preferred research methodology due to its iterative structure. Since we are conducting research that focuses on ensuring fairness in machine learning using algorithms, CRISP-DM is the best matching methodology that is appropriate for research aiming to develop such a framework including the whole machine learning pipeline. Therefore, the steps of the CRISP-DM methodology will be followed in order to obtain the best framework to ensure fairness.

CRISP-DM contains six main steps, which are business understanding, data understanding, data preparation, modeling, evaluation, and deployment. Even though these steps follow one another in the respective order, it is possible to loop back iteratively in between these steps.

Business Understanding. It is important for the companies to show that the applications they use which processes the data of individuals do not contain any discrimination. Fairness in machine learning is the main domain in this project, hence, it is essential to analyze the techniques that are developed to quantify and mitigate (un)fairness. Different kinds of fairness measures and algorithms that are proposed to eliminate or mitigate bias in different steps of the machine learning pipeline must be carefully studied so that this understanding can be used to implement a robust and successful mitigation algorithm for the framework. Using fairness measures is also the way to measure the performance of the mitigation algorithms which aims to see if the algorithm could remove or at least decrease the amount of bias that was initially identified. To measure the amount of bias existing in a dataset, demographic parity, equalized odds, calibration, and an individual fairness measure will be used.

Data Understanding. In order to test and compare the performance of the new framework, it is important to identify which datasets are available, what kind of characteristics the datasets have, and how these datasets can be used properly. There are several publicly available datasets that are widely used in many studies in this domain. Some of these datasets are German credit, Adult census income, and COMPAS recidivism datasets. Each dataset has its own sensitive attribute(s), or variable(s), which are mostly race, gender, and/or age. While age might be the sensitive attribute causing bias in one dataset, gender might be the sensitive attribute to cause bias instead of age in another dataset. Therefore, it is necessary to examine and identify different properties of the datasets such as which are the sensitive attributes and whether they contain categorical or continuous values.

Data Preparation. After data understanding, it is needed to prepare the datasets in an appropriate format before giving them as inputs to train the algorithms. Typical preparations are handling missing values, removing the attributes that are not needed, and transforming the values of some attributes into another format. In some cases, continuous variables must be partitioned in a dataset so that the algorithms can handle them. All the datasets are available in CSV format, and they are handled together with the implementation of the algorithms using Python programming language. In order to test the performance of the framework, the datasets are split into training and test sets.

Modeling. The aimed mitigation algorithm and the whole fair machine learning framework is implemented in this phase. The most appropriate mitigation strategies that are developed to reach the fairness goal are tested, compared, and then the chosen algorithms are combined, modified, and improved to create one final model which is a better mitigation algorithm and can robustly mitigate bias. After this investigation, fuzzy c-means algorithm is chosen for the clustering component, SMOTE algorithm is chosen for the oversampling component and the weighted cluster membership strategy is chosen for the classification component which includes class label prediction at the end of the framework.

Evaluation. The mitigation algorithm implemented in the modeling phase is evaluated by comparing its performance in both fairness and accuracy using different datasets. If the initial implementation has satisfactory performance, then it can be deployed in the next phase. However, if it does not yield satisfactory results, it is needed to go back to the business understanding phase to understand what went wrong, what can be improved and how can the improvement be achieved. Evaluation is set up with three main experiments, which includes the comparison of three different mitigation algorithms with the proposed framework using three different datasets.

Deployment. After it is certain that the implemented algorithm yields satisfactory results, it is deployed to be used in business practice regularly. The deployment includes not only implementing the model into daily practices but also includes providing maintenance and periodical checks. This research contains a deployment via an open-source GitHub repository after the evaluation phase.

1.5 Outline

This thesis project consists of six main chapters, which are the introduction, related work, theoretical background, the COSCFair framework, evaluation, and finally conclusion and future work chapters. After the introduction chapter, the Related Work chapter contains the previous work that has been done to provide the related background information under four sections. These sections are the fairness measures, which introduces the different kinds of measures to quantify fairness and their underlying points of view, mitigation algorithms, which discusses various techniques aiming to ensure fairness in the different stages of the machine learning pipeline, types of goals in fairness-aware machine learning, and classifiers used in algorithmic fairness. The theoretical background chapter introduces the problem statement formally, explains how bias mitigation is achieved, and introduces how fairness and performance measures are used as criteria to investigate the bias. In the COSCFair chapter, the structure of the framework and all of its components are explained in detail. These components include clustering, oversampling and ensembled classification. The evaluation chapter shows the setup of each experiment and all the results obtained during these experiments. Finally, in the conclusion and future work chapter, there are the final remarks, limitations, and future directions to investigate further to improve the COSCFair framework.

Chapter 2

Related Work

In this chapter, we discuss the relevant research from the literature. Various fairness measures, different mitigation algorithms, various fairness-aware machine learning tools, and the most commonly used classifiers in the algorithmic fairness domain are discussed in their respective sections to provide a detailed background information.

2.1 Fairness Measures

There are various fairness measures in the literature that are proposed to quantify the amount of fairness/bias in a dataset or the outcomes of a system/algorithm. Fairness measures can be used to measure bias in datasets or in the predicted outcome of classification algorithms. For example, the measures can be used on datasets in pre-processing stage before they are given to an algorithm for training, or they can be used on the predicted outcomes of an algorithm in the post-processing stage. This variety of approaches come into existence due to different points of view or improvements to previously proposed measures in the literature. However, there are two fundamental viewpoints or approaches to the notion of fairness due to its nature, which are axiomatically defined as “What You See Is What You Get” (WYSIWYG) and “We Are All Equal” (WAE) [26].

Friedler, Scheidegger, and Venkatasubramanian [26] formalized fairness as the mapping from “construct space” (the space that captures the whole population’s meaningful attributes) to “observed space” (the space that can capture only some observable part of the construct-space), and then to the “decision space” (the space of outcomes that are predicted). According to the former worldview (WYSIWYG), the difference between the construct space and the observed space should be smaller than a certain error threshold, which means that the collected dataset successfully reflects the original distribution and characteristics of the population. If there are differences observed among groups in the collected dataset, those differences between the individuals of different groups do exist in reality. On the other hand, the latter worldview (WAE) says that all groups are essentially the same, and there is no significant difference between these groups that could be base on any discriminatory characteristics. Any observed differences in the observed space are inaccuracies and it means that there is already a structural bias in the collected dataset. The measures regarding demographic parity-related measures reflect the WAE worldview, whereas the equality of odds-related measures reflect the WYSIWYG worldview. The rest of the measures do not have a certain choice of worldview but they are located in between these two ideas. These measures are called statistical fairness measures and they will be explained in more detail in the following subsection.

2.1.1 Statistical Fairness Measures

The first category of fairness measures is called *statistical fairness measures*. They can also be seen under the name *associational fairness* measures [62] in the literature. These measures are based on the statistical proportions of some limited number of groups defined by one or more sensitive attributes in the dataset. The main idea behind the statistical fairness measures is that there must be some parity of a statistical measure which should be obtained for all groups. However, the measure does not have to be very strict. An approximate threshold can be accepted such as allowing some small amount of difference in the measurements between the demographic groups existing in datasets [17]. Since these statistical measures are always applied to groups of people identified in the dataset, they could also be called group-based measures. Some statistical fairness measures consider only the predicted outcome, some other measures consider both predicted and actual outcomes, or predicted probabilities and actual outcomes based on the dataset [68].

The most straightforward and most intuitive definition of fairness is based on the measures that only consider the original class labels in a dataset or the predicted outcomes of a classifier. The first measure with this notion is called *demographic parity*, which is also known as *statistical parity* [23] or *group fairness* [68]. A classification algorithm is fair based on this measure if both protected and unprotected groups based on a sensitive attribute have the same probability rate to be assigned to the positive outcome [23], [41]. It means that the sensitive attribute and the outcome should be statistically independent of each other for a dataset or predicted outcomes to be considered fair. For instance, if the sensitive attribute of a dataset is race and the values for this attribute are white and non-white, then people from both of these groups should have the same opportunity to get a positive decision. Demographic parity is not a good measure to use alone that can identify discrimination in every situation because it cannot distinguish how the instances from privileged and unprivileged groups are chosen for the positive outcome. In other words, demographic parity cannot detect discrimination if the instances from the privileged group are chosen for the positive class label randomly while the instances from the unprivileged group are chosen for the positive class label only when they are the best candidates for the positive decision [23].

Another statistical fairness measure similar to demographic parity is *conditional statistical parity*, which adds a small set of “legitimate” attributes to consider while checking the parity in the outcome [18]. In other words, the sensitive attribute(s) should be independent of the outcome given a (set of) legitimate attributes. There is another statistical fairness measure definition which emanated from a legal rule [24] and formulated by Feldman et al. [25], which is called *disparate impact*. According to this fairness measure, a given dataset has a disparate impact if the probability of getting a positive outcome given that the sensitive attribute has a value that refers to the protected group divided by the probability of getting a positive outcome given that the sensitive attribute has a value that refers to the unprotected group is smaller than the threshold of 0.8. This measure can be seen as a relaxed version of conditional statistical parity since the difference between the outcomes of protected and unprotected groups is tolerated up to 20 percent.

The second type of statistical fairness measures compares the predicted outcomes of a classifier with the actual class labels to quantify fairness. [68]. *Predictive parity* is one of these measures which is also known as the *outcome test*. This measure deems a classifier fair if both defined protected and unprotected groups have the same positive predictive value (PPV), which means that the classifier makes an equal ratio of positive predictions that actually belong to the positive class (original label being positive) for both privileged and unprivileged groups [16]. In short, to ensure predictive parity, the fraction of correctly predicted positive outcomes for both privileged and unprivileged groups should be equal.

Equalized odds is a statistical fairness measure under the same type as predictive parity,

which is also named as *disparate mistreatment* [73]. This measure deems a classifier fair if a classification algorithm yields equal true positive rates and false positive rates for both protected and unprotected groups. True positive rate is calculated by dividing the number of positive predictions which actually have positive labels by the total number of positive labeled samples, and the false positive rate is calculated by dividing the number of positive predictions which actually have negative class labels by the total number of negative labeled samples in a dataset. Equalized odds is used to investigate whether the individuals with a positive label and individuals with a negative label have similar prediction outcomes without the classifier discriminating based on the sensitive attribute membership. Another measure similar to equalized odds is *equal opportunity*, which only considers one part of the definition of equalized odds to quantify fairness. A classifier is deemed fair by equal opportunity if only it yields an equal true positive rate for both protected and unprotected groups. Thus, this measure can be considered as relaxation to equalized odds.

There are more statistical fairness measures that consider both the original class labels and the predicted outcomes from a classifier, such as the *overall accuracy equality* [7]. According to this measure, a classifier is fair if the overall accuracy of both protected and unprotected groups are equal. The accuracy is calculated by dividing the sum of true positive and true negative samples to the total number of samples in a dataset. If this measure is used as the appropriate fairness measure for the problem or case at hand, it means that true negative outcomes are as important and desirable as true positive outcomes for that case [7]. Another measure defined by Berk et al. [7] is *treatment equality*, which requires a classifier to produce equal false negative and false positive ratios (either false positives divided by false negatives or vice versa) for both protected and unprotected groups. According to the authors, the “treatment” term is used to convey that these ratios can be a policy lever to achieve different kinds of fairness depending on the domain.

The last type of statistical fairness measures is based on both predicted probability scores calculated by a classifier and actual outcomes, or original labels, of a dataset. *Test fairness*, or also named as calibration, is one of these measures that requires a classifier to produce equal prediction probabilities for both protected and unprotected groups to truly belong to the positive class to be deemed as well-calibrated [16]. Well-calibrated means that a classifier does not contain any predictive bias. It is a widely used measure as a standard for fairness assessment [16]. Another measure is called *well-calibration*, which is an expanded version of the calibration measure. Based on this measure, both the equality conditions in calibration and the predicted probability value should be equal to some value P . This measure means that if a classifier finds that a set of samples in a dataset have a certain probability value P of being assigned to the positive class, then also P percentage of these samples (people) should originally have a positive class label.

The third fairness measure in this type after test fairness and well-calibration is called *the balance for positive class*, which deems a classifier fair if samples with the positive class label from both protected and unprotected groups have an equal predicted probability score [45]. Kleinberg, Mullainathan, and Raghavan have also formalized another fairness measure called *the balance for negative class* which is the opposite version of the previous fairness measure. This time, a classifier should provide an equal predicted probability score for the samples from both protected and unprotected groups constituting the negative class in a dataset. Thus, to satisfy the “balance for negative class” measure, people with negative class labels should have the same expected probability score, no matter what their sensitive attribute value is. It is important to note that these statistical measures that require predicted probability scores can only be used with a limited number of classifiers that can calculate such scores, such as logistic regression or support vector machines.

There are several more measures defined in the literature, however, most of them have different names for the same measure formalization, as shown above with some examples. At the first

glance, statistical fairness measures seem very attractive due to their easy-to-understand nature. Nonetheless, statistical fairness measures cannot guarantee fairness for individuals or more fine-grained sub-groups of the protected demographic groups defined by the values of the sensitive attributes in a dataset. This is because of the nature of these measures: they can guarantee fairness only for the individuals who are the “average” of a protected demographic group [17]. Furthermore, there is a disagreement among different statistical fairness measures since their goals and what they consider in their criteria are different, thus they are incompatible and all of them cannot be satisfied simultaneously. This limitation is formalized and proven with the *impossibility theorem* [16, 45, 54]. According to this theorem, it is impossible to satisfy both *equalized odds* and *predictive parity* or calibration for a classifier simultaneously. Only one of them can be satisfied at a time, unless the dataset has the same base rate for both protected and unprotected groups, which means that both groups have precisely the same number of samples with the positive class label, or the classifier in question is a perfect classifier which never makes any errors. Only in these two specific and rare cases, these three fairness measures can be satisfied simultaneously. Due to these limitations, researchers have come up with new fairness measures that have different points of view. In the next sections, other proposed measures that tackle these limitations and try to solve in respective papers are further explained.

2.1.2 Individual Fairness Measures

The idea of individual fairness measures differs from the statistical measures since individual fairness measures compare the outcomes of each individual using a similarity or distance measure while statistical measures compare two or more groups in the dataset using statistical measures, explained in the previous section. One of the proposed individual fairness measures is called *fairness through unawareness*, which deems an algorithm fair if any of the protected attributes are not used in the decision-making process [46]. This measure was initially proposed as a baseline method for comparison. The main idea behind the measure is that a classification algorithm should not use any protected attributes during the training process so that its decisions/predictions should be the same for individual i and j who share the same attributes while sensitive attributes of the individuals are absent [68]. The classification algorithm which will be used is considered as a black-box. The biggest shortcoming of this measure is that the attributes that could be proxy to sensitive attributes are not considered. This means that even though the obvious sensitive attributes are removed such as race and gender, the dataset can still contain bias due to proxy attributes. For example, a street number could be a proxy indicator of the race of an individual since some people from the same background live in the same neighborhoods. This can cause discrimination in the decisions of algorithms. Besides, if a domain expert could identify the proxy attributes analogous to the sensitive attributes and remove them from the dataset, this procedure could cause an information loss and would decrease the performance of the classifier.

Dwork et al. [23] introduced the notion of *individual fairness* which means that similar individuals should be treated similarly in classification tasks to overcome the weaknesses of statistical fairness measures, specifically the statistical parity. In their paper, Dwork et al. proved that demographic (or statistical) parity alone is insufficient in terms of several aspects, namely reduced utility, self-fulfilling prophecy, and subset targeting. Even though the statistical parity measure is satisfied, when the outcomes of individuals in demographic groups are compared, they were undoubtedly unfair. To resolve this type of bias, they have proposed their method, called *fairness through awareness*, which consists of a mapping function that maps the individuals in the dataset to probability distributions over outcomes, and a utility loss function to transform the problem at hand into an optimization problem. In this way, they could set up the problem to minimize the expected loss while mapping the individuals to probability distributions under the Lipschitz constraint. They used Lipschitz continuity as

a hard constraint so that the distance between two individuals' mapped probability distances cannot be greater than the distances between two individuals in the input space. The usage of Lipschitz continuity prevents discrimination between two similar individuals, thus also preventing certain types of biases which are reverse tokenism and the self-fulfilling prophecy [23]. The authors showed that this approach also ensures demographic parity between the sensitive subsets in the dataset.

Another individual fairness measure which is introduced by Joseph et al. [36] has a completely different approach to tackle the problem. They brought the *contextual multi-armed bandit* problem to the fairness domain to ensure that any individual who has worse qualities than another individual will not be favored by the algorithm. To exemplify, one can think of the loan application case, where two individuals are applying to get a loan. According to the proposed measure, the algorithm must grant the loan to a more qualified individual no matter what the sensitive attributes are. The quality aspect of an individual is defined in compliance with the actual underlying label which the algorithm does not know. In this setting, the levers in the multi-armed bandit problem are denoted as different sensitive groups in the population, such as people having different racial backgrounds. Like each lever in the multi-armed bandit problem, each group has its unique underlying distribution function, and pulling the lever each time represents choosing an individual from a given group. The key idea behind this approach is that the algorithm cannot favor a lever that provides a lower reward when it is pulled, which means that an individual with worse qualifications will not be labeled with a positive outcome for the given classification objective. One of the limitations of this approach mentioned by the authors is that while some contextual bandit problems might be easy when there is no fairness constraint, they become computationally hard when the fairness constraint explained above is added. The second limitation is similar to the one in the study of Dwork et al.: this approach requires strong assumptions regarding the relationship between features and outcome labels and thus the underlying function describing this relationship to be implemented in real life.

Finally, Galhotra, Brun, and Meliou [31] have proposed another individual fairness measure from a causality point of view. According to this measure, for an algorithm (or software, according to the paper) to be fair, it must provide the same output for two individuals who have different values only in the sensitive attributes. To explain how the measure works, the authors gave an example of a loan application. If the algorithm grants a loan to a male applicant while it does not grant a loan to a female applicant who has the same attribute values as the male applicant except for gender, it is considered discrimination. The amount of inputs that are causally discriminated by the algorithm is the measure of causal discrimination in this approach. There is another type of discrimination defined by Galhotra, Brun, and Meliou, which is "apparent discrimination". This type of discrimination occurs when the focus of the algorithm is on an attribute that is correlated or related to the sensitive attributes, even though it does not consider the sensitive attributes in the dataset. For example, the algorithm in the aforementioned example can only look at the income level of the applicant and grant loans to people whose income level is higher. However, if the income level is positively correlated with the age group, then apparent discrimination occurs in the decisions of the algorithm [31].

This proposed fairness measure (apparent discrimination) is used to calculate a discrimination score for a given algorithm in the range of 0 and 1. If there are multiple outcomes for the decision, an "output domain distance function" is used to identify the significance of the difference between outcomes. The measure has a causal basis because it measures the fraction of individuals for which changing specific attributes of an individual cause a change in output decision regarding that individual. Hereby, it can be identified which attributes affect the decision of the algorithm directly. Galhotra, Brun, and Meliou [31] assume that there are input characteristics, which are attributes in the dataset, belonging to individuals and they are

mapped by a black-box algorithm or software to an output characteristic, which is the decision or the outcome label as a result of classification. If the algorithm provides multiple output characteristics, then they define fairness based on every single output separately. Furthermore, they assume that both input and output characteristics are categorical variables for the sake of simplicity. The limitation of this algorithm comes from the fact that the proposed measures cannot handle continuous input attributes directly, some binning procedures must be applied to this type of variable to be able to handle them.

Despite individual fairness approaches were promising improvements for the domain of fairness, they have a significant limitation due to the fundamental assumption in their solution, which is assuming that the underlying distance measure for the given dataset is known or agreed upon. Furthermore, in order to choose an appropriate distance measure, a set of assumptions should also be made regarding the relationship between features and labels in the dataset.

2.1.3 Sub-group Fairness Measures

Both statistical fairness measures and individual fairness measures have specific shortcomings that hinder fairness, aforementioned in their respective sections. Researchers have proposed new approaches to the problem, which aims to get the best ideas from both individual and group fairness notions and improve the outcomes, even though the proposed approaches are different from both former measures in practice [50].

The individual fairness notion was improved by Kim, Reingold, and Rothblum [44] by overcoming the assumption of the prior study, which is that the designer knows the actual distance function underlying the relationship between individuals. Their fairness definition is based on treating the individuals in similar sub-populations similarly, which is in-between individual and group fairness notions. They introduced the *metric multifairness*, which is quantified by a similarity measure based on pairs of individuals. It helps to classify these individuals in sub-populations to treat the ones in the same sub-population similarly. They eliminated the need to precisely know the underlying measure by a relaxation technique similar to the statistical fairness approaches, which asks the individuals to be classified in pre-defined sub-groups with probabilities proportional to the mean distance between the individuals in those sub-groups. In this study, there is an oracle which the learning algorithm can access and that estimates the distance between randomly chosen two individuals based on an unknown fairness measure. The learner algorithm can query the unknown metric a limited number of times. The oracle does not need to make any structural assumptions regarding the underlying distance metric, so it works even when the metric cannot be learned. However, the authors also stated that efficiently achieving high-utility metric multifairness predictions and satisfying their constraints require some learnability assumptions on the collection of comparisons which contains the associated distances between pairs of individuals.

Kearns et al. [42] introduced a new notion called *fairness gerrymandering* to point at one of the problems in group-based (or also named as statistical) fairness measures, which refers to the situations when a classifier seems fair on each group existing in a sensitive attribute, for example, male and female values in gender, but it fails at being fair for one or more subgroups in the dataset defined over multiple sensitive attributes. The subgroups can be the combinations of different sensitive attribute values, such as the combination of race and gender (i.e. black female and white male). To overcome the problem, they define the subgroup fairness measure where they choose a statistical constraint, such as false-positive rates, and then they ask a class of functions that the chosen statistical constraint holds over a very large collection of subgroups, which are defined by this class of functions. This subgroup fairness approach can be considered in-between individual and group fairness because it considers much more fine-grained and smaller subgroups in the population than statistical measures.

Although this approach is very promising since it does not need specific assumptions regarding the data just like the group-based or statistical measures, one shortcoming of this approach is that it is not certain which function classes are feasible or reasonable to use for each dataset at hand, and there is no clear guidance about which attributes should be included as protected attributes to define the subgroups later [17].

2.1.4 Causal Reasoning Measures

The fourth category of fairness measures is based on investigating the causal relationships between the attributes and the outcome labels. This class of measures was proposed by researchers as a result of the limitations and shortcomings of the measures in previous categories due to their nature and the assumptions made to construct them. Causal reasoning measures are not completely data-driven, it requires an additional understanding and knowledge of how the world is structured in the form of a causal model per domain [49]. The required knowledge is substantial since it provides information about how a change in an attribute can cause a change in the system.

Understanding the causal relationship between the sensitive attributes and other attributes, as well as the relationship between attributes and the decision (or outcome) attribute is important since it will enable the designers to identify and recover from certain types of bias in the dataset, such as sampling bias. There are several types of attributes in this approach introduced by different papers [43], [46]:

- A proxy attribute is an attribute whose value can be used to derive the value of another attribute,
- A resolving attribute is an attribute that is affected by the sensitive attribute but not in a discriminatory sense so that it does not convey any discriminatory effect, and
- Latent attributes which are attributes that are not caused by any other observable attributes in the dataset.

To measure the fairness in a dataset in terms of causal relationships, Kusner et al. [46] proposed a fairness measure called *counterfactual fairness*. It is the first research paper that has explicitly introduced causal reasoning and the usage of directed acyclic graphs (DAG) in the fairness domain of machine learning. According to this measure, a dataset is counterfactually fair if a decision regarding an individual is identical in the actual world as well as a counterfactual world where that individual belongs to a different demographic sub-population. It means that distribution over possible predictions for an individual should not change if the same individual would have different values for the sensitive attributes. In general terms, a causal graph of a dataset is fair according to this measure if the predicted outcome in the graph does not change depending on a descendant of the protected attribute. The proxy attributes can be the descendants of a sensitive attribute. If the outcome attribute is directly dependent on a proxy attribute, then the graph is counterfactually unfair. The authors stated that counterfactual fairness is also an individual-level measure since it checks the counterfactual worlds for each individual in the dataset. One remarkable shortcoming of this measure is that a domain expert might be needed to construct an accurate causal graph with correct identification of attributes and relationships for a given dataset. The misconstruction of a causal graph for any dataset might hinder the discovery of actual underlying discrimination and cause misinterpretation.

Another study after the proposal of counterfactual fairness was conducted by Kilbertus et al. [43], using causal reasoning in their paper as a base concept. They show that the statistical measures fail to determine if a protected attribute has a direct causal influence on the predictor that is not mitigated by resolving variables. They also emphasize that it is important to distinguish the sensitive attributes from their related proxy attributes so that the underlying

effects of sensitive attributes on the decision attribute can be revealed. This work in general proposes a new flexible framework to fairness, which is making arguments about the attributes of a dataset and finding a justified causal data generating process for it instead of trying to find a single statistical fairness measure to apply. The authors also mentioned that the previous work in [46] requires creating counterfactuals per individual in the dataset, thus it is a delicate task. For this reason, they have proposed a more general framework without needing to evaluate the individual-level causal effects so that the complexity of the task would not be as high as the previous work.

In the relaxed framework, a causal model or graph is constructed when a predictor maps the input features to the predicted outcome or decision [46]. Both input and outcome attributes are drawn as nodes in the graph, and the relationships among them are denoted with arrows. The outcome attribute does not have any child nodes, and its parent nodes are the input attributes. Based on this idea, Kusner et al. defined a new measure called *unresolved discrimination*. This discrimination occurs in the causal graph of a dataset if there is a directed path from a sensitive attribute to the decision attribute Y that is not blocked by any resolving attribute [43]. Another measure defined to measure bias is called *potential proxy discrimination*, which occurs when the directed path from a sensitive attribute to the outcome attribute is blocked by a proxy variable in a causal graph [46]. With these causal measure definitions, one can find different effects of sensitive attributes on the outcome attribute without diving into individual-level comparisons. However, the proposed measures in [46] are still prone to the same limitation of the previous work: it is assumed that they can identify sensitive, proxy, and resolving attributes correctly and able to construct an appropriate causal graph for a given dataset. There is a need for a domain expert to successfully construct such graphs for a dataset, which is not always reachable easily in real life.

2.2 Mitigation Algorithms

Researchers have been not only working on finding the best measure to quantify bias but also working on finding an appropriate technique to eliminate the bias identified in a dataset or a model. Thus, there are several proposed bias mitigation techniques to eliminate or mitigate unfairness considering the accuracy performance.

2.2.1 Pre-processing Algorithms

The first category of bias mitigation algorithms is *pre-processing* algorithms, or techniques, where the dataset is altered before the training of a classifier in order to obtain a fair dataset as an input. There are several ways to pre-process a dataset before training classifiers. The most trivial approach in this group of techniques is called *fairness through unawareness*, which considers a predictor model fair if it does not use any of the protected attributes in the prediction process [29]. In order to achieve fairness according to this point of view, it is enough to remove all the sensitive attributes from the dataset before training the model. However, it is known that deleting only sensitive attributes is an inefficient technique to achieve fairness especially when other attributes allow the removed sensitive attributes to be easily deduced in the dataset [33]. Thus, it is not a preferred pre-processing approach to eliminate discrimination in a dataset.

One of the sophisticated approaches to pre-process a dataset is re-sampling the data instances. Kamiran and Calders [37] proposed the “preferential sampling” approach, where they sample the data objects with replacement in order to eliminate bias. The core idea behind this approach is choosing the data objects that are the best possible choice to eliminate the discrimination in the dataset. In this technique, the data objects from the protected group which are close to the borderline are considered more vulnerable to have been discriminated

against, while the data objects from the unprotected group are highly likely to be favored due to unfairness in the dataset. Thus, these objects are preferred for sampling. To identify the borderline objects, the algorithm learns a ranker on the given training set and then orders the objects. Then, the objects from the protected group with the positive outcomes which are near the decision borderline are duplicated, while the objects from the protected group with a negative outcome near the decision borderline are removed. This procedure is done for the unprotected (privileged) group opposite to the protected group, where the ones near the decision borderline with positive outcomes are removed and the ones with negative outcomes are duplicated.

Another re-sampling technique is recently proposed by Salimi et al. [61] with the causal model approach called “interventional fairness”. They consider the problem as a database repair problem. In this technique, the training data is “repaired” by inserting or removing tuples and alter the probability distribution in the dataset to remove any causal relationship between the sensitive attributes and the decision variable. Salimi et al. stated that their approach does not require the knowledge of the underlying causal model because it is based on “intervention”, which can be guaranteed even when the actual causal model is unknown. The only input required for this approach is labels of the attributes in the dataset as “admissible” or “inadmissible”, where admissible attributes are the attributes that are allowed to influence the outcome even though they have a causal relationship with the sensitive attribute.

Massaging is another pre-processing technique that refers to changing the actual class labels of some of the instances in the training set to ensure fairness [38]. A ranker algorithm is used to choose the appropriate instances to relabel. Using the massaging algorithm, Kamiran and Calders [38] have changed the labels of several instances that belong to the protected group from negative to positive, while they have changed the same number of other instances which belong to the unprotected group from positive to negative. In order to achieve this operation, they have learned a ranker algorithm R , which orders the instances in the training set in descending order according to their positive class probability. After this ranker is learned, the instances which are good candidates for promotion are sorted in descending order, while the instances which are good candidates for demotion are sorted in ascending order, then the top instances in these lists are chosen to change their labels. By choosing the top candidates, the ones closest to the decision border will be relabeled. This procedure is repeated until the discrimination in the training set is eliminated. The authors aimed to have minimal accuracy loss by choosing the instances closest to the decision border in this technique. They state that this technique decreases the amount of discrimination while the overall distribution of the class labels remains the same.

Massaging is not always applied on the class labels of instances, it is also applied on the attributes (variables) except the sensitive attribute(s) of a dataset, which is a technique proposed by Feldman et al. [25]. The authors have introduced a notion called ϵ -fairness, which is based on predicting the sensitive attribute values of individuals based on the other attributes in a dataset. According to this idea, a classifier f should fail predicting the sensitive attributes values of individuals using the other attributes. It means that a dataset is ϵ -fair when the balanced error rate (BER) of predicting the protected and unprotected groups, which is the unweighted average class-conditioned error rate of f , is greater than ϵ based on the empirical probabilities. In other words, when the sensitive attribute of given dataset is not predictable by the model based on other attributes, the dataset is free from disparate impact, thus it is fair. A disparate impact in a dataset is identified, or *certified* according to the authors, by using a hinge-loss SVM and optimizing the BER for a given dataset, then trying to predict the protected attribute value of instances based on other remaining attributes. After disparate impact is identified, the changes on the values of attributes are done in such a way that it will not be possible to predict the sensitive attribute for a classifier while it can predict the class label of instances. To eliminate disparate impact, the relative

per-attribute ordering of instances based on the cumulative distribution functions of the dataset per attribute is preserved while changing the values of attributes (except the sensitive attributes). As a result, a classifier trained on this repaired dataset chooses better instances that are ranked higher in the ranking over the lower-ranked ones. The authors also talked about the fairness-accuracy trade-off in their study. They introduce partial repair notions to handle this trade-off at the desired level.

The third pre-processing technique is *reweighing*, which is also a pre-processing technique that is applied to the input dataset before training the model. Calders, Kamiran, and Pechenizkiy [12] have proposed reweighing in order not to change the labels of the instances, and not to add or remove an instance from the dataset, which means minimum interference with the dataset while eliminating unfairness. Authors have developed reweighing since massaging technique intervenes datasets in an undesirable way which is a disadvantage of this technique. Assigning weights to each instance in the training set needs the weights to be carefully chosen to provide balance with respect to the sensitive attributes. This approach will assign higher weights to the instances from the protected group with positive outcomes than the instances from the protected group with the negative outcomes to lower weights. Vice versa, the instances from the unprotected group with the positive outcomes will be assigned to lower weights than the instances from the same group with the negative outcomes.

Zemel et al. [75] have proposed another technique that has a different approach from the others, which is called *learning fair representations* (LFR). The authors defined the problem as an optimization problem, and they find an appropriate intermediate representation of a given dataset that encodes the data as accurately as possible while concealing any information about the sensitive attributes of individuals at the same time with a learning algorithm. The learned mapping by an algorithm ensures that the information regarding protected group membership is lost. The goals of LFR are to create prototype sets (with mapping) that can satisfy statistical parity in which these prototypes keep the information about attributes except the sensitive attribute membership regarding instances, and the final mapping of these prototypes to the class labels is close to the original classification function. The authors also stated that their learning algorithm achieves both group and individual fairness simultaneously, which is a very good advantage over other bias mitigation algorithms if one would like to achieve fairness.

A close study to Zemel et al.'s LFR [75] is conducted by Calmon et al. [14], which also considers unfairness as an optimization problem with a probabilistic framework. However, test data are also transformed probabilistically before they are given to a classifier model as well as the training data in this pre-processing technique. The authors formulated a convex optimization problem as the trade-off between discrimination control, data utility, and individual distortion to mitigate the bias while minimizing the changes in the original dataset. The distortion control is used to restrict the mapping to minimize or completely avoid certain large changes in the values of the individuals in a dataset, such as restricting a very low credit score to be mapped into a very high credit score during transformation of the dataset. Utility preservation in a dataset refers to having small differences between the original dataset (X, Y) and the transformed dataset (to ensure fairness) with classifier predictions as labels (\hat{X}, \hat{Y}) where X is the set of attributes and Y is the class label. Lastly, the purpose of the discrimination control is to limit the dependence of the transformed class label \hat{Y} on the sensitive attributes by using the disparate impact ratio (80%-rule). In [14], Calmon et al. improved the previous study [75] by providing more solid structure, detailed formulations, and the required conditions to obtain a convex optimization problem in order to successfully optimize these three constraints simultaneously in a dataset.

Finally, the unfairness problem is also approached with a different point of view, where it is thought that the source of bias is the class imbalance in a dataset, which means that the number of instances from the protected group, or groups, are significantly less than the number

of instances from the privileged group. In other words, because the protected groups in general are under-represented in a dataset, it causes models to learn discrimination while being trained. In order to eliminate the bias in imbalanced datasets, Yan, Kao, and Ferrara [72] have proposed a fair data oversampling technique called *fair class balancing*, which does not use any information regarding the sensitive attributes. The authors stated that standard balancing techniques such as the synthetic minority over-sampling technique (SMOTE) can increase the bias after balancing since these techniques do not consider the sensitive attributes. To solve this problem, the authors have used a “cluster-based” class balancing technique, where the *group structure* of the dataset which consists of natural subgroups with similar features is identified and clustered by a clustering algorithm of choice. After finding the best number of clusters for the dataset, the instances which are near the cluster borders are removed. Then, the minority class for each cluster is identified and new instances are generated based on k-nearest neighbors of the instances belonging to the minority class. One advantage of this algorithm is that when there is a legal prohibition to use any sensitive attributes such as race and gender, it works without needing the sensitive attribute information. However, one limitation of this work is that it only considers the class labels while oversampling instead of considering both the sensitive attributes and the class labels.

2.2.2 In-Processing Algorithms

In-processing algorithms are the algorithms that tune or adjust a classification algorithm in order to make the model yield fair predictions. There are several classifiers that are altered for in-processing such as support vector machines (SVM), logistic regression, and random forest algorithms, which are discussed in section 2.4. In processing algorithms are mostly dependent on the classifiers that are implemented upon, or the type of classifiers that could be implemented, such as the probabilistic classifiers.

One of the earliest studies in in-processing is conducted by Kamiran, Calders, and Pechenizkiy [39] using decision trees as a classifier to adjust, or *constraint*, to ensure fairness. The authors transformed the decision tree algorithm into a *discrimination aware* classifier with the goal that when a potentially biased historical data given as an input to this classifier, it outputs accurate and fair predictions at the same time. To achieve this, they propose two techniques for the decision tree construction process in training. While determining the best split on attributes, they enforce the algorithm to consider both the information gain score for the quality of that split and another score called *discrimination gain*, derived from information gain, which represents the influence of the new split on the resulting tree in terms of discrimination. They sum these two scores to construct a tree that is homogeneous in terms of both accuracy and the binary sensitive attribute. Instead of assigning the label of the majority in a given leaf, they relabel a set of leaves optimally by keeping the loss in the accuracy minimal while reducing the discrimination. The authors stated that they approximate the most optimal relabeling for a given tree. Even though the proposed technique sounds promising, it is only limited to the decision trees, which is not a popular algorithm for classification anymore with the arrival of more improved techniques, such as random forests.

Zafar et al. [74] have also implemented an in-processing algorithm based on constraining classifiers, which is formulated as a regularized optimization problem, using logistic regression and SVM algorithms. Specifically, they have implemented a decision boundary measure as a proxy to their chosen fairness constraint (the 80%-rule) for the convex margin-based classifiers that prevents both disparate treatment and disparate impact. The authors have derived two functions that complement each other in order to train the classifiers fairly. The first function maximizes the accuracy while satisfying the fairness constraints, and the second one maximizes fairness while satisfying the accuracy constraints. They achieve these by defining the *decision boundary fairness*, which is the covariance between the sensitive attributes of individuals in the training set and the signed distance between feature vectors of these individuals and the

decision boundary. To maximize the accuracy under fairness constraint, the parameters of decision boundary which minimize the loss function and satisfy the covariance threshold are found. This threshold is an upper bound value which can be tweaked to obtain the best loss in accuracy-fairness trade-off. In order to maximize the fairness under accuracy constraint, the calculation is done the other way around. The parameters of decision boundary are found, which minimize the decision boundary covariance under the constraint of the loss function. With this function, the maximum fairness that can be achieved without having loss in the accuracy. In this procedure, the sensitive attributes are not used while making decisions about the class label of individuals at the prediction time, thus the solution does not contain any disparate treatment. It is promising that the *decision boundary fairness* technique can handle both multiple sensitive attributes simultaneously and multi-valued attributes (more than two outcomes), however, it is still limited to the algorithms that has a decision boundary to decide the class label of the individuals.

Kamishima, Akaho, and Sakuma [41] have proposed another in-processing approach with *regularization* that can be applied on any probabilistic classifier to mitigate bias. The proposed technique is called *regularized prejudice remover*, which enforces classifiers to make the predictions independent from a sensitive attribute. The authors first identified the causes of unfairness, namely prejudice, underestimation, and negative legacy, then they have defined three types of prejudice that can be found. These are direct prejudice, indirect prejudice, and latent prejudice [41]. The authors focused on eliminating the indirect prejudice by developing a technique that regularizes and restricts the behavior of classifiers since direct prejudice can be easily avoided by removing the sensitive attribute and latent prejudice is hard to identify. They also propose two fairness measures to quantify the degree of different types of discrimination. These measures are *indirect prejudice index* which is defined by using mutual information between the predicted class label and a sensitive attribute for prejudice and *underestimation index* which is defined based on Hellinger distance for underestimation. There are two regularizers implemented in a probabilistic classifier: a standard L2 regularizer to prevent overfitting, and a regularizer to enforce fair class prediction by reducing the indirect prejudice index. The regularizer in a classifier becomes greater when the predictions of a classifier are highly based on a sensitive attribute so that it can decrease the influence of this sensitive attribute on the prediction of class labels. This regularization technique is applied to the logistic regression algorithm, and can only be implemented with classifiers that can calculate probability distributions based on a given dataset.

Adversarial learning is one of the techniques that has been used as an in-processing technique to ensure fairness. Zhang, Lemoine, and Mitchell [76] have proposed a framework with adversarial debiasing to mitigate bias, which can be implemented with gradient-based models for both classification and regression tasks. This framework is built on including a sensitive variable, a learning predictor together with a competing adversary simultaneously, which aims to maximize the predictor’s capability of predicting the class label while minimizing the ability of the “adversary” to predict the sensitive attribute. The amount of fairness can be measured with demographic parity, equalized odds and equal opportunity while using this framework. It is a technique where a predictor model is trained by adjusting its parameter weights to minimize the loss using a gradient based method. The output layer of this predictor is given as an input to the adversary, which tries to predict the sensitive attributes based on the input. The goal is to train the predictor algorithm to predict the class labels of individuals as accurately as possible. However, the predictor must satisfy both demographic parity, equalized odds and equal opportunity to ensure fairness. Thus, there will also be an adversary introduced, which will attempt to predict the sensitive attribute from the predicted outcome given by the predictor to first achieve demographic parity. Then, the gradient of the adversary will be used in the weight update of the predictor model so that the amount of information regarding the sensitive attribute which is transmitted through the predicted outcome will be reduced. In order to ensure equalized odds in this technique, the adversary

will have access to the true class label so that it can limit any information about the sensitive attribute that the predicted outcome contains more than the information already contained in the original class label. If the predictions of the predictor contains more information about the sensitive attribute than the original class labels, the adversary will improve its loss. As a result, the predictor that tries to fool the adversary will move its parameters toward where its predictions will not leak information about the sensitive attribute. It is another promising technique which can handle both categorical and numerical attributes, however it is limited to the gradient based methods, and it requires a specific tweaking in order to avoid converging the local minima or not to satisfy only one of the fairness measures satisfactorily while optimizing the parameters.

Finally, in-processing algorithms have been used to mitigate the discrimination emerged due to imbalanced datasets. For example, to eliminate the bias in SVM classifier trained with imbalanced datasets, Ristanoski, Liu, and Bailey [56] have proposed an empirical loss-based tuning on SVM which also considers the imbalance in the number of samples with positive and negative class labels. In their paper, the authors first demonstrated how having a very low percentage of positive class samples affect the discrimination. Especially if the unprivileged group has a low percentage of positive samples, it causes the classifier to predict even less positive outcomes and thus there will be a greater discrimination score. However, if the privileged group also has a low percentage of positive samples, then the classifier will have less positive predictions for this group too, which will cause a lower discrimination score. The authors defined discrimination in an equation, where the total discrimination equals to the sum of explanatory discrimination in the dataset and additional discrimination added by the trained classifier. They choose to optimize the additional discrimination introduced by the classifier itself since it can be interpreted as the difference between the predicted discrimination score derived from the predictions of a classifier and actual discrimination score derived from the imbalanced dataset itself, which is the *discrimination aware empirical loss* defined by the authors. Then, this defined loss is implemented with SVM algorithm and the imbalanced dataset is given as an input for training. This algorithm does not focus on eliminating the existing bias in the dataset, instead, it only works to prevent the discrimination exacerbation of the classifier during the training. Thus, its scope is very limited and it cannot potentially eliminate the discrimination completely.

2.2.3 Post-Processing Algorithms

The last category of mitigation algorithms is the post-processing algorithms, which change the predicted outcomes of classifiers based on certain rules or constraints. Thus, the goal here is to eliminate the discrimination from the final predictions instead of the input dataset or within the models. One of the earlier proposed algorithms was implemented by Kamiran, Karim, and Zhang [40] to change the predicted class labels of the instances that are close to the decision boundary, which is called *reject option classification* (ROC). In [40], the authors provided two possible solutions to reduce discrimination in predictions, which are probabilistic rejection on both single and multiple classifiers (ROC), and the disagreement region of classifier ensembles, which is called *discrimination-aware ensemble* (DAE). Their solution is based on the idea that the instances near the decision boundary are mostly discriminated in decision making. In the ROC technique, the posterior probabilities between 0.5 and a threshold value θ are considered as *critical region* and the instances having posterior probabilities in this region are labeled as “reject”, which are considered to have a biased outcome.

For multiple classifiers, the labeling is done in ROC by averaging all posterior probabilities calculated by each classifier for a given instance and then labeled as “reject” if it falls in the critical region. However, when a classifier does not provide any probability estimates, then DAE can be used. In this technique, the main idea is that if member classifiers have more disagreement in their predictions for the class label of an instance, then that instance

is close to the decision boundary and thus it might be discriminated more. When there is a disagreement among the classifiers of the ensemble, instead of standard majority voting strategy, the authors compensated the protected group members by assigning the desired class label, which is 1 or *positive* label in most of the cases, and then penalize the privileged group members by assigning the undesired class label. If all the ensembles agree with the predicted class label, then they are left untouched. The ROC algorithm that Kamiran, Karim, and Zhang have proposed can be considered as a *thresholding* technique since it considers a certain threshold and a critical region to modify the predicted outcomes of classifiers.

Another post-processing algorithm with thresholding approach is proposed by Hardt, Price, and Srebro [33]. The authors used the equalized odds as their core fairness measure and adjust the predictions of a classifier obtained at the end of the training step based on it. Their main goal is to create some new predicted outcomes that will minimize the expected loss defined by them. Since satisfying the equalized odds in both true positive and false-positive rates and also the expected loss can be formalized as a linear function, the optimal derived predictor can be obtained as a solution to a linear program with these two functions and four variables, which come from the combination of binary values that the class label and the sensitive attribute can have. In [33], they mention how thresholding can be applied to the classifiers providing predictive scores next to the classifiers that only provide binary predictions. These threshold values are optimized to maximize accuracy while ensuring fairness based on equalized odds.

Hardt, Price, and Srebro [33] state that the notions of non-fairness that they define are non-oblivious since they only need to know the joint distribution of the sensitive attribute, original class label, and the predicted label or predicted score depending on the classifier. They do not need to know anything about other attributes in the dataset. In order to eliminate the discrimination, they construct a *non-discriminating predictor*, which is the adjusted version of the predicted labels, based on the equalized odds values which are derived from the predicted label or predicted score and the sensitive attribute. The authors also noted that at the testing or prediction time, they only need to access the values of the predicted score and the sensitive attribute, which is an advantage of this technique over other post-processing techniques. However, to optimize such a system, the user has to provide a ratio to the loss function in order to define either false positives or false negatives are more costly and how severe the cost is in this system.

Pleiss et al. [54] have shown a different point of view in fairness and post-processing in their study. They illustrated with their post-processing algorithm that calibration and error rate constraints are incompatible goals and ensuring both constraints by relaxation does not provide satisfactory results. Even though it does not completely prevent bias, *calibration* is one of the main goals in risk assessment tools with different settings that provide a group of people with a predicted probability P to have P fraction of people with a positive class label. The error rate is calculated by using equalized odds measure. The authors also talked about the impossibility theorem, where equalized odds and calibration cannot be satisfied simultaneously unless very specific conditions. Thus, the authors have implemented a post-processing algorithm that relaxes the equalized odds constraints to become compatible with calibration constraints. To relax the constraints in the equalized odds, the authors have defined a cost function, which is a linear function concerning false positive rate (FPR) and false-negative rate (FNR) and depends on the base rates of the groups defined by the sensitive attribute. Relaxed equalized odds with calibration is satisfied by a classifier for both protected and unprotected groups if it is calibrated and satisfies the constraints of the cost function which is a linear function with arbitrary dependence on the base rates of demographic groups in a dataset. The authors have noted that if the cost increases then FPR and FNR also directly increases, and the equalizing cost penalty is magnified when the group base rates considerably differ. They have also noted that the proposed algorithm is unsatisfactory since a significant amount of the individual predictions are withheld which is an undesirable setting,

especially when working on sensitive domains such as healthcare.

Finally, Kilbertus et al. [43] have proposed two post-processing algorithms, namely avoiding proxy discrimination and avoiding unresolved discrimination, to eliminate unfairness on the predictions of a classifier based on causal perspective and two causal measures they have introduced. These causal measures are proxy discrimination and unresolved discrimination, which are discussed in the causal reasoning measures section. The authors have used the Bayes optimal classifiers, which seek to find the most probable underlying model hypothesis for a given dataset. Then, they used these classifiers to predict the most probable class label for a test instance based on the training set. The proxy discrimination is avoided by intervening in the proxy attribute: removing its connections with its parents, substituting the attributes for the predictor function from the hypothesis class iteratively in a way that the distributions of these hypothesis functions become independent of the values of the proxy attribute. As the last step, this predictor function is optimized based on the non-discrimination constraints. To avoid unresolved discrimination, the resolving attribute is intervened: the resolving attribute is fixed to a random attribute whose marginal distribution is equal to the marginal distribution of the resolving attribute and it is ensured that the distribution of the predictor function to be invariant to the values that the sensitive attribute gets. However, it is important to note that there is an assumption similar to the general causal measures, which is assuming that the valid causal graph given by a dataset can be constructed.

Unfortunately, most post-processing algorithms have a set of common limitations in practice. For example, the post-processing algorithms which make corrections on the classifier predictions by randomizing them cannot be used in specific domains due to ethical reasons, such as in healthcare to diagnose fatal diseases. Furthermore, it is also demonstrated by Woodworth et al. [71] that post-processing algorithms might deliver sub-optimal performance in terms of accuracy compared to the other fairness techniques. Thus, post-processing algorithms are not the best option for practitioners who would like to achieve fairness while obtaining as high accuracy as possible.

The study in this thesis differs from most of the studies in the literature since any of the information in a dataset is not changed or transformed, so the dataset at hand is kept with its original values. Also, neither classifier training procedure nor the predicted outcomes are not intervened. The closest study to ours is [72], however, our study also differs from [72] since we consider the combination of both multiple sensitive attributes and the class labels in datasets, and we do not remove any original samples from the dataset while oversampling.

2.3 The Goals in Fairness-Aware Machine Learning

There are two main goals in the domain of fairness in machine learning. The first goal is to detect and quantify the amount of discrimination in a dataset or a system, which can be also considered as bias diagnosis. The second goal is to eliminate the existing discrimination, or at least mitigate it under a certain threshold, after identifying the amount of discrimination. In order to facilitate the processes, researchers have come up with different tools for these two different goals. The tools that are currently available in the literature are explained in their respective sections.

2.3.1 Discrimination Detection

One of the earliest tools was developed by Adebayo et al. [1] for bias diagnosing and auditing the predictive models, which is called *FairML*. It is an open-source end-to-end toolbox that approaches models as they are a black box and measures the significance of the attributes in a dataset. The FairML processes the initial dataset through orthogonal transformation and then feeds it to the black-box model or algorithm. Then, it compares the prediction results

obtained from the initial dataset and orthogonally transformed dataset by using a distance measure. By looking at the significance level of the attributes in prediction, one can identify if the sensitive attributes are affecting the prediction outcomes while using a machine learning model. This tool also contains a graphing module to visualize the significance ranking of the attributes with the amount of significance. It is relatively a simple tool to investigate how heavily the sensitive attributes and also proxy attributes (if they are known) have an effect on the predictions of the algorithms.

Another tool by Galhotra, Brun, and Meliou [30] is called *Themis*, which is also an open-source tool that is created to provide automatically generated test suites in order to quantify the discrimination in the decisions of a predictive system. It is used to detect discrimination on a software system instead of specifically the machine learning algorithms. The inputs of this tool are an executable software, the desired confidence level, and an error bound together with an input schema that describes the valid system inputs. According to these inputs, Themis generates discrimination tests automatically. It has two types of fairness measures implemented, which are group and causal fairness measures. Themis can capture the causal relationships between inputs and outputs of a system. Checking both types of discrimination in the system is a computationally complex activity, thus the authors implemented optimization strategies to provide output in a reasonable duration. This tool can be seen as a good example of a system or a pipeline auditing in terms of fairness.

Tramer et al. [67] implement another open-source discrimination detection tool called *FairTest* for auditing based on *unwarranted associations* between the groups defined by the sensitive attribute(s) and the decision outputs of an algorithm. Unwarranted associations framework consists of multiple fairness measures to identify disparities and unfair treatments in subgroups. The authors defined the unwarranted associations as any statistically significant association between a protected attribute and decision outcomes in user subgroups where the association does not have any explanatory factor. Fairtest learns a special decision tree to divide the population in the dataset into subgroups for further investigation. It contains several statistical fairness measures for canonical association measurement such as binary ratios and their difference between subgroups.

Another open-source fairness auditing toolkit that is recently implemented is called *Aequitas*, developed by Saleiro et al. [60]. It has three versions: a Python library and a command-line tool for offline auditing, and a website audit tool where the users can upload their dataset on the website and choose the desired measures for analysis¹. It has only statistical fairness measures implemented such as demographic parity, disparate impact, and false discovery rate. Aequitas also contains a *fairness tree*, which helps users to find the correct fairness measure to use for their particular dataset and situation. It outputs reports regarding the predictive and fairness performance of the dataset based on the chosen measures over the population subgroups together with the explanations about those measures. Even though the implemented fairness measures are limited on the website tool compared to the python library, its website is very compact and easy to use for auditors without a technical background.

Fat Forensics is one of the recently implemented open-source toolkits developed by Sokol et al. [64]. The term FAT refers to fairness, accountability, and transparency. It is also a Python toolbox that is built on SciPy and NumPy libraries with a public license. It inspects the chosen fairness, accountability, and transparency measures or aspects of these systems to automatically report them back to its users. It provides functionality for inspecting both datasets, predictive models, and outcome predictions. This tool contains statistical fairness measures such as demographic parity, equal opportunity, equal accuracy, and a causal fairness measure, which is counterfactual fairness.

Finally, the most recent tool for bias detection is implemented by Google, which is called

¹<http://aequitas.dssg.io/>

What-If Tool [69]. The name comes from a functionality of the tool which is investigating the “what-if” hypotheses about the models by identifying counterfactuals. It provides this functionality by allowing to easily edit the points in a dataset. It provides six statistical fairness measures, some of which are demographic parity and equal opportunity and general performance measures such as the ROC curve over subsets (groups defined by the sensitive attribute) of a dataset. This tool contains detailed visualization options for performance and fairness results, both in text, tables, graphs, and plots. This tool is a plugin, which can be used via Jupyter Notebook or Google Colab. The significant limitation of this tool is that it is only compatible with TensorFlow models, which means that if one wants to use this plugin, he/she needs to transform their model or dataset into TensorFlow models.

Detecting the discrimination in the dataset is only half of the story for the businesses and organizations that want to achieve fair outcomes from their systems or algorithms. Unless the purpose is solely auditing as a third party, one will want to eliminate the identified discrimination at hand using a tool. The discrimination detection tools alone will not be enough for these parties. Thus, discrimination elimination tools are more comprehensive and appropriate to both identify and try to eliminate discrimination.

2.3.2 Discrimination Elimination

One of the open-source discrimination elimination (or bias mitigation) tools is implemented by Bantilan [4], which is called *Themis-ml*. It is proposed as Fairness-aware Machine Learning Interface (FMLI), which is an application programming interface (API) implemented for binary classifiers. It means that the tool can only handle two outcomes for the decision variable. The tool is implemented with Python programming language and the Scikit-learn library. The tool contains both individual and a few statistical fairness measures such as mean difference, together with several pre-processing, in-processing, and post-processing mitigation algorithms, such as reweighting, prejudice remover regularizer, reject-option classification, and additive counterfactually fair model, which were explained in the previous section. It also has readily available datasets such as German Credit, Taiwan Credit Default, Australian Credit Approval, Disabled Residents Expenditure, and Census Income. Even though it has several mitigation algorithms in this tool, the number of fairness measures offered is limited.

Another discrimination elimination tool has emerged from a benchmark study as a library, conducted by Friedler et al. [27] with the name *Fairness-Comparison*². This library can be used with both R and Python. It contains several fairness measures such as equal opportunity difference, true negative rate, true positive rate, and calibration. It contains the five most famous datasets in the fairness domain, which are Adult Income, Ricci, German credit, and ProPublica Recidivism (normal and violent) datasets. It also has several mitigation algorithms such as disparate impact remover [25], prejudice remover [41], and convex relaxation of fairness constraints [73]. In total, for bias elimination/mitigation, it has one pre-processing algorithm (disparate impact remover [25]) and three in-processing algorithms (decision boundary covariance [74], regularization on logistic regression [41], and three naive bayes [13]) in its library. It also contains both accuracy and statistical fairness measures.

Microsoft has also developed an open-source toolkit to mitigate the algorithmic fairness, which is called *Fairlearn* [9]. This tool implements parity-based (statistical) fairness measures such as demographic parity difference, equalized odds difference, and false-negative rate, and mitigation algorithms for binary classification. However, it only has one post-processing algorithm (threshold optimizer [33]) and some reduction algorithms for disparity mitigation such as error rate parity, grid search [2], and exponential gradient [2]. It has only Adult Income, Bank Marketing, and Boston Housing datasets in its library. Fairlearn toolkit contains two core components, which are the bias mitigation algorithms and the interactive dashboard that can

²<https://github.com/algofairness/fairness-comparison>

be invoked on Jupyter Notebook for graphical display of the visualizations. However, it is also a limited toolkit in terms of both the number of provided fairness measures and mitigation algorithms.

Finally, the most extensive and open-source bias mitigation tool is developed by IBM, with the name *AI Fairness 360* (AIF360) [6]. The purpose of this tool is explained as a transition of the proposed bias mitigation algorithms in the literature to the industrial setting and providing a common framework for the researchers to benchmark, evaluate, and share the algorithms. It can also be used with Python or R. The unique property of this tool is that it contains an explainer class that automatically generates explanations for both the measures and the algorithms to mitigate the bias in the models or datasets. AIF 360 has the largest number of algorithms among these tools with fourteen algorithms from both pre-processing, in-processing and post-processing approaches. It also has an interactive web experience³ to provide an introduction to the tool like Fairlearn’s interactive dashboard. Currently, the service is limited to five built-in datasets which are famous in the fairness domain, but the user can also upload his/her own dataset by expanding the dataset class. The service is also limited to building logistic regression classifiers, but again this can be expanded. The tool also has a separate scikit-learn interface in order to make it easier for the users to integrate this tool in their machine learning pipeline even though the capabilities of this interface are more limited compared to the whole toolkit of AIF 360. The only limitation regarding AIF 360 is its own object-oriented structure. While it is very beneficial in terms of scalability while developers are expanding the tool, users will have to deal with more technical aspects since they have to write their own wrapper classes whenever they would like to upload a new dataset that does not exist in the dataset class.

2.4 Classifiers Used in Algorithmic Fairness

Classification algorithms are trained to make categorical predictions about the class labels of the samples. The classifiers can handle the decision attributes with either binary (with two outcomes only) or multiple class labels (more than two possible outcomes). There are different kinds of classifiers in the literature: the ones that only provide the predicted class labels, and the ones that can provide both predicted class labels and the predicted probabilities for every sample in a test set. Since there are several fairness measures that use both the predicted probability scores and the original class labels, the researchers mostly prefer using the classifiers that are capable of providing predicted probabilities in addition to the predicted class labels so they can use whichever technique they prefer without any limitation due to the classifier. It is possible to see that multiple classifiers which satisfy these conditions such as logistic regression, support vector machines, and random forests are used in the same study to validate the results.

One of the most commonly used classifiers is *logistic regression* [34]. Since it produces predicted probability for each sample, it has been used as a baseline comparison algorithm in benchmarks [25, 27, 35]. The algorithm was used as an example of a modified classifier (in-processing) by Kamishima, Akaho, and Sakuma [41] to eliminate discrimination during the training with regularization. Agarwal et al. [2] have also used logistic regression to apply their reduction approach and compare its performance with other proposed mitigation algorithms. Finally, logistic regression is also used as a testing classifier to compare the performance of a pre-processing or a post-processing technique applied on the dataset [14, 25, 75].

Support vector machine (SVM) [19] is another common classification algorithm that is used widely in algorithmic fairness literature with the same purposes [56, 25, 74]. For example, SVM has been used by Friedler et al. [27] for their benchmarking setup. Furthermore, Feld-

³<https://aif360.mybluemix.net/data>

man et al. [25] have used SVM as one of the oracle classifiers to measure discrimination and accuracy change after they applied their pre-processing approach on the dataset. Also, Ristanoski, Liu, and Bailey [56] have used SVM to implement their discrimination aware loss definition as well as their optimization method, then compared it with standard SVM’s prediction results to measure its performance for imbalanced datasets. Likewise, Zafar et al. [74] have also used SVM to implement their fairness constraints approach to evaluate the performance of their proposed bias mitigation technique.

The third classifier is the *random forest* algorithm [20], which is a collection of decision trees that decides the final outcome by voting among the decision trees within the forest. This classifier has been used by Calmon et al. [14] to fit the pre-processed dataset with their convex optimization technique and then measure the performance of their proposed technique and compare it with other techniques. Salimi et al. [62] have also used random forest classifier as one of the appropriate classifiers to evaluate the performance of their causal database repairing pre-processing technique. Also, Perrone et al. [53] have implemented this classifier to apply their fair Bayesian optimization approach to show the performance and effectiveness of their proposed solution. The study shows that random forest is the best performing classifier among others. Lastly, random forest is one of the chosen algorithms to compare fairness-aware machine learning algorithms systematically in a benchmark framework by Jones et al. [35].

Naive Bayes [77] is another classifier that is used by the researchers who develop bias mitigation algorithms. For example, Calders and Verwer [13] have proposed three different approaches to modify the Naive Bayes classifier or apply post-processing on its predicted outcomes and make it discrimination-free. Kamiran and Calders [38] have used Naive Bayes classifier in order to test the performance of their *massaging* technique, which is changing the labels of some objects to eliminate discrimination in the data, and they used the classifier as a ranker for their *preferential sampling* technique. Feldman et al. [25] have used Gaussian Naive Bayes classifier in order to test the accuracy and measure the discrimination after their data repairing pre-processing approach. Even though Naive Bayes is not used as frequently as logistic regression and SVM, it is still a significant classifier in algorithmic fairness.

Finally, *decision tree* algorithm [55] is the least common classifier used in the literature compared to other classifiers. For instance, it has been used by Kamiran, Calders, and Pechenizkiy [39] in one of their earlier studies to change the splitting criterion and the pruning strategy of the decision tree in order to obtain fair outcomes. As a result, after investigating the literature, we have seen that logistic regression, SVM, and random forest are the most popular classifiers to implement and use either for performance evaluation of the proposed bias mitigation techniques or altering their structure to ensure fair models, which are created for pre-processing, in-processing or post-processing steps in the machine learning pipeline.

Chapter 3

Theoretical background

3.1 Problem Statement

In this thesis, we are proposing this framework to mitigate bias in datasets in pre-processing step in the machine learning pipeline with the goal of obtaining fair predictions from a classifier when it is trained with the pre-processed data. Given a dataset $D = \{X, S, Y\}$, X is the set of attributes that do not contain any sensitive information, S is the set of sensitive attributes containing sensitive information regarding individuals, and $Y \in \{0, 1\}$ is the original class label of individuals, which indicates the decision outcome. Let $|D|$ represent the cardinality of the dataset D . We assume, without loss of generality, that D is an imbalanced dataset where, for example, the groups $G, G' \in S_i$ have a different number of samples, where G and G' represent different values that a sensitive attribute S_i can have. While G represents the protected or unprivileged group, G' represents the unprotected or privileged group. Let \hat{Y} be the predicted class labels of D_{test} derived by a classifier that is trained on D_{train} . In order to deem the predictions of a classifier trained with D_{train} fair as well as satisfactorily accurate, there are a set of fairness measures $F = \{F_{m_1}, F_{m_2}, F_{m_3}, \dots, F_{m_n}\}$ and a set of prediction performance measures $A = \{A_{m_1}, A_{m_2}, A_{m_3}, \dots, A_{m_n}\}$ that need to be satisfied. In this research, we propose a framework to improve the fairness measures while minimizing the loss in performance measures. We will consider five fairness measures $F = \{F_{m_i}, 1 \leq i \leq 5\}$ which are *demographic parity*, *disparate impact*, *equalized odds*, *predictive parity*, and *consistency*, and three prediction performance measures $A = \{A_{m_1}, A_{m_2}, A_{m_3}\}$, which are *accuracy*, *balanced accuracy* and *F1-Score*.

During this research, the following problems will be tackled:

- Quantifying the bias/fairness in datasets or classifier predictions
- Identifying the limitations in the existing bias mitigation techniques proposed in the literature
- Developing a framework that will tackle the identified limitations
- Generating new synthetic samples with high quality for training sets
- Evaluating the effectiveness of the proposed framework in terms of fairness and predictive performance

Demographic Parity: The instances in both protected and unprotected group should have equal probability of being predicted as positive outcome for a classifier to be considered fair according to the Demographic Parity. Thus, the following condition must be satisfied:

$$P \left[\hat{Y} = 1 | S_i = G' \right] = P \left[\hat{Y} = 1 | S_i = G \right].$$

This means that the Demographic Parity difference between two demographic groups should be close to zero, which is denoted as:

$$DP_{diff} = P \left[\hat{Y} = 1 | S_i = G' \right] - P \left[\hat{Y} = 1 | S_i = G \right] \approx 0.$$

Disparate Impact: It is also known as the *80%-rule*. A dataset or a classifier can satisfy disparate impact if the ratio between the probability of protected and unprotected groups getting positive or desired outcomes is at least 0.8. It is formulated as:

$$DI = \frac{P[Y=1|S_i=G]}{P[Y=1|S_i=G']}.$$

Equalized Odds: The instances from protected and unprotected groups should have equal true positive rate (TPR) and false positive rate (FPR), which is denoted as:

$$P \left[\hat{Y} = 1 | S_i = G', Y = 1 \right] = P \left[\hat{Y} = 1 | S_i = G, Y = 1 \right],$$

$$P \left[\hat{Y} = 1 | S_i = G', Y = 0 \right] = P \left[\hat{Y} = 1 | S_i = G, Y = 0 \right].$$

In order to use the Equalized Odds in our experiments, we calculate Average Equalized Odds Difference (AEO Diff.), which is defined as:

$$AEO_{diff} = \frac{(P_1 - P_2) + (P_3 - P_4)}{2},$$

where $P_1 = P \left[\hat{Y} = 1 | S_i = G', Y = 1 \right]$, $P_2 = P \left[\hat{Y} = 1 | S_i = G, Y = 1 \right]$, $P_3 = P \left[\hat{Y} = 1 | S_i = G', Y = 0 \right]$ and $P_4 = P \left[\hat{Y} = 1 | S_i = G, Y = 0 \right]$.

In order to deem a classifier fair based on this measure, the calculated AEO difference should be close to 0.

Predictive Parity: To deem a classifier satisfactory in terms of predictive parity, both protected and unprotected groups should have the same positive predictive value (PPV). It is denoted as:

$$P \left[Y = 1 | \hat{Y} = 1, S_i = G \right] = P \left[Y = 1 | \hat{Y} = 1, S_i = G' \right].$$

To use this measure in our experiments, we calculate Predictive Parity Difference (PP Diff.) between the privileged and the unprivileged groups, which is formulized as the following:

$$PP_{diff} = P \left[Y = 1 | \hat{Y} = 1, S_i = G \right] - P \left[Y = 1 | \hat{Y} = 1, S_i = G' \right].$$

In order to deem a classifier fair based on this measure, the calculated PP difference should be close to 0.

Consistency: This individual fairness measure calculates how similar the labels are for the similar instances in a dataset based on k-neighbors that each instance has. Thus, the instances should have the same or similar labels if they are similar in terms of features. It is formulated as:

$$yNN = 1 - \frac{1}{n} \sum_{i=1}^n \left| \hat{y}_i - \frac{1}{n_{neighbors}} \sum_{j \in \mathcal{N}_{n_{neighbors}}(x_i)} \hat{y}_j \right|.$$

3.2 Improving Fairness Theoretically

This section discusses how the DI measure is improved and then how the improvement in fairness will affect the values of other important measures such as accuracy and F1-Score. Let $|D|$ be the number of instances in the dataset D , N_p be the total number positive examples in the dataset, $N_{G_p}/N_{G'_p}$ be the number positive examples from the unprivileged/privileged groups, respectively. Let ξ be the percentage value of DI for the original dataset ($DI(D) = \xi/100$). Our goal is to increase the value of DI by $\delta/100$, with $0 < \delta < 125 - \xi$, to make $DI(C)$ close to or greater than 80%, where C is a given classifier. To do so, we should increase/decrease the number of instances that are predicted positive from the unprivileged/privileged groups.

Notation	Description
$D(X, S, Y)$	training dataset
$T(X, S, Y)$	testing dataset
X	the set of attributes with non-sensitive information about individuals.
S	the set of attributes with sensitive information.
Y/\hat{Y}	the original/predicted class labels of the instances in a given dataset, respectively.
D_G	$D_G = \{\mathbf{x} \in D \mid S(\mathbf{x}) = G\}$ the set of records with unprivileged values in their sensitive attributes.
$D_{G'}$	$D'_{G'} = \{\mathbf{x} \in D \mid S(\mathbf{x}) = G'\}$ the set of records that have privileged values.
$N_G, N_{G'}$	$N_G = D_G $, $N_{G'} = D_{G'} $.
D_{G_p}	$D_{G_p} = \{\mathbf{x} \in D_G \mid Y(\mathbf{x}) = 1\}$.
$D_{G'_p}$	$D_{G'_p} = \{\mathbf{x} \in D_{G'} \mid Y(\mathbf{x}) = 1\}$.
$N_{G_p}, N_{G'_p}, N_p$	$N_{G_p} = D_{G_p} $, $N_{G'_p} = D_{G'_p} $, $N_p = N_{G_p} + N_{G'_p}$.
F_{m_i}	fairness measure.
A_{m_i}	performance measure.

Table 3.1: Notations used in Chapter 3

If $p(Y(\mathbf{x}) = 1 \mid S(i) = G) = \frac{N_{G_p}}{N_G}$, $p(Y(\mathbf{x}) = 1 \mid S(i) = G') = \frac{N_{G'_p}}{N_{G'}}$, and $DI(D) = \xi\%$ then:

$$\frac{N_{G_p}/N_G}{N_{G'_p}/N_{G'}} = \frac{\xi}{100} \text{ and } N_{G_p} = \frac{\xi N_G N_{G'_p}}{100 N_{G'}}. \quad (3.1)$$

To increase the value of $DI(C)$ to $(\xi + \delta)\%$, we need:

$$\frac{(N_{G_p} + \epsilon)/N_G}{(N_{G'_p} - \gamma)/N_{G'}} = \frac{\xi + \delta}{100}, \quad (3.2)$$

where ϵ is the number of instances (records) from the unprivileged group that should be predicted positive while their original label is negative. Conceptually, ϵ can take any integer value between 0 and $N_G - N_{G_p}$. Conversely, $0 \leq \gamma < N_{G'_p}$ is the number of instances from the privileged group that should be predicted negative while their original label is positive. Solving for ϵ and γ , we get:

$$\frac{(N_{G_p} + \epsilon) N_{G'}}{(N_{G'_p} - \gamma) N_G} = \frac{\xi + \delta}{100}. \quad (3.3)$$

Substituting N_{G_P} from Eq. (3.1) in Eq. (3.3), we get:

$$(\xi + \delta) (N_{G'_P} - \gamma) N_G = 100N_{G'} \left(\frac{\xi N_G N_{G'_P}}{100N_{G'}} + \epsilon \right)$$

Hence:

$$100\epsilon N_{G'} + \gamma (\xi + \delta) N_G = \delta N_{G'_P} N_G \quad (3.4)$$

There are three special cases that can be recognized:

- Case 1:** $\epsilon = \gamma$, in this case we need to increase the number of instances from the protected group that are predicted positive by $\epsilon = \frac{\delta N_{G'_P} N_G}{100N_{G'} + (\xi + \delta) N_G}$ and decrease the number of instances from the unprotected group that are predicted positive by the same number.
- Case 2:** $\gamma = 0$, in this case we need to increase the number of instances from the protected group that are predicted positive by $\epsilon = \frac{\delta N_{G'_P} N_G}{100N_{G'}}$ while keeping the same number of positives from the unprotected group.
- Case 3:** $\epsilon = 0$, in this case we need to decrease the number of instances from the unprotected group that are predicted positive by $\gamma = \frac{\delta N_{G'_P} N_G}{(\xi + \delta) N_G}$ while keeping the same number of positives from the protected group.

Using the formula provided above, it can be easily shown that increasing the positives of the protected group while keeping the number of positive from the unprotected group unchanged (Case 2: $\gamma = 0$) will incur the minimum number of changes because of the assumption that the number of instances from the unprivileged group is smaller than the number of instances from the privileged group. However, this cannot be achieved in reality. In most of the real-life cases, the imbalanced datasets that contain bias have a significantly smaller number of positive instances from the unprivileged group compared to the number of positive instances from the privileged group. It is not the desired option to change the original class labels of the samples from both the unprivileged group that has negative labels and from the privileged group that has positive. Because we do not know which samples were labeled with a negative or positive outcome based on bias and which samples were labeled negatively due to the individuals' ineligibility for the desired outcome. Thus, altering the existing labels of original datasets is not an optimal strategy.

As a solution to this problem, more synthetic samples with positive class labels should be generated using the existing samples from the unprivileged group with positive class labels. This will eventually increase the probability that a classifier predicts the class label of a sample as positive given that it is from the unprivileged group since the classifier will see more positive samples for the unprivileged group during the training step. To generate more positive samples for the unprivileged group, an oversampling technique called SMOTE [15] is used. However, since SMOTE generates the new synthetic samples based on interpolating the original instances in the training set, the quality of the generated instances depends on the similarity (or distance) between the instances they are used in the oversampling. To increase the similarity between the instances, datasets are clustered via a clustering algorithm before generating the new instances (see Section 3.3). In Chapter 5, the number of instances that have been predicted differently after using the proposed mitigation framework are reported in terms of percentages to highlight the effects of the framework on outcomes of the classifiers.

It should be noted that improving the *DI* measure will certainly affect the other measures. For example, according to Eq. (3.2), increasing *DI* by δ will have the following effects depending on the values of ϵ and γ :

- i. The number of True Positives (TP) will be decreased by γ . We assume that we have trained a perfect classifier, which can predict all the labels in the test set correctly. Based on the required changes in the classifier's predictions, if the original true positives is TP then the new true positives $TP' = TP - \gamma$;
- ii. Similarly, the True Negatives will be decreased by ϵ (i.e. $TN' = TN - \epsilon$);
- iii. The False Positives (FP) will be increased by ϵ ($FP' = FP + \epsilon$) and the False Negatives (FN) will be increased by γ ($FN' = FN + \gamma$).

Thus, the perfect classifier's accuracy will be decreased by $\left(\frac{\gamma+\epsilon}{|D|}\right)$. To quantify the changes on the F1-Score, we write it as the following:

$$F1 = \frac{2 * TP}{2 * TP + FN + FP}. \quad (3.5)$$

If $F1'$ is the new *F1-Score* after the mitigation, then it can be calculated as the following:

$$F1' = \frac{2 * (TP - \gamma)}{2 * TP + FN + FP + \epsilon - \gamma}. \quad (3.6)$$

For the case of perfect classifier $F1 = 1$, the new F1 score after mitigation will be $F1' = \frac{2(TP-\gamma)}{2TP-\gamma+\epsilon}$ because TP and FP values were initially 0 (assuming that we have a perfect classifier). Since the initial perfect classifier's F1-Score is 1, the amount of decrease in the *F1-Score* after mitigation will be then $1 - \frac{2(TP-\gamma)}{2*TP-\gamma+\epsilon}$.

3.2.1 Examples of Improving Fairness and the Effects on Performance Measures

The German dataset [22] has 1000 instances. We split the dataset for training and testing using the 70/30 rule with stratification. In the test set T , we have $N_{G'}$ = 181 instances from the privileged group and N_G = 31 from the unprivileged group. The number of positive instances from privileged/unprivileged is ($N_{G'_p}$ = 134)/(N_{G_p} = 17), respectively. The total number of positive instances in T is 210, which includes the other subgroups too. Note that this calculation keeps the other subgroups in the dataset constant. Assuming that we have a perfect classifier's predictions, the Disparate Impact Ratio (DIR) of the predictions of the perfect classifier is calculated as the following:

$$DI(C) = \frac{(17/31)}{(134/181)} = 0.74 \quad (3.7)$$

To improve $DI(C)$ from 74% to be equal to or greater than 80% (which means that $\xi = 74$ and $\delta = 6$) while assuming that we have a perfect classifier. We will find the value for ϵ and γ by using the equation 3.4 and consider three base cases:

Case 1: $\epsilon = \gamma$

$$6 * 31 * 134 = 100 * \epsilon * 181 + \epsilon * 31 * (74 + 6) \quad (3.8)$$

From this equation, we find $\epsilon = 1.21$, but we get the ceiling value to achieve DIR of at least 0.8, which is $\lceil 1.21 \rceil = 2$ for both ϵ and γ . The new DIR after the changes in TP, FP, TN, and FN values due to mitigation will be:

$$DI'(C) = \frac{((17+2)/31)}{((134-2)/181)} \cong 0.84 \quad (3.9)$$

which surpasses the minimum disparate impact threshold of 0.8 for fairness. In this case, the decrease in accuracy will be:

$$\frac{\gamma + \epsilon}{|D|} = \frac{2 + 2}{300} \cong 0.013 \quad (3.10)$$

which corresponds to 1.3% decrease, and the decrease in the F1-Score will be:

$$1 - \frac{2 * (210 - 2)}{210 * 2 - 2 + 2} \cong 0.01 \quad (3.11)$$

which means 1% decrease in F1-Score.

Case 2: $\gamma = 0$

$$6 * 31 * 134 = 100 * \epsilon * 181 + 0 * 31 * (74 + 6) \quad (3.12)$$

this equation yields $\epsilon \cong 1.377$, which means that we need to change $\lceil 1.377 \rceil = 2$ samples' predictions to have a positive class label from the unprivileged group. Such change will make the new DIR become:

$$DI'(C) = \frac{((17 + 2)/31)}{(134/181)} \cong 0.83 \quad (3.13)$$

the decrease in the accuracy and the F1-Score will be calculated similar to the first case, where the decrease in the accuracy will be $2/300 = 0.00\bar{6}$, which is 0.6%, and the decrease in the F1-Score will be $1 - \frac{2 * (210 - 0)}{2 * 210 + 2 - 0} \cong 0.005$ which is 0.5%.

Case 3: $\epsilon = 0$

$$6 * 31 * 134 = 100 * 0 * 181 + \gamma * 31 * (74 + 6) \quad (3.14)$$

where $\gamma = 10.05$ as the solution of this equation. In this case, we need to change the predictions of $\lceil 10.05 \rceil = 11$ samples to have a negative class label from the privileged group to satisfy DIR. Then, the new DIR will be:

$$DI'(C) = \frac{(17/31)}{((134 - 11)/181)} \cong 0.81 \quad (3.15)$$

This change in the DIR will result in a decrease in accuracy with $11/300 = 0.03\bar{6}$, which means 3.6%. Then, the decrease in the F1-Score will be $1 - \frac{2 * (210 - 11)}{2 * 210 - 11 + 0} \cong 0.027$, which corresponds to 2.7%. It is important to note that if the initial DIR of the predictions of a classifier is lower, then we will need to change the predicted class labels of more samples to achieve the minimum threshold of DIR for fairness, which will increase the amount of loss in accuracy and F1-Score.

3.3 Clustering

The clustering step in the COSCFair framework is implemented with the **fuzzy c-means clustering** [8], which is a soft clustering algorithm that allows each sample in a dataset to be assigned to more than one cluster. In fuzzy clustering, each sample belongs to a cluster with a certain probability, which adds up to 1 in total. The algorithm works with the core

ϵ	γ	Used ϵ	Used γ	DIR	Acc.	F1-Score	Bal. Acc.
0	10.05	0	11	0.81	0.963	0.973	0.974
0	10.05	0	10	0.80	0.966	0.976	0.988
1.21	1.21	2	2	0.84	0.986	0.99	0.984
1.21	1.21	1	1	0.79	0.993	0.995	0.992
1.377	0	2	0	0.83	0.993	0.995	0.976
1.377	0	1	0	0.78	0.996	0.998	0.994

Table 3.2: The effects of the values of ϵ and γ with their ceiling and floor values on Disparate Impact Ratio (DIR), Accuracy (Acc.), F1-Score, and Balanced Accuracy (Bal. Acc.). It shows that using the ceiling values of ϵ and γ ensure surpassing the DIR threshold. Note that Balanced Accuracy is calculated as $\frac{TP - \gamma + TN - \epsilon}{2}$, where P and N correspond to the total number of positive and negative samples in the test dataset, respectively.

idea of assigning the samples to clusters in a way that the samples in the same cluster are as similar as possible, while the samples in different clusters are as dissimilar as possible. Clusters are formed based on a distance metric (such as Euclidean distance), which is used to calculate (and minimize the sum of) the distances between the samples and the assigned cluster centroids. Thus, it is important to apply standardization on the numerical features of the datasets in the data preparation step to prevent the unjustified domination of these features with large numerical values on the distances.

Assume that there is a dataset $X = \{x_1, \dots, x_n\}$ with n number of samples. In fuzzy clustering, the existing data points in the training set are randomly initialized into clusters $C = \{c_1, \dots, c_c\}$ based on the chosen number of clusters (c) and calculating the membership matrix $W = w_{i,j} \in [0, 1], i = 1, \dots, n, j = 1, \dots, c$ with the size $(n * c)$, where each element w_{ij} in W tells the degree to which sample x_i belongs to cluster c_j (i.e. the probability of a sample belonging to each cluster). The centroid point of each cluster is identified:

$$c_k = \frac{\sum_x w_k(x)^m x}{\sum_x w_k(x)^m} \quad (3.16)$$

where k represents the k^{th} cluster, and m is the fuzziness parameter, which is chosen as two in most of the cases. The notation $w_k(x)$ refers to the membership probability of sample x to the cluster k . After the cluster centroids are identified, the distances between each sample point and cluster centroids are calculated:

$$\|x_i - c_k\|^2 \quad (3.17)$$

Then, the new membership probabilities of each sample is calculated as the next step to update the membership matrix:

$$w_{ij} = \frac{1}{\sum_{k=1}^c \left(\frac{\|x_i - c_j\|}{\|x_i - c_k\|} \right)^{\frac{2}{m-1}}} \quad (3.18)$$

As mentioned before, the membership probabilities of a sample belonging to each cluster should add up to 1, which means that $w_{i1} + \dots + w_{ic} = 1$. These steps are repeated iteratively

until the algorithm satisfies its objective goal. The objective of this algorithm is to minimize the distances between each sample and the cluster centroids, which means:

$$\operatorname{argmin}_C = \sum_{i=1}^n \sum_{j=1}^c w_{ij}^m \|x_i - c_j\|^2 \quad (3.19)$$

After all the cluster membership probabilities are calculated per sample, the soft clustering can be transformed to hard clustering by assigning samples to the clusters that have the highest probability of membership if desired.

3.4 Oversampling

SMOTE (Synthetic Minority Oversampling Technique) is an oversampling technique that is widely used to create new synthetic samples for the minority class(es) in imbalanced datasets [15]. SMOTE creates new synthetic samples by drawing lines in between the existing samples that are close to each other in feature space and that belong to the same minority class which needs to be oversampled. After the lines are determined, SMOTE produces the synthetic samples along these lines until the number of samples from the minority class becomes equal to the number of samples from the majority class. This procedure is repeated by choosing different original minority samples that exist in the dataset as the core (or main) point and finding its k nearest neighbors.

The neighbor sample points are chosen based on the number k defined by the user, which is a parameter to specify how many nearest neighbors should be taken into account to create new synthetic samples. The minority class is determined based on the total number of samples per decision label in a dataset. For instance, in a classical oversampling case, if a dataset has a binary class label where the positive label has 200 samples while the negative label has 600 samples, then the minority group needs to be oversampled, which are the samples with the positive outcome. Thus, based on the number of neighbors given by the user, for example, $k=3$, SMOTE creates synthetic samples for the minority class between the chosen main points and 3 nearest neighbors to them until the size of it also becomes 600. Figure 3.1 shows how the k nearest neighbors are identified and the lines between them are drawn. Figure 3.2 shows how the new synthetic samples are created along the lines drawn between these k neighbors.

3.5 Classification

In this section, the theoretical aspects of the classification algorithms that are used in the experiments conducted on the COSCFair framework.

Logistic Regression: Logistic regression is a probabilistic statistical model which calculates the probability of a certain event having positive or a negative outcome. The sum of probabilities belonging to the positive and negative class should be 1. Logistic regression is a binary probabilistic classifier, however, it can also be used to predict the probabilities for multiple class labels (known as multinomial logistic regression). In order to formalize how the algorithm works, assume that there are only two variables (x_1, x_2) to predict the outcome of an event with binary outcomes, the outcomes can be either 1 or 0, and we would like to find the probability of an event having the outcome 1, which is denoted as $p = P(Y = 1)$. Then, we also assume that there is a linear relationship between the two variables and the log odds

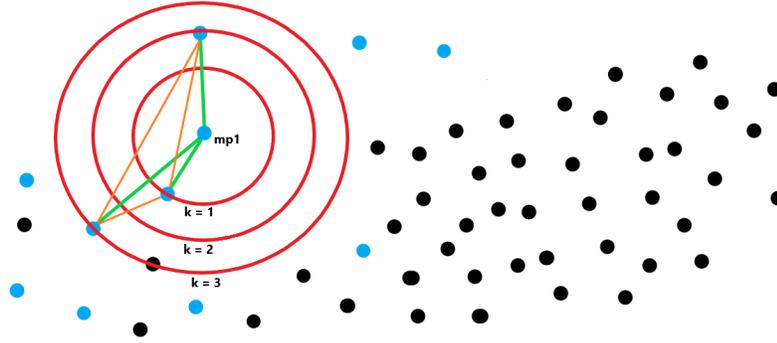


Figure 3.1: The procedure of identifying the nearest neighbors in SMOTE. Blue samples represent the minority class while black samples represent the majority class. The point *mp1* is the main point to start drawing a line to the nearest neighbors. Based on the number k defined by the user, which is 3, the nearest k points are chosen. The green lines are drawn initially from the main point to the nearest neighbors, then orange lines are also drawn between the nearest members before oversampling.

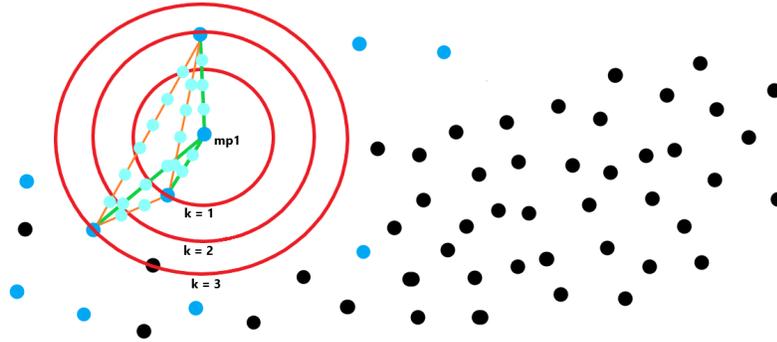


Figure 3.2: The cyan colored samples are the synthetic samples generated on the lines drawn between k neighbors to oversample the minority class (light blue samples).

of the desired outcome, which is $Y = 1$ in this case. This relationship can be formulated as:

$$\log_b \frac{p}{1-p} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 \quad (3.20)$$

where $\log_b \frac{p}{1-p}$ is the log of odds which are p and $p - 1$, b is the base of the algorithm, and $\beta_0, \beta_1, \beta_2$ are the parameters of the model. This equation can be transformed to find the probability of an event having a certain outcome ($Y = 1$) as follows:

$$p = \frac{1}{1 + b^{-\beta_0 + \beta_1 x_1 + \beta_2 x_2}}. \quad (3.21)$$

The calculated probabilities of logistic regression are used to predict the class labels of samples by defining a certain threshold value. In general, when a sample's probability is equal to or greater than 0.5, then the outcome of that sample is predicted as 1. Otherwise, its outcome is predicted as 0. Logistic regression uses a sigmoid function to form the probability distribution.

Random Forest: In easy terms, the random forest algorithm is an ensemble of many decision trees where each tree generates a set of rules to classify samples to obtain more stable and accurate predictions [20]. However, random forest constructs each tree differently from a standard decision tree. Each individual tree is trained with a set that is randomly sampled with replacement from the training set, which is called *Bootstrap Aggregation*, or shortly *Bagging*. Furthermore, each tree is trained with a randomly picked subset of the features in the training set. Such a procedure ensures variation in each decision tree and decreases the possible correlation among trees in the training step. The number of decision trees in a random forest is defined by the user.

Random forest algorithm uses *Gini Index* to decide the splits in its decision trees based on an attribute in each step [52]. Gini Index is a value used to measure the impurity of a specific split. Using this index, one can calculate the impurity reduction that a new split in a decision tree provides. Let T be the training set with binary class label 0 and 1. In order to decide which class ($C_i = \{0, 1\}$) a sample belongs to, the Gini Index of a split for a binary class dataset is calculated as:

$$i(t) = p(C_{i=0}|t)p(C_{i=1}|t), \quad (3.22)$$

or

$$i(t) = p(C_{i=0}|t)(1 - p(C_{i=0}|t)) \quad (3.23)$$

where $i(t)$ denotes impurity, t denotes the node that is being split, and $p(C_i|t)$ represents the probability that the given sample belongs to class C_i . Thus, to calculate the reduction in impurity with a new split s in node t , we calculate:

$$\Delta i(s, t) = i(t) - \{\pi(l)i(l) + \pi(r)i(r)\} \quad (3.24)$$

where l means the left child node and r means the right child node after the new split, $\pi(l)$ and $\pi(r)$ denotes the proportion of samples that are sent to the new left and right child nodes respectively. It is also a noteworthy difference between decision trees and random forests that decision trees prune their leaves back while random forest algorithm does not prune its fully grown trees since generalization error converges due to a large number of trees in this algorithm.

Gradient Boosting classifier: This algorithm is an ensemble technique containing decision trees or regression trees used in classification or regression problems [28]. It combines multiple weak learners in order to minimize the prediction error and thus produce better predictions in the final outcome. The algorithm ensembles the predictions by using the feedback of previous weak learners' feedback to improve the prediction performance, and then makes a final majority voting among all the last weak learners. The mathematical definition of the gradient boosting algorithm will be recalled from Friedman's study [28]. The estimation of the function predicting class label y given the input variable $x = \{x_1, \dots, x_n\}$, which minimizes the output of a loss function can be formulated as follows:

$$\hat{F}(x) = \arg \min_F E_{y,x} L(y, F(x)) \quad (3.25)$$

where the loss function is $L(y, F(x))$, the original function of the distribution is $F(x)$, and $\hat{F}(x)$ denotes the approximation of the original function, which maps x to its outcome label y . Since it is not possible to identify the underlying original function $F(x)$ due to the infinite amount of possibilities, the function space, to search the underlying function, should be limited. It can be accomplished by assuming the underlying function to be a parameterized function $F(x, P)$ where P is a limited set of parameters $P = \{P_1, P_2, \dots\}$. This way, the objective can become optimizing the parameters of the model. The parameter optimization can be formalized as:

$$\hat{P} = \arg \min_P \Phi(P) \quad (3.26)$$

where $\Phi(P)$ is the approximation of the original parameters of the underlying function. Thus, we can show the approximation of the underlying function as:

$$\hat{F}(x) = F(x, \hat{P}) \quad (3.27)$$

Based on the approximation of parameters in the underlying function, the parameters can be found in the algorithm 3.5:

Algorithm 1 Pseudocode of the Gradient Boosting

Input: number of iterations M

Output: Approximated parameters \hat{P}

- 1: $\hat{P} = \hat{P}_0$ //Decide initial approximation values for parameters
 - 2: **for** each $t \in M$ **do**
 - 3: $\nabla L_P(\hat{P}) = \left[\frac{\partial L(y, F(x, P))}{\partial P} \right]_{P=\hat{P}}$ //Calculate the gradient loss function for the current approximation of parameters (\hat{P})
 - 4: $\hat{P}_t \leftarrow -\nabla L_P(\hat{P})$ // Set the current approximation of the iteration t (\hat{P}_t) based on the gradient calculated in this step
 - 5: $\hat{P} \leftarrow \hat{P} + \hat{P}_t = \sum_{i=0}^t \hat{P}_i$ // Update the new approximation of the function parameters
 - 6: $\hat{f}(x) = f(x, \hat{P})$ // Insert the new found approximated parameters in the function
 - 7: **end for**
-

There are three components in gradient boosting classifiers, which are the loss function L to minimize its output, a weak decision tree learner to make the predictions in every iteration, and an additive model that reduces the prediction errors using the loss function's outputs. The loss function changes depending on the type of problem at hand. For example, if the problem is a regression problem, the loss function can be L1, L2, or a quadratic function. If the problem is a classification problem, then the loss function can be binary cross-entropy loss or hinge loss. As an additive model, the weak learners (decision trees) are trained sequentially to reduce the prediction error that came from the previous trees so that a final strong ensemble model is achieved with high performance. None of the weak learners are updated or changed after they are trained in the model. After gradient boosting measures the loss of the prediction of these weak learners, it builds another weak learner tree in the next step by changing the parameters of the previous tree which reduces the measured loss. This procedure is also called *functional gradient descent*.

it is worth mentioning that there is no single classifier that performs the best for all datasets. A classifier might perform well on one dataset while it might perform poorly on another dataset depending on the dataset characteristics such as the number of samples, the number of attributes, and the type of attributes. There are only a few cases where some of the

classification algorithms should or should not be used. For example, if the number of samples is less than the number of attributes in a dataset, then Logistic Regression should not be used. Otherwise, it is going to cause overfitting in the model. Instead, an algorithm that is resilient to overfitting such as Random Forest should be used. However, all the benchmark datasets that are used in the experiments have more samples than the number of attributes. Thus, we deployed in our framework three different types of classifiers to see how they perform on different datasets and compare their results.

Chapter 4

The COSCFair Framework

In this section, the COSCFair framework is presented and the components used to implement each step are introduced. The framework consists of four main components. It starts with the pre-processing step for the dataset, where the **subgroup ID** of each sample is identified. Then, after splitting the dataset into training and test set, a clustering algorithm of choice is applied on the training set to discover the natural groups (clusters). The next step is dividing the training set into **cluster sets**, where the training samples are grouped according to their cluster IDs. Then, we oversample each of these clusters based on the subgroup IDs of the samples to achieve an equal number of samples for each subgroup that exists in the cluster. The final step is the classification step, where the classifier training and class label predictions of the test set occur. Here, we have studied three possible strategies for classifier training and label prediction, which are discussed further in Section 4.4. The pseudocode of the COSCFair framework is given in Algorithm 2 and the high level representation of the framework is presented in Figure 4.1.

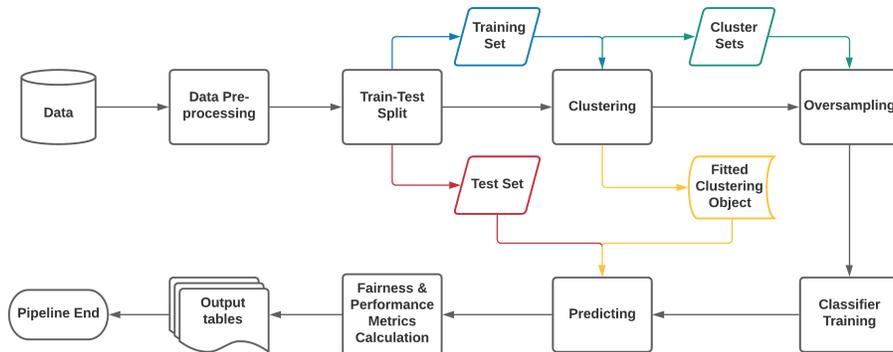


Figure 4.1: The main flowchart of the COSCFair framework

Algorithm 2 Pseudocode of the COSCFair Framework

Input: data $D = \{x_1, \dots, x_n\}$, train-test split ratio ρ , sensitive attributes S , decision attribute Y , Strategy

Output: Fairness and Performance measures' values.

```
1: for each  $x \in D$  do
2:    $G_x \leftarrow \text{subgroup}(x, S, Y)$  //Identify subgroup ID of each sample
3: end for
4:  $A_{G_x} \leftarrow \{G_x, \forall x \in D\}$  //Create an attribute for subgroup IDs
5:  $D_{train}, D_{test} \leftarrow \text{Split}(D, \text{train}, \text{test}, \rho)$ 
6:  $C_1, C_2, \dots, C_m \leftarrow \text{Cluster}(D_{train})$  //find m clusters
7: for Each cluster set  $C_i$  do
8:    $ma = \operatorname{argmax}_{A_{G_x} \subseteq C_i} |A_{G_x}|$ 
9:    $C_i \leftarrow \{\}$ 
10:  for  $A_{G_x}$  in  $C_i$  do
11:     $A'_{G_x} \leftarrow \text{oversample}(A_{G_x}, ma)$ 
12:     $C_i \leftarrow C'_i \cup A'_{G_x}$ 
13:  end for
14: end for
15: if Strategy == 1 then
16:    $D'_{train} \leftarrow \bigcup_{i=1}^m C'_i$ 
17:   Train a model  $M$  using  $D'_{train}$ 
18: end if
19: if Strategy == 2 or Strategy == 3 then
20:   for  $i = 1$  to  $m$  do
21:     train a model  $M_i$  using  $C'_i$  data
22:   end for
23: end if
24: labels  $\leftarrow \{\}$ 
25: for  $x \in D_{test}$  do
26:   labels  $\leftarrow$  labels  $\cup \{(x, \text{class}(x))\}$ 
27: end for
28: for subgroup in subgroups do
29:   Calculate fairness and performance measures
30: end for
```

4.1 Data Preparation

The data preparation step of the COSCFair consists of several sub-steps. These sub-steps include identifying the subgroup IDs, adding this information as a new variable to the dataset and removing the other sensitive attributes, and then splitting the dataset as training and test sets. Other aspects such as deciding the number of samples and which attributes to remove or keep in a dataset are not counted as part of this specific framework since it is a preliminary necessity for every machine learning pipeline. Identifying the subgroup labels of each sample is the most important component of the framework, which we discuss here in more detail. The subgroups in datasets are discovered based on the total number of binary sensitive attributes and the binary decision label, which corresponds to 2^n , where n is the number of subgroups identified in a dataset. In our experiments, we have two binary sensitive attributes and one binary class label in all of the datasets, which corresponds to eight subgroups per dataset. Considering only the sensitive attributes, there are 2^{n-1} subgroups that are identified as the privileged and unprivileged subgroups. For example, with two sensitive attributes, there will

be four main subgroups in each dataset that we compare their outcomes in our experiments.

There are two base groups in the extracted subgroups that are always privileged or unprivileged. If a subgroup has unfavorable values in both sensitive attributes, that subgroup becomes the most unprivileged subgroup in the dataset. If a subgroup has favorable values for both sensitive attributes, then that subgroup becomes the most privileged subgroup. The other subgroups that have different combinations of favorable and unfavorable values for different sensitive attributes are interpreted as both potentially privileged and unprivileged subgroups. Thus, while investigating their position in a dataset, they are compared against the most unprivileged and the most privileged groups. After the subgroup ID variable is added, the sensitive attributes are removed from the dataset to prevent redundancy since the new subgroup IDs already contain information regarding these sensitive attributes.

Finally, if a dataset contains a set of numerical variables, these variables are standardized in training and test sets separately in order to achieve values only between 0 and 1 so that they do not have domination over other variables during the clustering and classification steps. Furthermore, if the dataset contains both categorical and numerical attributes, *Factor Analysis of Mixed Data* (FAMD) is used as the dimensionality reduction technique to represent the dataset in a lower dimensional space [51]. Using dimensionality reduction simplifies the calculations in the next step for the clustering algorithm. In the implementation of the framework, an open source implementation of FAMD [32] is used to reduce the dimensionality in the datasets.

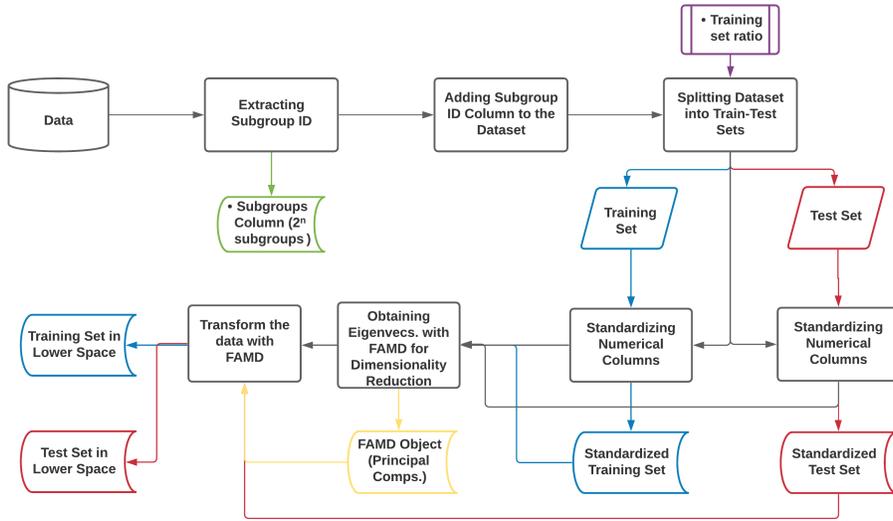


Figure 4.2: Flowchart of pre-processing step of the COSCFair framework. The ratio to split the dataset into training and test sets is given by the user.

4.2 Fuzzy c-means

As mentioned in Section 3.3, the clustering algorithm used in the COSCFair framework is *fuzzy c-means*. This algorithm requires the number of clusters to be given as an input. Thus, the fuzzy c-means algorithm is run multiple times using a predefined list of candidate values as the number of clusters (e.g. 2 to 10) to find the optimal number of clusters for every dataset.

In each run, **fuzzy partition coefficient** (FPC) and the **silhouette score** are computed. FPC is a single value between 0 and 1 which is the dot product of membership matrix W (see Section 3.3) and the transpose of W divided by the number of samples in the dataset. If FPC is close to 1, it means that the clusters are well-formed, the samples belong to their assigned clusters with higher probability values than the other clusters. On the other hand, the silhouette score shows how similar each sample to the samples in its own cluster compared to the samples in the other clusters with a value between -1 and 1. Having a silhouette score close to 1 indicates that the samples are matching well with their clusters. We choose the number of clusters that yield the best combination of these two scores to achieve the highest possible performance.

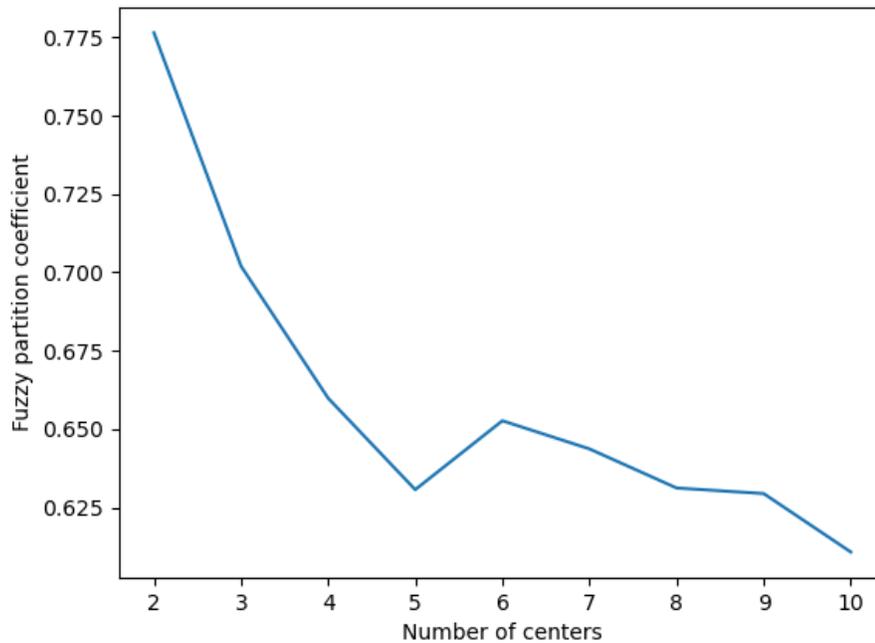


Figure 4.3: An example plot of the fuzzy partition coefficient per number of clusters, which is denoted as number of (cluster) centers, for the Adult dataset. It is used with the silhouette scores plot to decide the optimal number of clusters for the given dataset.

We are using the fuzzy c -means clustering algorithm before oversampling because the synthetic samples are created based on the line between two existing samples (see Section 3.4). Thus, these samples must be close enough to each other to ensure good quality of synthetic sample generation. Therefore, we cluster the training set into smaller cluster sets where the samples used for oversampling in each of these clusters are close to each other, which decreases the distances between the samples that belong to the same subgroup for a better quality and more realistic samples.

Before using fuzzy c -means, it is recommended to use a dimensionality reduction technique such as principal component analysis (PCA) if the whole dataset consists of numerical variables, or the multiple correspondence analysis (MCA) if the dataset consists of only categor-

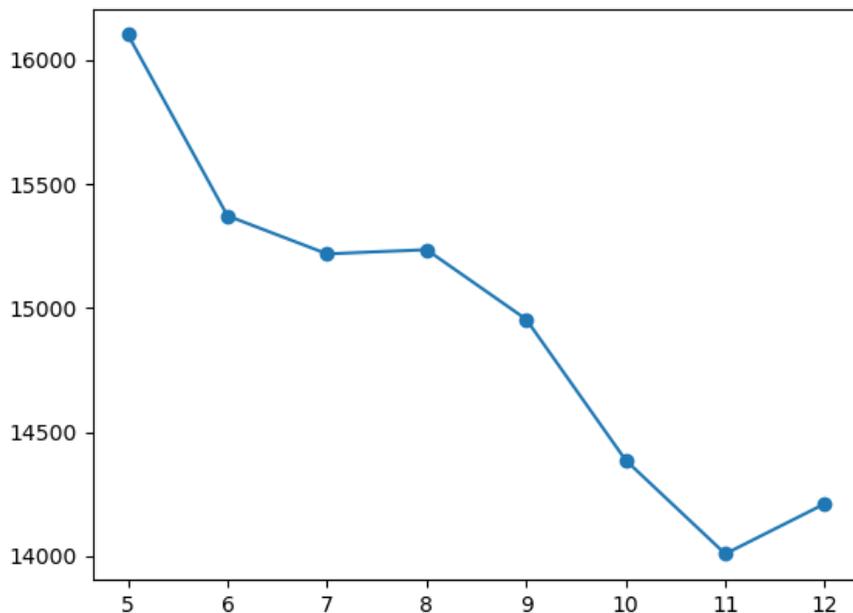


Figure 4.4: An example plot of the silhouette score per number of clusters for the Adult dataset. The x-axis denotes the number of clusters while the y-axis denotes the corresponding silhouette score. It is used together with the fuzzy partition coefficient to decide the optimal number of clusters for a dataset.

ical variables. As mentioned in the previous section, FAMD is used as the dimensionality reduction technique in COSCFair since it is a suitable technique for the datasets with mixed numerical and categorical attributes.

4.3 SMOTE Oversampler

In the COSCFair framework, SMOTE [15] is used as the oversampling technique applied to the cluster sets. After splitting the training set into cluster sets by using the cluster memberships of the training samples, we oversample each cluster set. The oversampling criterion is the subgroup IDs of the samples (2^n), which is identified in the data preparation step. We use subgroup IDs to oversample so that we can obtain an equal representation for each subgroup in each cluster, where they all have precisely the same number of samples with both positive and negative outcomes. We use SMOTE to oversample our clusters, which is described in Section 3.4, although different oversampling algorithms can also be used in this step depending on the characteristics of the training set at hand.

The main reason behind using oversampling is to mitigate bias is that most of the datasets that contain bias are actually imbalanced, where different subgroups are not represented equally, especially in terms of the number of positive and negative samples per subgroup. If a classifier is trained with more positive outcomes than the negative outcomes for a specific

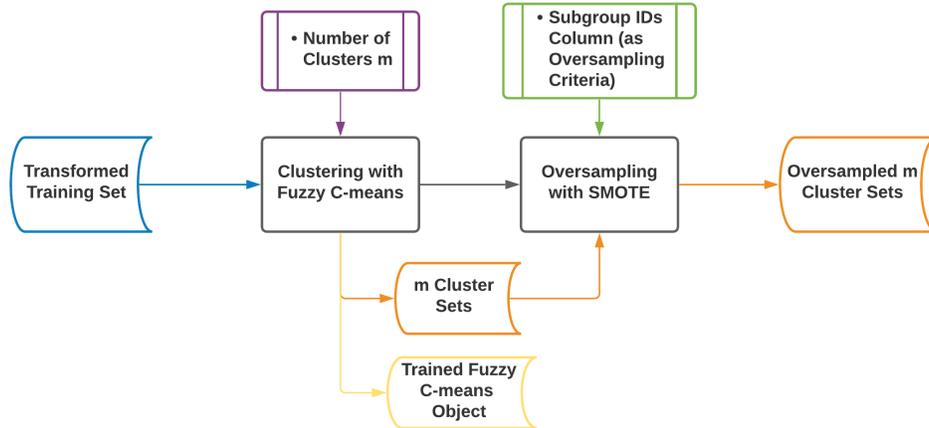


Figure 4.5: The flowchart of clustering and oversampling components of the COSCFair framework. Number of clusters m is given by the user as an input based on the silhouette score and the fuzzy partition coefficient.

subgroup, it will also predict more positive outcomes for that subgroup [72]. Furthermore, if a classifier encounters many samples from a given subgroup, its prediction performance will also increase. On the other hand, if a subgroup is represented with a small number of samples and if most of these samples have a negative outcome, then the classifier trained on such data will have a worse performance at predicting positive outcomes for this subgroup, and most of its predictions for the subgroups will also be negative.

The problem of representing subgroups disproportionately can easily be spotted in Table 5.2 in all of the datasets that are used in this study. The most privileged subgroups have the most number of samples in German and Adult datasets. In addition, the most privileged subgroups in these datasets have more positive class labeled samples than other subgroups. This situation is different in the COMPAS dataset, where the most privileged group has the least number of samples while the most unprivileged group has the most. However, it is important to note that while all other subgroups have more positive labeled samples, the most unprivileged group has more negative labeled samples, which is still an imbalance problem that requires oversampling for equal representation of each subgroup with both positive and negative outcomes.

4.4 Classifiers for Prediction

In this step, a classification algorithm of choice or multiple classification algorithms of the same type (i.e. logistic regression) is trained depending on the strategy that will be followed. The input for the classification step is the oversampled clusters. Depending on the strategy, single or multiple classifiers are trained, then the class labels of the test set are predicted. However, every strategy has its own unique prediction procedure, which will be described in detail in their respective sections. We should note that during classifier training and test set prediction, the sensitive attributes and the subgroup IDs are not used, which ensures *fairness through unawareness*. The COSCFair framework allows users to use any classification algorithm of choice. However, 3 well-known algorithms are recommended to use with this framework, which were discussed in Section 3.5. These classification algorithms are also used

in the experiments. The detailed flowchart of the classification component of the COSCFair framework is given in Figure 4.6.

4.5 Recommended Strategies

4.5.1 Using Single Classification Model

This strategy is the most straightforward and similar one to the mainstream classification training and prediction. After all clusters are oversampled, they are concatenated back together to form a single training set, which is larger than the initial training set. Then, only one classifier of choice is trained with this new training set and the class labels of the transformed test set are predicted based on only this classifier.

4.5.2 Using Cluster Membership for Prediction

This strategy requires training multiple classifiers, which means that one classifier will be trained based on each oversampled cluster. Before the class labels of the test set are predicted, each sample’s cluster membership is predicted by using the fuzzy *c*-means clustering object created in the third step. The clustering object will first calculate the membership probability matrix for each test sample (see Section 3.5). At the beginning of this process, the classifiers that are trained based on the clusters, which do not contain samples from the same subgroup as the test sample at hand are discarded. After that, the remaining classifiers are considered for the rest of the process to find the classifier that is trained with the cluster that the test sample has the highest membership probability. Then, we retrieve the ID of a cluster for the sample. Finally, the classifier object that is trained with that cluster is used to predict the class label of that sample. It is important to note that this procedure is repeated for the class prediction of every test sample individually in an iterative way until all the samples’ classes are predicted.

4.5.3 Using Weighted Cluster Memberships for Prediction

Similar to the second strategy, our final strategy also requires training multiple classifiers using the oversampled clusters. However, instead of choosing one classifier this time, all the trained classifiers are taken into consideration while predicting the class label of a test sample. First, the fuzzy *c*-means clustering object is used to retrieve the membership probability matrix of the test set which contains each sample’s probability of belonging to each cluster. After that, just like the second strategy, some of the cluster IDs are discarded if their clusters do not contain samples from the same subgroup as the test sample. Finally, in the prediction step, the cluster membership probabilities of a sample that are retrieved for the remaining clusters are used as a weighting factor for the predicted class labels from each corresponding classifier.

The weighting is applied to the predicted outcomes by dividing the probability of each eligible cluster C_i by the sum of the probabilities of all eligible clusters and then multiplying the corresponding weight with the predicted outcome of the classifier that is trained with the cluster having the same ID as the cluster C_i . Then, all of the weighted prediction values are summed into a single value. If this value is greater or equal to 0.5, the weighted prediction label becomes 1, otherwise 0. Our experimental results show that this is the best strategy for COSCFair framework, and thus it is used in the final comparison with the other baseline methods.

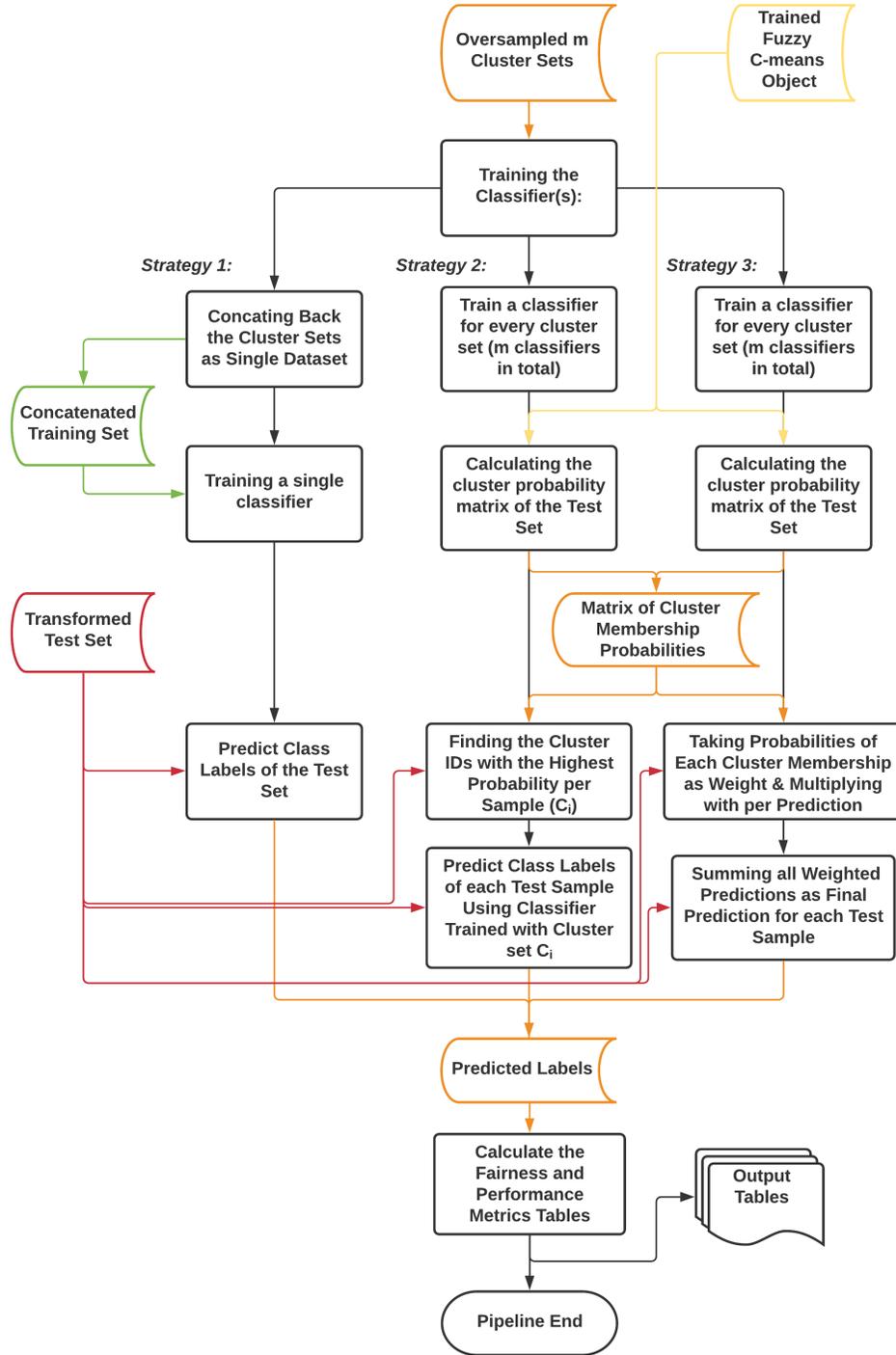


Figure 4.6: Flowchart of the classification component of the COSCFair Framework including all the strategies proposed.

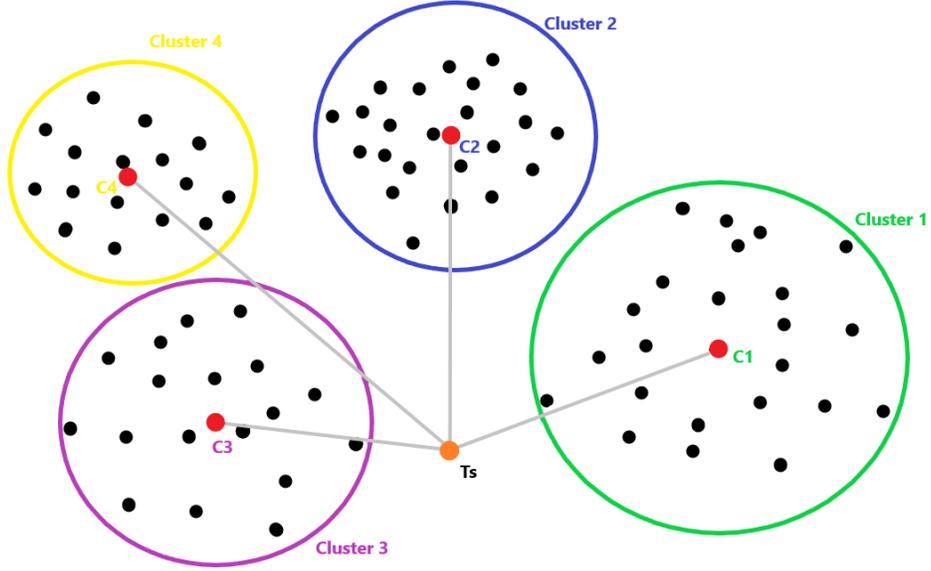


Figure 4.7: The sample T_s colored with orange represents a sample from the test set, black colored points represent the clustered training set, and the red colored points represent the centroid of each cluster. Using fuzzy c-means, the probabilities of T_s belonging to each cluster are calculated based on the distance between T_s and the cluster centroids C_1 , C_2 , C_3 , and C_4 .

This weighing process of this strategy can be formulated as:

$$\sum_{i=0}^{n_{clusts}} \left[\frac{(P(C(T_s) = Cluster_i))}{(\sum_{n=0}^{n_{clusts}} Prob_{cluster_n})} * Pred_{classifier_i} \right] \quad (4.1)$$

Where n_{clusts} indicates the number of eligible clusters left after discarding the clusters that do not contain samples from the same subgroup of the sample at hand, while $P(C(T_s))$ represents the probability of T_s being clustered in $Cluster_i$.

Chapter 5

Evaluation

This chapter provides information regarding datasets and baseline methods and presents the numerical results obtained from the conducted experiments to evaluate the performance of the COSCFair framework in terms of fairness and predictive accuracy. Our focus in the experiments is on improving the DI Ratio, DP Difference, and AEO Difference on all datasets while having minimal or no loss in the rest of the fairness and performance measures.

5.1 Datasets

In order to evaluate the performance of the algorithms in terms of fairness and prediction capabilities, three famous datasets are chosen that are also commonly used in the literature. We have chosen the German Credit dataset as a representative of small datasets, the UCI Adult dataset as a representative of relatively large datasets, which exist in the UCI Machine Learning Repository [22], and finally, the COMPAS dataset which exists in ProPublica Data Store¹. All the columns that contain "Charge Description" in the COMPAS dataset and all the columns that contain "Native Country" in the Adult dataset are removed to reduce the dimensionality of the datasets after one-hot encoding of the categorical variables. Also, "education-num" column is also deleted from the Adult dataset since it is a duplicate information of the column which gives the education level of individuals with categorical values. Finally, all the samples that have missing data from any of the columns are deleted from all datasets (if such samples exist).

All the datasets contain two binary sensitive attributes and a binary decision label in our experiments. The details regarding each dataset can be found in Table 5.1. The favorable sensitive values define the privileged and the unprivileged subgroups in the datasets. For example, while Caucasian males are the most privileged subgroups, African-American females are considered the most unprivileged subgroups in the Adult dataset. The people in-between these subgroups who are having a privileged value in one of the sensitive attributes can be considered privileged or unprivileged depending on the dataset. However, the bias between each of these subgroups should be investigated. The detailed results regarding the bias comparison of all these subgroups can be found in Table 5.6. Since these datasets also exist in AIF 360 with slight improvements such as transforming the continuous sensitive attributes (e.g. age) into an appropriate categorical version, they are invoked from its library instead of importing them as raw CSV files for ease of use.

¹<https://www.propublica.org/datastore/dataset/compas-recidivism-risk-score-data-and-analysis>

Dataset	UCI Adult Income	German Credit	COMPAS Recidivism
Domain	Income	Credit approval	Criminal risk assessment
# of Attributes	14	20	51
# of Instances	48,842	1,000	7,918
# of sensitive attributes	2	2	2
Names of the sensitive attributes	Gender, Race	Age, Sex	Sex, Race
Favorable sensitive value(s)	Male, Caucasian	≥ 25 , Male	Female, Caucasian
Decision labels (desired, undesired)	High Income ($\geq 50k$), Low Income ($< 50k$)	Good Credit, Bad Credit	Did not Recidivate, Did Recidivate
Binary decision labels	yes	yes	yes

Table 5.1: The table showing the datasets that are used in the experiments. Note that the statistics in the table represent the raw versions of the datasets, without any pre-processing such as removing instances with missing values or eliminating some of the attributes.

The first sensitive attribute of the German Credit dataset is *age*. When it is equal to or greater than 25 it is considered as the privileged group and when it is smaller than 25 it is considered as the unprivileged group. The second sensitive attribute of the German Credit dataset is *gender*, namely male being the privileged and female being the unprivileged group. The sensitive attributes of the UCI Adult dataset are *gender* and *race*, where the males and Caucasians are the privileged groups while females and African Americans are the unprivileged groups. Finally, the sensitive attributes of the COMPAS dataset are also *race* and *gender*, but this time Caucasians and females are the privileged groups while males and African Americans are the unprivileged groups.

Datasets	German			Adult			COMPAS		
	Base	Positive	Negative	Base	Positive	Negative	Base	Positive	Negative
attr1: 0, attr2: 0	0.105	0.058	0.047	0.048	0.010	0.039	0.498	0.221	0.277
attr1: 1, attr2: 0	0.205	0.143	0.062	0.217	0.065	0.152	0.307	0.184	0.124
attr1: 0, attr2: 1	0.085	0.052	0.033	0.075	0.035	0.040	0.104	0.066	0.038
attr1: 1, attr2: 1	0.605	0.447	0.158	0.660	0.390	0.269	0.091	0.059	0.032

Table 5.2: The ratios of the demographic subgroups existing in each dataset, together with the ratios of the positive and the negative class labels that subgroup has. The names "attr1" and "attr2" correspond to the sensitive attributes each dataset contains in order (see Table 5.1), the labels 0 and 1 represent the unprivileged and the privileged groups of a given sensitive attribute respectively.

5.2 Measures

There are four statistical fairness measures and an individual fairness measure defined in the Section 3.1. There are also three prediction performance measures are used for the evaluation of the COSCFair framework and the comparison with other baseline techniques.

5.2.1 Fairness Measures

- **Disparate Impact Ratio:** this measure is used to quantify the fairness of original datasets, transformed versions of the datasets after pre-processing, and predicted outcomes from classification algorithms after in and post-processing. Having a disparate impact ratio score of 1 means that the outcomes for both the privileged and the unprivileged groups are fair. The minimum threshold that can be accepted as fair is 0.8 while maximum value should be $1/0.8 = 1.25$ (see Section 3.1).
- **Demographic Parity Difference:** It is also used to measure the fairness of original datasets, transformed versions of the datasets after pre-processing, and predicted outcomes from classification algorithms after in and post-processing. Having a demographic parity difference score of 0 means that the outcomes for both the privileged and the unprivileged groups are fair.
- **Predictive Parity Difference:** This measure is used to measure the fairness of predicted outcomes of classifiers before and after using mitigation algorithms. It cannot be used on the original datasets without any classifier prediction since it requires both the original outcomes and the predicted outcomes. Having 0 as a predictive parity difference indicates that the outcomes are perfectly fair.
- **Average Equalized Odds Difference:** Average EO Difference is also used to measure the fairness of predicted outcomes of classifiers before and after using mitigation algorithms. It cannot be used on the original datasets without any classifier predictions either since it requires both the original outcomes and the predicted outcomes. Having 0 as average equalized odds difference indicates that the outcomes are fair.
- **Consistency:** This measure is used to measure the individual fairness performance of both original datasets (test sets), and the predicted outcomes of classification algorithms after pre-processing, in-processing, and post-processing. If the consistency score of a dataset is 1, it means that the dataset satisfies individual fairness for all instances.

5.2.2 Prediction Performance Measures

- **Accuracy:** Accuracy measures the ratio of the total number of correctly classified positive and negative outcomes to the total number of instances. It is formulated as

$$Acc. = \frac{TP + TN}{P + N}.$$

However, accuracy might be a misleading measure if there is a high-class imbalance in a dataset since the accuracy of the more represented class will dominate the results. For example, assume that there are 100 samples whose class labels are predicted, where 90 of the samples are positive and only 10 of these samples are negative. Even though the classifier would predict all the samples as positive, its accuracy would be 0.9. As a result, it cannot be clearly understood if the classifier is good at predicting the negative labeled samples by only looking at the accuracy score. Thus, the other two measures are going to be used to support the results.

- **Balanced Accuracy:** It is the average of the true positive rate (TPR, a.k.a. recall or sensitivity) and the true negative rate (TNR, a.k.a. specificity). TPR is the ratio of

the true positive predictions and the actual number of positive samples in the predicted set. TNR is the ratio of true negative predictions and the actual number of negative samples in the predicted set. Thus, balanced accuracy is denoted as:

$$Bal.Acc. = \frac{TPR + TNR}{2}.$$

- **F1-Score:** This measure provides the harmonic mean of precision and recall (TPR). Precision is the ratio of the number of true positive predictions to the total number of positive predictions (a.k.a. PPV). It is formulated as

$$F1 - Score = 2 * \frac{TPR * PPV}{TPR + PPV}.$$

5.3 Baseline Mitigation Algorithms

We compare COSCFair against three mitigation algorithms used from the literature. The first baseline method is a pre-processing algorithm called *Learning Fair Representations* developed by Zemel et al. [75] which is explained in Section 2.2.1. The second baseline method is an in-processing algorithm called *Adversarial Debiasing* proposed by Zhang, Lemoine, and Mitchell [76] described in Section 2.2.2. The third baseline algorithm is a post-processing algorithm called *Calibrated Equalized Odds*, which is introduced by Pleiss et al. [54] and discussed in Section 2.2.3.

5.4 Experimental Setup

We have implemented our framework in Python and imported AIF360 library² to call and execute the baseline methods, which are Learning Fair Representations (LFR), Adversarial Debiasing (Adv. Deb.), and Calibrated Equalized Odds (Calibr. EO) in our experiments. We have collected all of the fairness and performance measures explained in Section 3.1 to evaluate the results. We have conducted three main experiments in total. For all of the experiments, each technique is run ten times with a random training and test set split per dataset and then the results are averaged. The standard deviation of each averaged result is also calculated for the baseline methods and the recommended COSCFair Framework. It should be noted that none of the hyperparameters of classifiers in our experiments are fine-tuned for the most optimal predictions, instead, the standard versions of these classifiers are used with no predefined or customized hyper-parameters to provide equality in the experiments.

In the first experiment, three different strategies that can be implemented with COSCFair framework are compared with each other and with a baseline classifier with no mitigation for all datasets to find the most optimal strategy. In order to achieve further insights on how different classifiers affect the performance of these techniques, logistic regression, random forest, and gradient boosting classifiers are used and their results are compared per strategy (see Tables 5.3, 5.4, and 5.5). The outcomes of the most unprivileged (attr1: 0, attr2: 0) and privileged (attr1: 1, attr2: 1) subgroups are used to calculate the fairness measures in this experiment. The predictive performance of the classifiers per strategy are also compared to find the most suitable strategy and the classification algorithm for our framework.

In the second experiment, every possible subgroup combination as privileged and unprivileged groups is compared with each other to investigate the improvement in fairness measures among all these subgroups. In the experiment, only the subgroups that have a favorable value for both sensitive attributes (attr1: 1, attr2: 1) are **always privileged**, and the subgroups having an unfavorable value for both sensitive attributes (attr1: 0, attr2: 0) are **always**

²<https://github.com/Trusted-AI/AIF360>

unprivileged for the comparisons. The other subgroups can be considered privileged and unprivileged groups in the experiments (see Table 5.6). We have extended this experiment by investigating how the percentage of positive and negative predictions change per subgroup when we use COSCFair. The results can be found in Table 5.7.

In the third and final experiment, all the baseline techniques mentioned in Section 5.3 are compared with two variations of our framework(COSCFairLR, COSCFair) based on all the datasets mentioned in Section 5.1. All the fairness measures are calculated also by comparing the most unprivileged (attr1: 0, attr2: 0) and privileged (attr1: 1, attr2: 1) subgroups in this experiment. The average of all fairness and performance measures are provided together with the standard deviation of these 10 times randomized runs.

5.5 Analysis and Results

5.5.1 Comparing Classification Strategies for the Optimal Framework Construction

Evaluating the different classification strategies mentioned in Section 4.5 with the German datasets shows that with logistic regression classifier, the third strategy (COSCFair3) had the best values for most of the fairness measures which are AEO difference, DIR, and DP difference. Since these three measures are the ones that we focus on, it is apparent that the third strategy is the best strategy when used with logistic regression to mitigate bias. In terms of predictive performance, even though the third strategy did not yield the best results, its values were very close to the other strategies' predictive performance results. When the random forest classifier is used, the second (COSCFair2) and the third strategy had competitive values in terms of fairness measures. While COSCFair3 had the best AEO difference, COSCFair2 had the best DIR and DP difference. However, COSCFair3 had better PP difference together with better predictive performance, thus it is possible to say that the third strategy was the best performing one with the random forest classifier. Finally, with the gradient boosting classifier, even though the first strategy (COSCFair1) had the best AEO difference score, the third strategy had the best scores for most of the other fairness measures. The predictive performance of the third strategy was also very close to the performance of the first strategy with less than 0.04 difference for all performance measures. When all predictive performances were compared, the random forest classifier turned out to be the best classifier for the German dataset.

The experiments based on the Adult dataset with logistic regression have shown that even though the output scores are very close to each other, COSCFair3 had the most bias mitigation in terms of AEO difference and DIR. When the predictive performance measures are considered, there was no significant difference between COSCFair1, which had the highest scores, and COSCFair3. When the random forest classifier was used, both COSCFair2 and COSCFair3 had competitively similar results in terms of fairness measures. However, the third strategy has slightly higher scores in predictive performance measures, which was the best among all strategies. Lastly, when gradient boosting classifier was used, COSCFair1 had seemingly the best scores for bias mitigation in terms of DIR, DP difference, and consistency. Especially, the consistency score was improved by 1.6% compared to the original classifier. In predictive performance measures, COSCFair1 and COSCFair3 had competitive scores since both had the same accuracy and balanced accuracy, and COSCFair1 had only 0.002 better F1-score than COSCFair3.

Finally, the experiments with the COMPAS recidivism dataset using logistic regression classifier have shown that COSCFair2 had the most bias mitigation in terms of the first 3 fairness measures and COSCFair3 had the second best bias mitigation performance with very close scores to COSCFair2. However, COSCFair3 had better predictive performance even though

Classifier	Technique	AEO Diff.	DI Ratio	DP Diff.	PP Diff.	Consis.	Acc.	Bal. Acc.	F1 Score
Logistic Regression	No Mitigation	-0.252	0.631	-0.311	-0.064	0.835	0.759	0.675	0.837
	COSCFair1	-0.075	0.768	-0.149	-0.110	0.776	0.701	0.686	0.773
	COSCFair2	-0.053	0.833	-0.114	-0.130	0.746	0.698	0.668	0.775
	COSCFair3	0.021	0.897	-0.045	-0.134	0.745	0.685	0.669	0.759
Random Forest	No Mitigation	-0.112	0.811	-0.162	-0.133	0.834	0.756	0.654	0.839
	COSCFair1	-0.023	0.899	-0.083	-0.166	0.814	0.749	0.661	0.831
	COSCFair2	0.020	0.954	-0.038	-0.175	0.794	0.744	0.658	0.827
	COSCFair3	-0.016	0.911	-0.072	-0.148	0.801	0.749	0.655	0.832
Gradient Boosting Classifier	No Mitigation	-0.096	0.793	-0.167	-0.155	0.811	0.757	0.678	0.835
	COSCFair1	-0.006	0.881	-0.089	-0.184	0.796	0.738	0.685	0.814
	COSCFair2	0.030	0.956	-0.032	-0.149	0.775	0.727	0.664	0.808
	COSCFair3	0.049	0.972	-0.020	-0.166	0.777	0.731	0.668	0.811

Table 5.3: Fairness and performance measures results are compared among the classifiers with COSCFair mitigation strategies and without and mitigation. In the calculations, the most privileged and the most unprivileged groups(attr1:0, attr2:0 vs attr1:1, attr2:1) are considered. The German dataset is used to obtain the results in this table. The highest performances on several fairness and performance measures are highlighted.

Classifier	Technique	AEO Diff.	DI Ratio	DP Diff.	PP Diff.	Consis.	Acc.	Bal. Acc.	F1 Score
Logistic Regression	No Mitigation	-0.260	0.248	-0.489	-0.010	0.937	0.816	0.816	0.821
	COSCFair1	-0.069	0.436	-0.306	-0.180	0.931	0.771	0.771	0.767
	COSCFair2	-0.061	0.438	-0.308	-0.153	0.903	0.768	0.768	0.764
	COSCFair3	-0.060	0.441	-0.307	-0.158	0.904	0.769	0.769	0.766
Random Forest	No Mitigation	-0.202	0.281	-0.433	-0.039	0.863	0.805	0.805	0.806
	COSCFair1	-0.138	0.350	-0.374	-0.116	0.849	0.790	0.790	0.788
	COSCFair2	-0.122	0.368	-0.364	-0.122	0.842	0.791	0.791	0.790
	COSCFair3	-0.126	0.370	-0.365	-0.139	0.845	0.794	0.794	0.793
Gradient Boosting Classifier	No Mitigation	-0.174	0.313	-0.417	-0.081	0.906	0.813	0.813	0.813
	COSCFair1	-0.091	0.403	-0.327	-0.202	0.922	0.795	0.795	0.791
	COSCFair2	-0.087	0.386	-0.336	-0.147	0.890	0.794	0.794	0.789
	COSCFair3	-0.088	0.385	-0.336	-0.148	0.891	0.795	0.795	0.789

Table 5.4: Fairness and performance measures results are compared among the classifiers with COSCFair mitigation strategies and without any mitigation. In the calculations, the most privileged and the most unprivileged groups(attr1:0, attr2:0 vs attr1:1, attr2:1) are considered. The Adult dataset is used to obtain the results in this table. The highest performances on several fairness and performance measures are shown in bold.

its scores were close to COSCFair2’s scores. It should be noted that COSCFair1 had the best predictive performance score among all strategies, but since its fairness measures were lower than both strategies except PP difference, it is not considered as the best strategy. When the random forest classifier is used, the scores of fairness measures were very close to each other for all strategies. However, COSCFair1 had the best scores in most of these measures. Only COSCFair3 had the best PP difference score. In terms of predictive performance, COSCFair3 had the best scores for all measures where its F1-score was even better than the classifier with no mitigation. As the last classifier, when gradient boosting is used, COSCFair2 had the best scores in the first 3 fairness measures. COSCFair3 had the second best scores, which were also very close to the scores of COSCFair2. In addition, COSCFair3 had the best PP difference score. In terms of predictive performance measures, COSCFair3 had slightly better scores than COSCFair2.

Based on the analyses on Tables 5.3, 5.4, and 5.5, the results on all datasets with different

Classifier	Technique	AEO Diff.	DI Ratio	DP Diff.	PP Diff.	Consis.	Acc.	Bal. Acc.	F1 Score
Logistic Regression	No Mitigation	-0.481	0.429	-0.522	-0.035	0.967	0.677	0.673	0.710
	COSCFair1	-0.222	0.621	-0.265	-0.111	0.943	0.644	0.643	0.665
	COSCFair2	-0.172	0.623	-0.223	-0.166	0.937	0.621	0.625	0.613
	COSCFair3	-0.178	0.613	-0.230	-0.162	0.937	0.623	0.626	0.617
Random Forest	No Mitigation	-0.189	0.659	-0.230	-0.120	0.816	0.627	0.625	0.652
	COSCFair1	-0.137	0.719	-0.188	-0.166	0.796	0.626	0.624	0.654
	COSCFair2	-0.149	0.712	-0.196	-0.155	0.795	0.627	0.624	0.656
	COSCFair3	-0.151	0.712	-0.197	-0.150	0.796	0.627	0.625	0.657
Gradient Boosting Classifier	No Mitigation	-0.328	0.532	-0.386	-0.078	0.948	0.679	0.674	0.712
	COSCFair1	-0.201	0.627	-0.263	-0.166	0.915	0.646	0.644	0.669
	COSCFair2	-0.193	0.650	-0.246	-0.162	0.877	0.635	0.632	0.661
	COSCFair3	-0.196	0.646	-0.249	-0.159	0.878	0.636	0.634	0.662

Table 5.5: Fairness and performance measures results are compared among the classifiers with COSCFair mitigation strategies and without and mitigation. In the calculations, the most privileged and the most unprivileged groups(attr1:0, attr2:0 vs attr1:1, attr2:1) are considered. The COMPAS Recidivism dataset is used to obtain the results in this table. The highest performances on several fairness and performance measures are shown in bold.

classifiers show that even though they are competitive, the third strategy (COSCFair3) performs the best compared to the other strategies in both achieving a high DI Ratio and causing the minimal loss in performance measures on average. Even though it looks like Random Forest classifier is not the best combination with COSCFair3 according to Table 5.3, it is the most consistent classifier with our framework in terms of providing high fairness scores (AEO Difference, DI Ratio, and DP Difference) while not causing a significant trade-off in other fairness and performance measures when all of the tested datasets are considered. Thus, we recommend the COSCFair framework to be used with the third strategy (COSCFair3) and Random Forest classifier. We have also added this recommended setup next to the variation with Logistic Regression classifier (COSCFairLR) in Table 5.3 to show its superiority in most of the cases.

5.5.2 Effect of COSCFair on All Subgroups

After all the subgroups in all datasets (German, Adult, COMPAS Recidivism) are compared using different classifiers and COSCFair strategies, it was found that COSCFair3 had the most balanced fairness mitigation for all subgroups since each comparison of each privileged and unprivileged subgroup had closer fairness scores to each other in most of the cases. For example, it can be interpreted in Table 5.9 that the fairness between all the subgroups are highly eliminated (for the changes in predictions per subgroup, see Appendix A.1). The results of using COSCFair3 with Random Forest classifier on German dataset show that all the AEO Differences are lower than 0.06, all the DI Ratios are above the threshold of 0.8, and all the DP Differences are also smaller than 0.1, which means that COSCFair satisfactorily provided fairness in this dataset for all possible combinations of privileged and unprivileged subgroups. However, such precise consistency is not the case with all other classifiers or strategies. For instance, it can be seen on Table 5.8 that DIR scores for each privileged-unprivileged subgroup comparison are not very close to each other, ranging from 0.87 to 1.12. Even though the scores still successfully satisfy the fairness thresholds for all subgroups, there is a significant range between privileged and unprivileged subgroups when logistic regression classifier is used. For more results including the changes in classifiers' predictions per subgroup in terms of positive and negative outcomes, see Appendix A.1 and A.2.

Having values greater than 1.0 in DI Ratio means that the subgroup considered as the un-

privileged group is actually more privileged than the subgroup considered as the privileged group in the equation. For example, in Table 5.6, the DI Ratio on the second row is 1.2 (1.119 precisely), which means that the subgroup "age:1, sex:0" is more privileged than the subgroup "age:0, sex:1". However, since the value is smaller than 1.25 which is the upper threshold for fairness in DIR, it is still considered as satisfactorily fair. The detailed investigation regarding the effect of COSCFair3 on the number of positive and negative predictions per subgroup reveals that COSCFair3 ensures fairness by increasing the positive predictions while decreasing the negative predictions for the unprivileged group(s). It also decreases the positive predictions while increasing the negative predictions for the privileged group(s) compared to the predictions without any bias mitigation, which is shown in Table 5.7.

Subgroups (G vs G')	AEO Diff.	DI Ratio	DP Diff.	PP Diff.	Consis.
A: 0, S: 0 vs A: 1, S: 0	0.021	0.95	-0.04	-0.15	0.80
A: 1, S: 0 vs A: 0, S: 1	0.02	1.20	0.067	0.10	0.80
A: 0, S: 1 vs A: 1, S: 1	-0.06	0.89	-0.10	-0.10	0.80
A: 0, S: 0 vs A: 0, S: 1	0.04	1.06	0.02	-0.05	0.80
A: 1, S: 0 vs A: 1, S: 1	-0.04	0.97	-0.03	0.00	0.80
A: 0, S: 0 vs A: 1, S: 1	-0.02	0.91	-0.07	-0.15	0.80

Table 5.6: Detailed results obtained from the third strategy of COSCFair with Random Forest classifier per unprivileged (G) and privileged (G') subgroup comparison. Sensitive attribute A represents *age* and S represents *sex*. The results are calculated using the German dataset. The performance measures of this result are on Table 5.3 (Random Forest classifier with COSCFair3).

Technique	RF w/o Mitigation			RF with COSCFair		
Ratios	Base	Pos.	Neg.	Base	Pos.	Neg.
age: 0, sex: 0	0.103	0.071	0.032	0.103	0.076	0.027
age: 1, sex: 0	0.207	0.166	0.040	0.207	0.161	0.046
age: 0, sex: 1	0.087	0.066	0.021	0.087	0.062	0.025
age: 1, sex: 1	0.603	0.514	0.089	0.603	0.487	0.116

Table 5.7: The changes in the ratios of the positive (Pos) and negative (Neg) outcomes in each demographic subgroups with and without the COSCFair framework implemented with strategy 3 to predict the German test set using Random Forest (RF) classifier.

<i>Subgroups (G vs G')</i>	AEO Diff.	DI Ratio	DP Diff.	PP Diff.	Consis.
A: 0, S: 0 vs A: 1, S: 0	-0.02	0.88	-0.08	-0.11	0.75
A: 1, S: 0 vs A: 0, S: 1	0.01	1.12	0.06	0.11	0.75
A: 0, S: 1 vs A: 1, S: 1	-0.03	0.87	-0.09	-0.13	0.75
A: 0, S: 0 vs A: 0, S: 1	-0.01	0.97	-0.02	0.00	0.75
A: 1, S: 0 vs A: 1, S: 1	-0.03	0.95	-0.03	-0.02	0.75
A: 0, S: 0 vs A: 1, S: 1	0.02	0.90	-0.05	-0.13	0.75

Table 5.8: Detailed results obtained from the third strategy of COCSFair with Logistic Regression classifier per unprivileged (G) and privileged (G') subgroup comparison. Sensitive attribute A represents *age* and S represents *sex*. The results are calculated using the German dataset. The performance measures of this result are on Table 5.3 (Logistic Regression classifier with COSCFair3).

<i>Subgroups (G vs G')</i>	AEO Diff.	DI Ratio	DP Diff.	PP Diff.	Consis.
A: 0, S: 0 vs A: 1, S: 0	0.05	0.99	-0.01	-0.13	0.78
A: 1, S: 0 vs A: 0, S: 1	-0.02	1.04	0.02	0.10	0.78
A: 0, S: 1 vs A: 1, S: 1	0.03	0.96	-0.03	-0.14	0.78
A: 0, S: 0 vs A: 0, S: 1	0.02	1.03	0.01	-0.03	0.78
A: 1, S: 0 vs A: 1, S: 1	0.00	0.98	-0.01	-0.03	0.78
A: 0, S: 0 vs A: 1, S: 1	0.05	0.97	-0.02	-0.17	0.78

Table 5.9: Detailed results obtained from the third strategy of COCSFair with Gradient Boosting classifier per unprivileged (G) and privileged (G') subgroup comparison. Sensitive attribute A represents *age* and S represents *sex*. The results are calculated using the German dataset. The performance measures of this result are on Table 5.3 (Gradient Boosting classifier with COSCFair3).

<i>Subgroups (G vs G')</i>	AEO Diff.	DI Ratio	DP Diff.	PP Diff.	Consis.
R: 0, S: 0 vs R: 1, S: 0	-0.05	0.66	-0.11	-0.08	0.89
R: 1, S: 0 vs R: 0, S: 1	-0.01	0.70	-0.14	-0.04	0.89
R: 0, S: 1 vs R: 1, S: 1	-0.02	0.84	-0.09	-0.03	0.89
R: 0, S: 0 vs R: 0, S: 1	-0.07	0.46	-0.25	-0.12	0.89
R: 1, S: 0 vs R: 1, S: 1	-0.04	0.59	-0.23	-0.07	0.89
R: 0, S: 0 vs R: 1, S: 1	-0.09	0.38	-0.34	-0.15	0.89

Table 5.10: Detailed results obtained from the third strategy of COCSFair with Gradient Boosting classifier per unprivileged (G) and privileged (G') subgroup comparison. Sensitive attribute R represents *Race* and S represents *sex*. The results are calculated using the Adult dataset. The performance measures of this result are on Table 5.3 (Gradient Boosting classifier with COSCFair3).

5.5.3 Comparison of COSCFair with Baseline Methods

The results of the final experiment indicate that the COSCFair framework with the third strategy (COSCFair3 in Experiments 5.5.1 and 5.5.2) successfully decreases the AEO Difference, DP Difference, while increasing the DI Ratio consistently and not causing a sharp decrease in predictive performance measures unlike the other baseline methods. Depending on the severeness of the bias in the dataset (e.g. Adult), DI Ratio does not always reach the minimum threshold, which is 0.8. However, our solutions outperform the other baseline

<i>Subgroups (G vs G')</i>	AEO Diff.	DI Ratio	DP Diff.	PP Diff.	Consis.
R: 0, S: 0 vs R: 1, S: 0	-0.11	0.77	-0.15	-0.10	0.80
R: 1, S: 0 vs R: 0, S: 1	0.07	1.10	0.06	-0.06	0.80
R: 0, S: 1 vs R: 1, S: 1	-0.11	0.84	-0.11	0.02	0.80
R: 0, S: 0 vs R: 0, S: 1	-0.04	0.85	-0.09	-0.17	0.80
R: 1, S: 0 vs R: 1, S: 1	-0.04	0.93	-0.05	-0.05	0.80
R: 0, S: 0 vs R: 1, S: 1	-0.15	0.71	-0.20	-0.15	0.80

Table 5.11: Detailed results obtained from the third strategy of COCSFair with Random Forest classifier per unprivileged (G) and privileged (G') subgroup comparison. Sensitive attribute R represents *race* and S represents *sex*. The results are calculated using the COMPAS Recidivism dataset. The performance measures of this result are on Table 5.3 (Random Forest classifier with COSCFair3).

<i>Subgroups (G vs G')</i>	AEO Diff.	DI Ratio	DP Diff.	PP Diff.	Consis.
R: 0, S: 0 vs R: 1, S: 0	-0.17	0.69	-0.20	-0.10	0.88
R: 1, S: 0 vs R: 0, S: 1	0.11	1.17	0.10	-0.08	0.88
R: 0, S: 1 vs R: 1, S: 1	-0.14	0.80	-0.14	0.02	0.88
R: 0, S: 0 vs R: 0, S: 1	-0.05	0.81	-0.11	-0.18	0.88
R: 1, S: 0 vs R: 1, S: 1	-0.03	0.93	-0.05	-0.06	0.88
R: 0, S: 0 vs R: 1, S: 1	-0.20	0.65	-0.25	-0.16	0.88

Table 5.12: Detailed results obtained from the third strategy of COCSFair with Gradient Boosting classifier per unprivileged (G) and privileged (G') subgroup comparison. Sensitive attribute R represents *race* and S represents *sex*. The results are calculated using the COMPAS Recidivism dataset. The performance measures of this result are on Table 5.3 (Gradient Boosting classifier with COSCFair3).

methods in most of the cases in terms of both AEO Difference and DI Ratio, which can be seen in Table 5.13. Only in the Adult dataset, LFR outperforms the COSCFairLR variant in DI Ratio with two percent. Furthermore, as mentioned before, the COSCFair framework which uses the Random Forest classifier yields the minimum loss when all of the performance measures in the experiments are compared to other mitigation techniques (LFR, Adversarial Debiasing, and Calibrated Equalized Odds) in most cases. It is found that the other baseline methods perform better at achieving a higher Consistency score, although our framework does not cause a significant decrease in this score compared to the classifiers with no mitigation, which is not more than a 0.1 decrease in most of the cases. The standard deviation scores reveal that our results in different randomized runs provide consistently similar improvements in results compared to other baseline mitigation techniques.

There were several difficulties in the experiments with some of the baseline methods when they were used with specific datasets. For example, Adversarial Debiasing algorithm had a significantly low score of 0.088 with the Adult dataset in DI Ratio because it could not predict any positive outcomes for the unprivileged subgroup in several runs. Furthermore, Adversarial Debiasing caused worse AEO difference and DIR scores compared to a classifier that did not have any mitigation implemented for the German dataset. It also yielded slightly worse results for AEO difference and DIR than the baseline classifier with no mitigation with COMPAS Recidivism dataset. Another difficulty in the experiments were experienced with the Calibrated Equalized Odds technique. Essentially, the user has to provide a criteria for the algorithm to focus on minimizing, so that it can mitigate the bias in the predictions of a classifier, which are False Positive Rate (FPR), False Negative Rate (FNR), or Weighted

(trying to equalize both FPR and FNR between privileged and unprivileged groups simultaneously). Calibrated EO could only output reasonable results with "FPR" as criteria when it was executed with the German dataset. Also, it could work only with "FNR" as criteria when it was executed with the Adult dataset, and with "weighted" as criteria when it was executed with the COMPAS Recidivism dataset. In addition, it performed very poorly with the Adult dataset in terms of predictive performance measures, especially with the F1-Score of 0.169, which is even far worse than random prediction. It also had a significantly bad F1-Score of 0.461 for the COMPAS Recidivism dataset.

Datasets	Technique	AEO Diff.	DI Ratio	DP Diff.	PP Diff.	Consistency	Accuracy	Bal. Acc.	F1 Score
German	Original DF	-	0.748	-0.186	-	0.682	-	-	-
	LR	-0.252/0.089	0.631/0.104	-0.311/0.087	-0.064/0.062	0.835/0.017	0.759/0.016	0.675/0.017	0.837/0.013
	LFR	-0.123/0.162	0.764/0.249	-0.159/0.167	-0.154/0.133	0.985/0.011	0.650/0.041	0.582/0.050	0.745/0.058
	Adv. Deb.	-0.362/0.263	0.570/0.259	-0.352/0.258	-0.041/0.097	0.983/0.009	0.683/0.034	0.540/0.021	0.798/0.031
	Calibr. EO	0.244/0.101	0.740/0.070	0.176/0.067	-0.321/0.070	0.897/0.01	0.524/0.019	0.580/0.018	0.565/0.023
	COSCFairLR	-0.021/0.074	0.897/0.085	-0.045/0.073	-0.134/0.062	0.745/0.016	0.685/0.022	0.669/0.019	0.759/0.022
COSCFair	-0.016/0.072	0.911/0.068	-0.072/0.056	-0.148/0.068	0.801/0.017	0.749/0.017	0.655/0.021	0.832/0.012	
Adult	Original DF	-	0.235	-0.248	-	0.848	-	-	-
	LR	-0.260/0.041	0.248/0.027	-0.489/0.019	-0.010/0.026	0.937/0.002	0.816/0.002	0.816/0.002	0.821/0.002
	LFR	-0.087/0.152	0.463/0.277	-0.201/0.201	-0.272/0.149	0.975/0.017	0.720/0.135	0.688/0.067	0.520/0.085
	Adv. Deb.	-0.238/0.064	0.088/0.115	-0.203/0.025	-0.536/0.155	1.0/0	0.794/0.003	0.673/0.004	0.506/0.008
	Calibr. EO	0.154/0.103	0.784/0.131	0.062/0.15	-0.116/0.066	0.942/0.024	0.755/0.009	0.545/0.070	0.169/0.145
	COSCFairLR	-0.067/0.027	0.443/0.035	-0.311/0.019	-0.175/0.039	0.904/0.004	0.771/0.005	0.771/0.005	0.769/0.007
COSCFair	-0.126/0.018	0.370/0.025	-0.365/0.016	-0.139/0.048	0.845/0.005	0.794/0.006	0.794/0.006	0.793/0.007	
COMPAS	Original DF	-	0.688	-0.201	-	0.675	-	-	-
	LR	-0.484/0.046	0.429/0.027	-0.522/0.041	-0.022/0.025	0.968/0.002	0.679/0.006	0.675/0.007	0.712/0.007
	LFR	-0.211/0.102	0.641/0.117	-0.249/0.108	-0.082/0.043	0.999/0.001	0.647/0.023	0.644/0.019	0.666/0.067
	Adv. Deb.	-0.485/0.072	0.420/0.069	-0.525/0.068	-0.044/0.034	0.998/0.001	0.664/0.015	0.660/0.014	0.696/0.019
	Calibr. EO	-0.505/0.077	0.412/0.044	-0.536/0.065	-0.046/0.038	0.990/0.004	0.558/0.038	0.573/0.032	0.461/0.08
	COSCFairLR	-0.178/0.031	0.613/0.031	-0.230/0.026	-0.162/0.045	0.937/0.007	0.623/0.015	0.626/0.014	0.617/0.018
COSCFair	-0.151/0.034	0.712/0.034	-0.197/0.033	-0.150/0.019	0.796/0.012	0.627/0.012	0.625/0.012	0.657/0.016	

Table 5.13: The comparison results of the baseline logistic regression (LR), other baseline mitigation techniques (LFR, Adv. Deb., and Calibr. EO), our framework trained with logistic regression (COSCFairLR), trained with random forest classifier (COSCFair), and the original dataset. LFR, Adversarial Debiasing and Calibrated EO uses logistic regression as the classifier algorithm in order to provide an equal ground to compare the results. The values on the left side of each cell show the average of ten runs, while the values on the right side give the standard deviation of these ten runs.

Chapter 6

Conclusion

In this chapter, we present the summary of the thesis, the limitation of the proposed framework, and the future directions to improve the COSCFair framework further.

6.1 Summary

In this thesis, imbalanced datasets have been declared as the actual source of bias problem in ML, where there is an unequal representation of different subgroups in terms of positive and negative outcomes in datasets. Most of the existing bias mitigation techniques are dependent on a single and binary sensitive attribute identifying the demographic groups in a dataset. However, it is shown in this thesis that using a single sensitive attribute while there are multiple sensitive attributes in a dataset exacerbates the actual underlying bias. Furthermore, the existence of numerous fairness measures in the literature that are emerged from different points of view has shown that depending on a single fairness measure is not a trustworthy way to identify and mitigate bias.

As a solution to this problem, the COSCFair framework is proposed, which is a bias mitigation technique that has a minimum explicit intervention to the machine learning pipeline since it changes neither the original class labels of a dataset nor any classification algorithm's training structure. It consists of several components, which are the pre-processing of the dataset, clustering the training set, oversampling each clusters, and then applying an ensemble classification technique to predict the class labels of the test set.

The pre-processing step includes the subgroup identification using multiple sensitive attributes, splitting the dataset into training and test sets with stratification, standardizing the numerical attributes between 0 and 1 to prevent any unjustified domination of an attribute, and dimensionality reduction of the training set. Clustering the dataset includes identifying the optimal number of clusters for a given dataset using FPC and Silhouette scores, and after clustering, dividing the training set into corresponding clusters. Oversampling component oversamples every subgroup with both positive and negative outcomes in each cluster. Finally, the classification component consists of training a classifier per cluster and getting the weighted average of all the trained classifiers that are eligible to decide the final prediction of the test set sample's class label.

Multiple experiments are conducted to measure and evaluate the performance of the proposed framework. Several baseline mitigation techniques are identified, multiple fairness measures which contain both group and individual point of view in fairness are used to measure the

framework’s effect on mitigating bias, and several predictive performance measures are also measured to investigate the effect of the proposed framework on prediction capabilities of a classifier. Three possible classification strategies that are developed for this framework are compared with each other to determine the optimal strategy for the classification component. Note that the strategies are completely independent of the classifiers, which means that the strategies can be applied to any classification algorithm. All the experiments conducted in this thesis show that the COSCFair framework provides consistent improvements in achieving higher fairness measures among different subgroups while avoiding significant decrease in classifier performance, which is highly competitive with other solutions in the literature.

COSCFair is a flexible framework because it can easily be integrated with different clustering, oversampling, and classification algorithms to find a customized solution that works best with a given dataset. If the dataset at hand is a completely numeric dataset, standardization of all numerical attributes is highly recommended before proceeding with the other steps. If desired, a dimensionality reduction technique such as Principal Component Analysis could be used on a fully numerical dataset. If a dataset consists of only categorical features, then using more compatible techniques for components such as using k-modes instead of fuzzy c-means as the clustering algorithm and using SMOTENC instead of SMOTE as the oversampler is important to be able to use this framework successfully. Finally, if the dataset at hand is a mix of categorical and numerical features, then using standardization on the numerical features and using an appropriate dimensionality reduction technique are the best choices for the COSCFair framework.

6.2 Answers Found for the Research Questions

The answers to the research sub-questions in this thesis are found in the results of the experiments, which are given in Chapter 5. The results in the first and the second experiments answered the first sub-question by showing that the COSCFair framework could improve the Average Equalized Odds Difference (AEO Difference), Disparate Impact Ratio (DIR), and Demographic Parity Difference (DP Difference) measures in all datasets used in the experiments by 20% on average. There is no consensus on what should be the minimum fairness threshold in the literature for the values of AEO Difference and the DP Difference except the ideal values (which is 0). In many studies, if there is an improvement in these measures compared to the unmitigated version of the results, and if the results are better than the other baseline methods, it is considered that the proposed solution could mitigate bias successfully [72, 40]. Based on this, it is possible to say that COSCFair significantly improves the 3 out of 5 chosen fairness measures calculated on a classifier’s predictions by at least 20%.

The second sub-question is answered by the final experiment. The COSCFair framework could outperform all of the baseline techniques (LFR, Adversarial Debiasing, Calibrated EO) on the German Credit and COMPAS Recidivism datasets in terms of AEO Difference, DIR, and DP Difference, without causing a significant loss in any of the performance by no more than 10% (0.1). Finally, the main research question is answered in Chapter 4, where the whole framework and its components are explained in detail. We developed a bias mitigation framework called COSCFair by combining integrated pre-processing and ensembled classification procedures which consist of several components.

6.3 Limitations

There are several limitations identified in the COSCFair framework. The first limitation is that the quality of the results is dependent on the synthetic data generation technique implemented in the framework. The ability of the technique to produce high-quality synthetic samples (i.e. realistic samples) might have an effect on the classifier training, which might

eventually affect the predictions of the classifier at hand. Another limitation of the framework is related to the clustering component. Which clustering technique is used and the number of clusters chosen for a dataset will affect the performance of the overall framework since oversampling is based on the clusters formed by the clustering algorithm. Finally, the dimensionality reduction technique that is used in pre-processing step will also affect the quality of the framework since the explained variance obtained from the transformed dataset after the dimensionality reduction technique will decide how much of the information from the original dataset could be kept in the transformed dataset. If the explained variance is low, then the information that the transformed dataset will contain regarding the original dataset is small. It means that the quality of the dataset will be low, and synthetically produced samples will potentially have even lower quality.

6.4 Ethical Considerations

Developing classification models trained with a dataset that has information regarding real individuals brings the need for ethical considerations in such procedures. In order to make ethically acceptable decisions, these decisions should not rely on or be affected by any intrinsic information about the individuals. For this reason, ensuring fairness in decision-making algorithms is indispensable. First of all, it is considered unethical to process or use any information regarding an individual’s demographic or intrinsic characteristics (which are defined by the sensitive attributes) while deciding their outcomes. Furthermore, it is forbidden to use the sensitive attributes defined by the law such as race, gender, and/or religion. Thus, the sensitive attributes must be removed from the decision variables before classifier training.

There are two sensitive attributes in each dataset that are used in the evaluation of our framework. The German Credit dataset has sex and age group, and both COMPAS Recidivism and the Adult Income datasets have race and sex as the sensitive attributes. In the COSCFair framework, we are also removing the sensitive attributes in these datasets including the subgroup IDs of the individuals before training a classifier. It should be noted that even though race and sex attributes are deleted from the COMPAS Recidivism dataset, there are still other attributes containing private information about the individuals that will trigger ethical problems such as name and surname. Such information regarding individuals should also be anonymized before the dataset is used in order to ensure ethical data processing. However, Adult and German datasets do not contain any private information, thus they do not raise such an ethical concern after their sensitive attributes are removed.

Unfortunately, removing sensitive attributes is not enough to eliminate ethical concerns in the decision-making context. Some “unintentional bias” could still exist in the decisions of a classifier after removing the sensitive attributes. The amount of bias is determined by law by checking the existence of *Disparate Impact* (explained in Section 2.1.1) in the decisions of a classifier. For example, if a business hiring decision results in disparate impact by race even if the decision was not explicitly determined based on race, such hiring decision is found illegal by the US Supreme Court since the decisions are unethical and unfair. Another example can be given from the datasets that are used in our experiments. Even though the Adult and German datasets are partly anonymized and the sensitive attributes are removed so that it is not possible to find the individuals based on the information provided in the data, the datasets still reflect bias in terms of Disparate Impact. It is because some values in the other attributes in these datasets still contain some information regarding the values of the sensitive attributes. The COSCFair framework eliminates this ethical concern by mitigating the Disparate Impact Ratio in the predictions of classifiers by creating an equal representation of the privileged and the unprivileged groups in terms of both positive and negative outcomes in training sets via synthetically created samples.

6.5 Future Work

There are several directions to be investigated further to improve the COSCFair framework. One of these directions is investigating the effects of different oversampling techniques and the quality of synthetic samples on the fairness and performance measures. Achieving higher quality or more realistic synthetic samples might improve the results in fairness measures further while preserving high values in performance measures. Another important direction is studying the cases where the sensitive attributes are not binary and the cases where there are more than two sensitive attributes. From the conducted experiments, it is discovered that the datasets that have an extremely high number of negative outcomes might require a customized oversampling procedure to achieve higher fairness measure scores. Thus, another future direction could be working on the effects of oversampling the samples with positive outcomes of the unprivileged groups instead of using an overall oversampling for both outcomes of each subgroup in clusters. The classifiers that were used in the experiments did not have any hyperparameter tuning to ensure an equal ground for the comparison of the COSCFair framework with other baselines. Therefore, studying the effects of fine-tuning the hyperparameters of classifiers on the fairness and predictive performance measures when they are used with the COSCFair framework might also be an important future direction. Finally, studying the effect of the borderline samples in clusters (which are at similar distances to different cluster centers) on the quality of the synthetic samples produced by an oversampling technique could be a good direction to improve the COSCFair framework.

Bibliography

- [1] Julius A Adebayo et al. “FairML: ToolBox for diagnosing bias in predictive modeling”. PhD thesis. Massachusetts Institute of Technology, 2016.
- [2] Alekh Agarwal, Alina Beygelzimer, Miroslav Dudík, John Langford, and Hanna Wallach. “A reductions approach to fair classification”. In: *International Conference on Machine Learning*. PMLR. 2018, pages 60–69.
- [3] Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. “Machine Bias”. In: *ProPublica* (May 2016). URL: <https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing>.
- [4] Niels Bantilan. “Themis-ml: A fairness-aware machine learning interface for end-to-end discrimination discovery and mitigation”. In: *Journal of Technology in Human Services* 36.1 (2018), pages 15–30.
- [5] Alistair Barr. “Google Mistakenly Tags Black People as ‘Gorillas,’ Showing Limits of Algorithms”. In: *The Wall Street Journal* (July 2015). URL: <https://www.wsj.com/articles/BL-DGB-42522>.
- [6] Rachel K. E. Bellamy, Kuntal Dey, Michael Hind, Samuel C. Hoffman, Stephanie Houde, Kalapriya Kannan, Pranay Lohia, Jacquelyn Martino, Sameep Mehta, Aleksandra Mojsilovic, Seema Nagar, Karthikeyan Natesan Ramamurthy, John Richards, Diptikalyan Saha, Prasanna Sattigeri, Moninder Singh, Kush R. Varshney, and Yunfeng Zhang. *AI Fairness 360: An Extensible Toolkit for Detecting, Understanding, and Mitigating Unwanted Algorithmic Bias*. Oct. 2018. URL: <https://arxiv.org/abs/1810.01943>.
- [7] Richard Berk, Hoda Heidari, Shahin Jabbari, Michael Kearns, and Aaron Roth. “Fairness in criminal justice risk assessments: The state of the art”. In: *Sociological Methods & Research* (2018), page 0049124118782533.
- [8] James C Bezdek, Robert Ehrlich, and William Full. “FCM: The fuzzy c-means clustering algorithm”. In: *Computers & geosciences* 10.2-3 (1984), pages 191–203.
- [9] Sarah Bird, Miro Dudík, Richard Edgar, Brandon Horn, Roman Lutz, Vanessa Milan, Mehrnoosh Sameki, Hanna Wallach, and Kathleen Walker. *Fairlearn: A toolkit for assessing and improving fairness in AI*. Technical report. Technical Report MSR-TR-2020-32, Microsoft., May 2020.
- [10] Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. “Man is to computer programmer as woman is to homemaker? debiasing word embeddings”. In:
- [11] Joy Buolamwini and Timnit Gebru. “Gender shades: Intersectional accuracy disparities in commercial gender classification”. In: *Conference on Fairness, Accountability and Transparency (FAT)*. PMLR. 2018, pages 77–91.

- [12] Toon Calders, Faisal Kamiran, and Mykola Pechenizkiy. “Building classifiers with interdependency constraints”. In: *2009 IEEE International Conference on Data Mining Workshops*. IEEE, 2009, pages 13–18.
- [13] Toon Calders and Sicco Verwer. “Three naive bayes approaches for discrimination-free classification”. In: *Data Mining and Knowledge Discovery* 21.2 (2010), pages 277–292.
- [14] Flavio P Calmon, Dennis Wei, Bhanukiran Vinzamuri, Karthikeyan Natesan Ramamurthy, and Kush R Varshney. “Optimized pre-processing for discrimination prevention”. In: *Proceedings of the NIPS’17*, pages 3995–4004.
- [15] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. “SMOTE: synthetic minority over-sampling technique”. In: *Journal of artificial intelligence research* 16 (2002), pages 321–357.
- [16] Alexandra Chouldechova. “Fair prediction with disparate impact: A study of bias in recidivism prediction instruments”. In: *Big data* 5.2 (2017), pages 153–163.
- [17] Alexandra Chouldechova and Aaron Roth. “A snapshot of the frontiers of fairness in machine learning”. In: *Communications of the ACM* 63.5 (2020).
- [18] Sam Corbett-Davies, Emma Pierson, Avi Feller, Sharad Goel, and Aziz Huq. “Algorithmic decision making and the cost of fairness”. In: *Proceedings of the 23rd ACM SIGKDD International Conference on KDD’17*, pages 797–806.
- [19] Corinna Cortes and Vladimir Vapnik. “Support-vector networks”. In: *Machine learning* 20.3 (1995), pages 273–297.
- [20] Adele Cutler, D Richard Cutler, and John R Stevens. “Random forests”. In: *Ensemble machine learning*. Springer, 2012, pages 157–175.
- [21] Amit Datta, Michael Carl Tschantz, and Anupam Datta. “Automated experiments on ad privacy settings: A tale of opacity, choice, and discrimination”. In: *Proceedings on privacy enhancing technologies* 2015.1 (2015), pages 92–112.
- [22] Dheeru Dua and Casey Graff. *UCI Machine Learning Repository*. 2017. URL: <http://archive.ics.uci.edu/ml>.
- [23] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. “Fairness through awareness”. In: *Proceedings of the 3rd innovations in theoretical computer science conference*. 2012, pages 214–226.
- [24] The U.S. EEOC. *Uniform guidelines on employee selection procedures*. Mar. 1979.
- [25] Michael Feldman, Sorelle A Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. “Certifying and removing disparate impact”. In: *Proceedings of the 21th ACM SIGKDD International Conference on KDD’15*, pages 259–268.
- [26] Sorelle A Friedler, Carlos Scheidegger, and Suresh Venkatasubramanian. “On the (im) possibility of fairness”. In: *arXiv preprint arXiv:1609.07236* (2016).
- [27] 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”. In: *Proceedings of the conference on fairness, accountability, and transparency*. 2019, pages 329–338.
- [28] Jerome H Friedman. “Greedy function approximation: a gradient boosting machine”. In: *Annals of statistics* (2001), pages 1189–1232.
- [29] Pratik Gajane and Mykola Pechenizkiy. “On formalizing fairness in prediction with machine learning”. In: *arXiv preprint arXiv:1710.03184* (2017).
- [30] Sainyam Galhotra, Yuriy Brun, and Alexandra Meliou. “Fairness Testing: Testing Software for Discrimination”. In: *Proceedings of 2017 11th Joint Meeting of*

- the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering*. 2017.
- [31] Sainyam Galhotra, Yuriy Brun, and Alexandra Meliou. “Fairness testing: testing software for discrimination”. In: *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering*. 2017, pages 498–510.
 - [32] Max Halford. Oct. 2020. URL: <https://github.com/MaxHalford/prince#factor-analysis-of-mixed-data-famd>.
 - [33] Moritz Hardt, Eric Price, and Nati Srebro. “Equality of Opportunity in Supervised Learning”. In: *NIPS’16*.
 - [34] David W Hosmer Jr, Stanley Lemeshow, and Rodney X Sturdivant. *Applied logistic regression*. Volume 398. John Wiley & Sons, 2013.
 - [35] Gareth P Jones, James M Hickey, Pietro G Di Stefano, Charanpal Dhanjal, Laura C Stoddart, and Vlasios Vasileiou. “Metrics and methods for a systematic comparison of fairness-aware machine learning algorithms”. In: *arXiv preprint arXiv:2010.03986* (2020).
 - [36] Matthew Joseph, Michael Kearns, Jamie Morgenstern, and Aaron Roth. “Fairness in learning: Classic and contextual bandits”. In: *arXiv preprint arXiv:1605.07139* (2016).
 - [37] Faisal Kamiran and Toon Calders. “Classification with no discrimination by preferential sampling”. In: *Proc. 19th Machine Learning Conf. Belgium and The Netherlands*. Citeseer. 2010, pages 1–6.
 - [38] Faisal Kamiran and Toon Calders. “Data preprocessing techniques for classification without discrimination”. In: *Knowledge and Information Systems* 33.1 (2012), pages 1–33.
 - [39] Faisal Kamiran, Toon Calders, and Mykola Pechenizkiy. “Discrimination aware decision tree learning”. In: *2010 IEEE International Conference on Data Mining*. IEEE. 2010, pages 869–874.
 - [40] Faisal Kamiran, Asim Karim, and Xiangliang Zhang. “Decision theory for discrimination-aware classification”. In: *2012 IEEE 12th International Conference on Data Mining*. IEEE. 2012, pages 924–929.
 - [41] Toshihiro Kamishima, Shotaro Akaho, and Jun Sakuma. “Fairness-aware learning through regularization approach”. In: *2011 IEEE 11th International Conference on Data Mining Workshops*. IEEE. 2011, pages 643–650.
 - [42] Michael Kearns, Seth Neel, Aaron Roth, and Zhiwei Steven Wu. “Preventing fairness gerrymandering: Auditing and learning for subgroup fairness”. In: *ICML’18*. 2018, pages 2564–2572.
 - [43] Niki Kilbertus, Mateo Rojas-Carulla, Giambattista Parascandolo, Moritz Hardt, Dominik Janzing, and Bernhard Schölkopf. “Avoiding discrimination through causal reasoning”. In: *Proceedings of NIPS’17*, pages 656–666.
 - [44] Michael P Kim, Omer Reingold, and Guy N Rothblum. “Fairness through computationally-bounded awareness”. In: *Proceedings of the 32nd International Conference on NIPS’18*, pages 4847–4857.
 - [45] Jon Kleinberg, Sendhil Mullainathan, and Manish Raghavan. “Inherent Trade-Offs in the Fair Determination of Risk Scores”. In: *8th Innovations in ITCS’17*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.
 - [46] MJ Kusner, J Loftus, Christopher Russell, and R Silva. “Counterfactual Fairness”. In: *NIPS’17*. 2017.
 - [47] Preethi Lahoti, Krishna P Gummadi, and Gerhard Weikum. “ifair: Learning individually fair data representations for algorithmic decision making”. In: *IEEE*

- 35th International Conference on Data Engineering (ICDE'19)*. IEEE. 2019, pages 1334–1345.
- [48] Rafi Letzter. “Amazon just showed us that ‘unbiased’ algorithms can be inadvertently racist”. In: *Insider* (Apr. 2016). URL: <https://www.businessinsider.com/how-algorithms-can-be-racist-2016-4?international=true&r=US&IR=T>.
- [49] Joshua R Loftus, Chris Russell, Matt J Kusner, and Ricardo Silva. “Causal reasoning for algorithmic fairness”. In: *arXiv preprint arXiv:1805.05859* (2018).
- [50] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. “A survey on bias and fairness in machine learning”. In: *arXiv preprint arXiv:1908.09635* (2019).
- [51] Jérôme Pagès. “Analyse factorielle de donnees mixtes: principe et exemple d’application”. In: *Montpellier SupAgro*, <http://www.agro-montpellier.fr/sfds/CD/textes/pages1.pdf> (2004).
- [52] Mahesh Pal. “Random forest classifier for remote sensing classification”. In: *International journal of remote sensing* 26.1 (2005), pages 217–222.
- [53] Valerio Perrone, Michele Donini, Krishnaram Kenthapadi, and Cédric Archambeau. “Bayesian Optimization with Fairness Constraints”. In: (2020).
- [54] Geoff Pleiss, Manish Raghavan, Felix Wu, Jon Kleinberg, and Kilian Q Weinberger. “On fairness and calibration”. In: *Proceedings of the 31st International Conference on NIPS'17*, pages 5684–5693.
- [55] J. Ross Quinlan. “Induction of decision trees”. In: *Machine learning* 1.1 (1986), pages 81–106.
- [56] Goce Ristanoski, Wei Liu, and James Bailey. “Discrimination aware classification for imbalanced datasets”. In: *Proceedings of the 22nd ACM CIKM'13*, pages 1529–1532.
- [57] Andrea Romei and Salvatore Ruggieri. “A multidisciplinary survey on discrimination analysis”. In: *The Knowledge Engineering Review* 29.5 (2014), pages 582–638. DOI: 10.1017/S0269888913000039.
- [58] Adam Rose. “Are Face-Detection Cameras Racist?” In: *TIME* (Jan. 2010). URL: <http://content.time.com/time/business/article/0,8599,1954643,00.html>.
- [59] Cynthia Rudin. “Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead”. In: *Nature Machine Intelligence* 1.5 (2019), pages 206–215.
- [60] 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”. In: *arXiv preprint arXiv:1811.05577* (2018).
- [61] Babak Salimi, Luke Rodriguez, Bill Howe, and Dan Suciu. “Capuchin: Causal database repair for algorithmic fairness”. In: *arXiv preprint arXiv:1902.08283* (2019).
- [62] Babak Salimi, Luke Rodriguez, Bill Howe, and Dan Suciu. “Interventional fairness: Causal database repair for algorithmic fairness”. In: *Proceedings of the 2019 International Conference on Management of Data*. 2019, pages 793–810.
- [63] Nripsuta Ani Saxena, Karen Huang, Evan DeFilippis, Goran Radanovic, David C Parkes, and Yang Liu. “How do fairness definitions fare? Examining public attitudes towards algorithmic definitions of fairness”. In: *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*. 2019, pages 99–106.

- [64] Kacper Sokol, Alexander Hepburn, Rafael Poyiadzi, Matthew Clifford, Raul Santos-Rodriguez, and Peter Flach. “FAT Forensics: A Python Toolbox for Implementing and Deploying Fairness, Accountability and Transparency Algorithms in Predictive Systems”. In: *Journal of Open Source Software* 5.49 (2020), page 1904. DOI: 10.21105/joss.01904. URL: <https://doi.org/10.21105/joss.01904>.
- [65] Spencer Soper. “Amazon to Bring Same-Day Delivery to Roxbury After Outcry”. In: *Bloomberg* (Apr. 2016). URL: <https://www.bloomberg.com/news/articles/2016-04-26/amazon-to-bring-same-day-delivery-to-roxbury-after-outcry>.
- [66] Latanya Sweeney. “Discrimination in online ad delivery”. In: *Communications of the ACM* 56.5 (2013), pages 44–54.
- [67] Florian Tramer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, Jean-Pierre Hubaux, Mathias Humbert, Ari Juels, and Huang Lin. “Fairtest: Discovering unwarranted associations in data-driven applications”. In: *2017 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE. 2017, pages 401–416.
- [68] Sahil Verma and Julia Rubin. “Fairness definitions explained”. In: *2018 IEEE/ACM international workshop on software fairness (fairware)*. IEEE. 2018, pages 1–7.
- [69] J. Wexler, M. Pushkarna, T. Bolukbasi, M. Wattenberg, F. Viégas, and J. Wilson. “The What-If Tool: Interactive Probing of Machine Learning Models”. In: *IEEE Transactions on Visualization and Computer Graphics* 26.1 (2020), pages 56–65. DOI: 10.1109/TVCG.2019.2934619.
- [70] Rüdiger Wirth and Jochen Hipp. “CRISP-DM: Towards a standard process model for data mining”. In: *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining*. Volume 1. Springer-Verlag London, UK. 2000.
- [71] Blake Woodworth, Suriya Gunasekar, Mesrob I Ohannessian, and Nathan Srebro. “Learning non-discriminatory predictors”. In: *Conference on Learning Theory*. PMLR. 2017, pages 1920–1953.
- [72] Shen Yan, Hsien-te Kao, and Emilio Ferrara. “Fair Class Balancing: Enhancing Model Fairness without Observing Sensitive Attributes”. In: *Proceedings of the 29th ACM International CIKM’20*, pages 1715–1724.
- [73] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rodriguez, and Krishna P Gummadi. “Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment”. In: *Proceedings of WWW’17*, pages 1171–1180.
- [74] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rogriguez, and Krishna P Gummadi. “Fairness constraints: Mechanisms for fair classification”. In: *Artificial Intelligence and Statistics*. PMLR. 2017, pages 962–970.
- [75] Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. “Learning fair representations”. In: *ICML’13*. PMLR, pages 325–333.
- [76] Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. “Mitigating unwanted biases with adversarial learning”. In: *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*. 2018, pages 335–340.
- [77] Harry Zhang. “The optimality of naive Bayes”. In: *AAAI*. 2004.

Appendix A

Appendix: More Tables From Experimental Results

A.1 Ratio Tables of Strategy 3

Technique	GB w/o Mitigation			GB with COSCFair3		
Ratios	Base	Pos.	Neg.	Base	Pos.	Neg.
race: 0,sex: 0	0.049	0.008	0.041	0.049	0.010	0.039
race: 1, sex: 0	0.217	0.058	0.159	0.217	0.069	0.148
race: 0, sex: 1	0.074	0.038	0.036	0.074	0.034	0.040
race: 1, sex: 1	0.660	0.425	0.234	0.660	0.360	0.300

Table A.1: The changes in the ratios of the positive (Pos) and negative (Neg) outcomes in each demographic subgroups with and without the COSCFair framework implemented with strategy 3 to predict the Adult test set using Gradient Boosting (GB) classifier.

Technique	RF w/o Mitigation			RF with COSCFair3		
Ratios	Base	Pos.	Neg.	Base	Pos.	Neg.
race: 0, sex: 0	0.497	0.218	0.280	0.497	0.241	0.256
race: 1, sex: 0	0.307	0.195	0.112	0.307	0.194	0.113
race: 0, sex: 1	0.104	0.068	0.036	0.104	0.060	0.044
race: 1, sex: 1	0.091	0.061	0.030	0.091	0.062	0.029

Table A.2: The changes in the ratios of the positive (Pos) and negative (Neg) outcomes in each demographic subgroups with and without the COSCFair framework implemented with strategy 3 to predict the COMPAS Recidivism test set using Random Forest (RF) classifier.

Technique	GB w/o Mitigation			GB with COSCFair3		
Ratios	Base	Pos.	Neg.	Base	Pos.	Neg.
race: 0, sex: 0	0.497	0.217	0.280	0.497	0.225	0.273
race: 1, sex: 0	0.307	0.220	0.087	0.307	0.201	0.106
race: 0, sex: 1	0.104	0.073	0.032	0.104	0.058	0.046
race: 1, sex: 1	0.091	0.075	0.016	0.091	0.064	0.027

Table A.3: The changes in the ratios of the positive (Pos) and negative (Neg) outcomes in each demographic subgroups with and without the COSCFair framework implemented with strategy 3 to predict the COMPAS Recidivism test set using Gradient Boosting (GB) classifier.

Technique	LR w/o Mitigation			LR with COSCFair3		
Ratios	Base	Pos.	Neg.	Base	Pos.	Neg.
age: 0, sex: 0	0.103	0.055	0.049	0.103	0.057	0.046
age: 1, sex: 0	0.207	0.161	0.045	0.207	0.130	0.077
age: 0, sex: 1	0.087	0.058	0.029	0.087	0.050	0.037
age: 1, sex: 1	0.603	0.507	0.097	0.603	0.399	0.204

Table A.4: The changes in the ratios of the positive (Pos) and negative (Neg) outcomes in each demographic subgroups with and without the COSCFair framework implemented with strategy 3 to predict the German test set using Logistic Regression (LR) classifier.

Technique	GB w/o Mitigation			GB with COSCFair3		
Ratios	Base	Pos.	Neg.	Base	Pos.	Neg.
age: 0, sex: 0	0.103	0.066	0.037	0.103	0.074	0.030
age: 1, sex: 0	0.207	0.157	0.050	0.207	0.149	0.058
age: 0, sex: 1	0.087	0.059	0.027	0.087	0.061	0.026
age: 1, sex: 1	0.603	0.486	0.117	0.603	0.442	0.161

Table A.5: The changes in the ratios of the positive (Pos) and negative (Neg) outcomes in each demographic subgroups with and without the COSCFair framework implemented with strategy 3 to predict the German test set using Gradient Boosting (GB) classifier.

A.2 Other Tables from the Experiment 2

<i>Subgroups (G vs G')</i>	AEO Diff.	DI Ratio	DP Diff.	PP Diff.	Consis.
A: 0, S: 0 vs A: 1, S: 0	-0.11	0.74	-0.17	-0.07	0.78
A: 1, S: 0 vs A: 0, S: 1	0.10	1.39	0.16	0.11	0.78
A: 0, S: 1 vs A: 1, S: 1	-0.06	0.79	-0.14	-0.15	0.78
A: 0, S: 0 vs A: 0, S: 1	-0.01	1.03	-0.01	0.04	0.78
A: 1, S: 0 vs A: 1, S: 1	0.04	1.03	0.02	-0.04	0.78
A: 0, S: 0 vs A: 1, S: 1	-0.08	0.77	-0.15	-0.11	0.78

Table A.6: Detailed results obtained from the first strategy of COCSFair with Logistic Regression classifier per unprivileged (G) and privileged (G') subgroup comparison. Sensitive attribute A represents *age* and S represents *sex*. The results are calculated using the German dataset. The performance measures of this result are on Table 5.3 (Logistic Regression classifier with COSCFair1).

Technique	LR w/o Mitigation			LR with COSCFair1		
	Base	Pos.	Neg.	Base	Pos.	Neg.
age: 0, sex: 0	0.103	0.055	0.049	0.103	0.050	0.053
age: 1, sex: 0	0.207	0.161	0.045	0.207	0.136	0.071
age: 0, sex: 1	0.087	0.058	0.029	0.087	0.043	0.043
age: 1, sex: 1	0.603	0.507	0.097	0.603	0.384	0.219

Table A.7: The changes in the ratios of the positive (Pos) and negative (Neg) outcomes in each demographic subgroups with and without the COSCFair framework implemented with strategy 1 to predict the German test set using Logistic Regression (LR) classifier.

<i>Subgroups (G vs G')</i>	AEO Diff.	DI Ratio	DP Diff.	PP Diff.	Consis.
A: 0, S: 0 vs A: 1, S: 0	0.00	0.92	-0.06	-0.15	0.81
A: 1, S: 0 vs A: 0, S: 1	0.03	1.11	0.07	0.09	0.81
A: 0, S: 1 vs A: 1, S: 1	-0.05	0.89	-0.09	-0.11	0.81
A: 0, S: 0 vs A: 0, S: 1	0.02	1.02	0.01	-0.06	0.81
A: 1, S: 0 vs A: 1, S: 1	-0.02	0.98	-0.02	-0.01	0.81
A: 0, S: 0 vs A: 1, S: 1	-0.02	0.90	-0.08	-0.17	0.81

Table A.8: Detailed results obtained from the first strategy of COCSFair with Random Forest classifier per unprivileged (G) and privileged (G') subgroup comparison. Sensitive attribute A represents *age* and S represents *sex*. The results are calculated using the German dataset. The performance measures of this result are on Table 5.3 (Random Forest classifier with COSCFair1).

Technique	RF w/o Mitigation			RF with COSCFair1		
Ratios	Base	Pos.	Neg.	Base	Pos.	Neg.
age: 0, sex: 0	0.103	0.071	0.032	0.103	0.075	0.029
age: 1, sex: 0	0.207	0.166	0.040	0.207	0.162	0.044
age: 0, sex: 1	0.087	0.066	0.021	0.087	0.062	0.025
age: 1, sex: 1	0.603	0.514	0.089	0.603	0.486	0.117

Table A.9: The changes in the ratios of the positive (Pos) and negative (Neg) outcomes in each demographic subgroups with and without the COSCFair framework implemented with strategy 1 to predict the German test set using Random Forest (RF) classifier.

Subgroups (G vs G')	AEO Diff.	DI Ratio	DP Diff.	PP Diff.	Consis.
A: 0, S: 0 vs A: 1, S: 0	-0.02	0.89	-0.08	-0.15	0.80
A: 1, S: 0 vs A: 0, S: 1	-0.03	1.06	0.03	0.13	0.80
A: 0, S: 1 vs A: 1, S: 1	0.04	0.96	-0.03	-0.16	0.80
A: 0, S: 0 vs A: 0, S: 1	-0.05	0.93	-0.06	-0.02	0.80
A: 1, S: 0 vs A: 1, S: 1	0.01	0.99	-0.01	-0.03	0.80
A: 0, S: 0 vs A: 1, S: 1	-0.01	0.88	-0.09	-0.18	0.80

Table A.10: Detailed results obtained from the first strategy of COCSFair with Gradient Boosting classifier per unprivileged (G) and privileged (G') subgroup comparison. Sensitive attribute A represents *age* and S represents *sex*. The results are calculated using the German dataset. The performance measures of this result are on Table 5.3 (Gradient Boosting classifier with COSCFair1).

Technique	GB w/o Mitigation			GB with COSCFair1		
Ratios	Base	Pos.	Neg.	Base	Pos.	Neg.
age: 0, sex: 0	0.103	0.066	0.037	0.103	0.065	0.038
age: 1, sex: 0	0.207	0.157	0.050	0.207	0.148	0.059
age: 0, sex: 1	0.087	0.059	0.027	0.087	0.060	0.027
age: 1, sex: 1	0.603	0.486	0.117	0.603	0.435	0.168

Table A.11: The changes in the ratios of the positive (Pos) and negative (Neg) outcomes in each demographic subgroups with and without the COSCFair framework implemented with strategy 1 to predict the German test set using Gradient Boosting (GB) classifier.

<i>Subgroups (G vs G')</i>	AEO Diff.	DI Ratio	DP Diff.	PP Diff.	Consis.
A: 0, S: 0 vs A: 1, S: 0	-0.02	0.87	-0.08	-0.12	0.75
A: 1, S: 0 vs A: 0, S: 1	0.00	1.11	0.05	0.12	0.75
A: 0, S: 1 vs A: 1, S: 1	-0.03	0.87	-0.09	-0.13	0.75
A: 0, S: 0 vs A: 0, S: 1	-0.02	0.96	-0.03	0.00	0.75
A: 1, S: 0 vs A: 1, S: 1	-0.03	0.95	-0.03	-0.01	0.75
A: 0, S: 0 vs A: 1, S: 1	-0.05	0.83	-0.11	-0.13	0.75

Table A.12: Detailed results obtained from the second strategy of COCSFair with Logistic Regression classifier per unprivileged (G) and privileged (G') subgroup comparison. Sensitive attribute A represents *age* and S represents *sex*. The results are calculated using the German dataset. The performance measures of this result are on Table 5.3 (Logistic Regression classifier with COSCFair2).

Technique	LR w/o Mitigation			LR with COSCFair2		
	Base	Positive	Negative	Base	Positive	Negative
age: 0, sex: 0	0.103	0.055	0.049	0.103	0.057	0.046
age: 1, sex: 0	0.207	0.161	0.045	0.207	0.131	0.075
age: 0, sex: 1	0.087	0.058	0.029	0.087	0.050	0.036
age: 1, sex: 1	0.603	0.507	0.097	0.603	0.403	0.200

Table A.13: The changes in the ratios of the positive (Pos) and negative (Neg) outcomes in each demographic subgroups with and without the COSCFair framework implemented with strategy 2 to predict the German test set using Logistic Regression (LR) classifier.

<i>Subgroups (G vs G')</i>	AEO Diff.	DI Ratio	DP Diff.	PP Diff.	Consis.
A: 0, S: 0 vs A: 1, S: 0	0.02	0.95	-0.04	-0.16	0.79
A: 1, S: 0 vs A: 0, S: 1	0.10	1.21	0.13	0.05	0.79
A: 0, S: 1 vs A: 1, S: 1	-0.10	0.84	-0.13	-0.07	0.79
A: 0, S: 0 vs A: 0, S: 1	0.12	1.14	0.09	-0.10	0.79
A: 1, S: 0 vs A: 1, S: 1	0.00	1.01	0.00	-0.02	0.79
A: 0, S: 0 vs A: 1, S: 1	0.02	0.95	-0.04	-0.18	0.79

Table A.14: Detailed results obtained from the second strategy COCSFair with Random Forest classifier per unprivileged (G) and privileged (G') subgroup comparison. Sensitive attribute A represents *age* and S represents *sex*. The results are calculated using the German dataset. The performance measures of this result are on Table 5.3 (Random Forest classifier with COSCFair2).

Technique	RF w/o Mitigation			RF with COSCFair2		
Ratios	Base	Pos.	Neg.	Base	Pos.	Neg.
age: 0, sex: 0	0.103	0.071	0.032	0.103	0.078	0.025
age: 1, sex: 0	0.207	0.166	0.040	0.207	0.165	0.042
age: 0, sex: 1	0.087	0.066	0.021	0.087	0.058	0.029
age: 1, sex: 1	0.603	0.514	0.089	0.603	0.478	0.125

Table A.15: The changes in the ratios of the positive (Pos) and negative (Neg) outcomes in each demographic subgroups with and without the COSCFair framework implemented with strategy 2 to predict the German test set using Random Forest (RF) classifier.

Subgroups (G vs G')	AEO Diff.	DI Ratio	DP Diff.	PP Diff.	Consis.
A: 0, S: 0 vs A: 1, S: 0	0.04	0.98	-0.01	-0.12	0.77
A: 1, S: 0 vs A: 0, S: 1	-0.02	1.05	0.03	0.11	0.77
A: 0, S: 1 vs A: 1, S: 1	0.01	0.94	-0.04	-0.13	0.77
A: 0, S: 0 vs A: 0, S: 1	0.02	1.03	0.01	-0.02	0.77
A: 1, S: 0 vs A: 1, S: 1	-0.01	0.98	-0.02	-0.03	0.77
A: 0, S: 0 vs A: 1, S: 1	0.03	0.96	-0.03	-0.15	0.77

Table A.16: Detailed results obtained from the second strategy of COCSFair with Gradient Boosting classifier per unprivileged (G) and privileged (G') subgroup comparison. Sensitive attribute A represents *age* and S represents *sex*. The results are calculated using the German dataset. The performance measures of this result are on Table 5.3 (Gradient Boosting classifier with COSCFair2).

Technique	GB w/o Mitigation			GB with COSCFair2		
Ratios	Base	Pos.	Neg.	Base	Pos.	Neg.
age: 0, sex: 0	0.103	0.066	0.037	0.103	0.072	0.031
age: 1, sex: 0	0.207	0.157	0.050	0.207	0.148	0.059
age: 0, sex: 1	0.087	0.059	0.027	0.087	0.060	0.027
age: 1, sex: 1	0.603	0.486	0.117	0.603	0.442	0.162

Table A.17: The changes in the ratios of the positive (Pos) and negative (Neg) outcomes in each demographic subgroups with and without the COSCFair framework implemented with strategy 2 to predict the German test set using Gradient Boosting (GB) classifier.