Master Thesis
Artificial Intelligence

Algorithmic Fairness: Which Algorithm Suits my Purpose?

Fenna Woudstra
5848865
f.woudstra@students.uu.nl

First supervisor
Dr. A.A.A. (Hakim) Qahtan

Second supervisor
Prof. Dr. I. (Yannis) Velegrakis

Utrecht University

Department of Information and Computing Sciences
Utrecht University
The Netherlands
August 23, 2022

Abstract

Machine learning (ML) algorithms are widely used in decision-making tasks. These decisions can have a big impact on the lives of people. Therefore, it is important that the outcomes of ML models are fair and do not lead to discrimination. Unfair outcomes could be a result of societal biases reflected in the assigned class labels, biases that arise during the data collection and processing, or the design choices made within an algorithm. Over the last decade, the topic of fairness in machine learning has become an important area of research that has led to many bias mitigation algorithms. These algorithms have shown to perform differently on different datasets. For this reason, data profiling can give a better understanding of the effectiveness of various bias mitigation algorithms. In this thesis, we analyzed sixteen bias mitigation algorithms and identified several characteristics of the data that help to decide which algorithm should be used for a given dataset to improve fairness. Based on that, we developed a Fair Algorithm Selection Tool (FairAST), that inspects the data and recommends the optimal algorithm to improve a given fairness measure. The experimental evaluation shows that, to a great extent, these recommendations are in line with the best performing algorithms found through exhaustive search.

Acknowledgements

I would like to express my gratitude to all the people who helped me complete my thesis project. First of all, I am very grateful for the guidance of my supervisor, Hakim Qahtan. I really appreciate how you took the time to meet regularly, and helped me with the questions I faced during the project. Next, I would like to thank my friends and family. In particular, my mother, my sisters, and their families. Your love and all that you have taught me, have brought me to where I am today. Finally, my greatest love and appreciation goes out to Derek. Thank you for your endless support, for keeping me motivated, for believing in me, and for always being there for me.

Contents

List of Abbreviations

Fairness principles	
EO	**E**qualized **O**dds
EoO	**E**quality **of** **O**pportunity
SP	**S**tatistical **P**arity
Measures	
BR	**B**ase **r**ate
C	**C**onsistency
DI	**D**isparate **I**mpact
FDR	**F**alse **D**iscovery **R**ate
FNR	**F**alse **N**egative **R**ate
FPR	**F**alse **P**ositive **R**ate
PPV	**P**ositive **P**redictive **V**alue
TNR	**T**rue **N**egative **R**ate
TPR	**T**rue **P**ositive **R**ate
Bias Mitigation Algorithms	
AD	**A**dversarial **D**ebiasing
CEOP	**C**alibrated **E**qualized **O**dds **P**ostprocessing
CR	**C**orrelation **R**emover
DIR	**D**isparate **I**mpact **R**emover
EGR	**E**xponentiated **G**radient **R**eduction
EOP	**E**qualized **O**dds **P**ostprocessing
FC	**F**airness **C**onstraints
GFC	**G**erry **F**air **C**lassifier
GSR	**G**rid **S**earch **R**eduction
LFR	**L**earning **F**air **R**epresentations
MFC	**M**eta **F**air **C**lassifier
OP	**O**ptimized **P**reprocessing
PR	**P**rejudice **R**emover
Rew	**Rew**eighing
ROC	**R**eject **O**ption **C**lassification
TO	**T**hreshold **O**ptimizer

Chapter 1

Introduction

Recent advances in Machine Learning (ML) techniques increase their use in many real life applications such recommendation systems, search engines, facial recognition, machine translation, and many more domains. Focusing on classification tasks, the predictions of ML algorithms are often used for making decisions that can severely affect people's lives. For example, when an algorithm predicts whether someone applying for a loan from a bank will be able to pay it back, this will affect the decision of whether or not the person will receive a loan. However, a lot of ML algorithms have been shown to be discriminatory against specific subgroups of the community because they are trained using biased datasets. For that reason, a set of bias mitigation algorithms has been developed to improve the algorithmic fairness. Yet, there is no clear indication which algorithm is most suitable for a given dataset or application.

ML models are supposed to make more rational and consistent decisions than humans, because they make decisions based on data, not on (unconscious) thoughts or beliefs. Unfortunately, the data that is used to train the ML models is not purely factual and neutral. In fact, this data carries the biases of the people who previously made decisions, or who were involved in the data collection process. At many points in the process of developing a ML algorithm, human biases can slip into the system, causing the algorithmic decisions to be just as biased and discriminatory as human decisions, or even worse [38].

There have been numerous examples showing that algorithmic decisions can create unfair situations. In 2014, Amazon created an algorithm that would screen job applicants' resumes and select the best candidates. Soon they found that the algorithm was biased in favor of men, denying all women the opportunity to get the job. This algorithm was trained on Amazon's history of job applicants, that contained more examples of men receiving the job than women, causing the algorithm to learn the pattern that men are more successful candidates [12]. Another well-known example is the tool COMPAS, used in the American justice system. This tool would rate defendants on their probability of re-offending. A study of ProPublica showed that this tool would give much higher risk scores to Black defendants than to White, reflecting racial biases in society [34].

We consider the above examples unfair because they discriminate people based on what are called 'sensitive attributes', like gender or race. Other common sensitive features are nationality, religion, age or sexual orientation. In many countries, there are even legal frameworks that forbid to discriminate people on specific personal characteristics [19].

It is clear from these examples that algorithms can have a severe impact on people's lives when they are used in critical decision-making tasks. That is why it is extremely important that the outcomes of an algorithm are fair, and do not discriminate people based on their personal characteristics. In order to prevent discriminatory situations, a lot of research has been going into the transparency, explainability and fairness of ML algorithms. Over the last decade many methods have been proposed that help people to investigate how an algorithm came to a decision and whether the decisions are fair [30, 31].

It is, however, a difficult research area, because the definition and quantifiability of a 'fair algorithm' remains ambiguous. Should the outcomes be equal for all different groups of people? Or should similar people receive similar opportunities? Different interpretations of what constitutes fair have led to numerous fairness measures like statistical parity, equality of opportunity and calibration [39]. Moreover, it has been shown that it is impossible to satisfy all fairness measures at once [29].

Based on these definitions of fairness, many bias mitigation algorithms have been proposed. These are techniques that aim to reduce the data bias by either transforming the data, modifying the classification algorithm, or changing the predictions from an algorithm in order to improve the fairness of the classifier with respect to the chosen fairness measure [23].

In this thesis, we are investigating the possibility to decide which bias mitigation technique to use, based on the characteristics of the dataset. In the following section, the motivation for this research will be further explained.

1.1 Motivation

Many bias mitigation techniques have already been proposed that would improve the fairness of a classifier. There are pre-, in-, and post-processing techniques, either adjusting the data, the algorithm or the predictions respectively. However, given a biased dataset, it remains unclear which bias mitigation algorithm will result in the most possible fair outcomes. There are several issues that complicate the decision of which algorithm to use.

First of all, the different proposed techniques aim to achieve different notions of fairness. We know from the impossibility theory that not all fairness measures can be fulfilled at once [29]. Therefore, it is important to analyze how bias mitigation algorithms perform on different measures, but also which measures are most suitable for a given dataset.

Secondly, bias mitigation techniques are often tested on differently pre-processed datasets, showing by the way the sensitive features are specified, or how the privileged and unprivileged groups are chosen. This makes it difficult to compare the performance of these methods, because it has been noticed that results are very susceptible for changes in the data and the way the data is pre-processed [16, 36].

Thirdly, all datasets have different characteristics, such as their size, the distribution of class labels, or the type of features. Given that the mitigation techniques need to process the data, the question is in what way data actually influences the results of the bias mitigation techniques. Are there specific characteristics of datasets that will lead to better results for one algorithm than for another?

In this study, we will address these issues by applying data profiling techniques to investigate the properties of the datasets being used. We will analyze the performance of a wide variety of bias mitigation algorithms on these datasets, for different fairness measures. By doing so, we aim to provide a framework that will help a user to choose which bias mitigation strategy can best be applied to their data.

1.2 Relevance for AI

The topic of fairness in ML is of great importance for AI research. We have seen several examples of biased ML models that were used in practice and caused severe problems. Decisions based on these models might unfairly deny people a job, a loan, or even a hospital admission. All these kinds of decisions make a great impact on someone's future life. Therefore, it is important to investigate how such harm can be prevented. It is the responsibility of AI and ML researchers to create models that are fair and do not harm people.

Besides, the examples of discriminatory algorithms are harmful for the field of AI as a whole. The negative consequences of biased models will scare people, or make people sceptical about AI. In short, if algorithms are not fair, people will (and should) no longer accept to use these techniques. The future of AI depends on the trustworthiness of the models, which is why research into fairness is an incredibly important part of AI research.

1.3 Research questions

The main purpose of this thesis is to provide a framework that will help a user to decide which bias mitigation technique to use for a given dataset. Therefore, the main research goal is to investigate how data profiling techniques can identify characteristics of the data that help to determine the optimal method to enhance fairness. This brings us to the following research questions, divided into a main question and several subquestions.

Research question: How can data profiling techniques contribute to selecting the optimal bias mitigation strategy for any given dataset?
Subquestions:

1. How does the choice of fairness measure influence the choice of bias mitigation algorithm?
2. How do different bias mitigation algorithms perform compared to each other?
3. To what extent are the bias mitigation algorithms robust enough to ensure fairness for any given dataset?
4. Which statistics obtained by data profiling are important for the selection of a bias mitigation technique?

1.4 Definitions

Below, several important terms and notations are explained that will often be used in the following chapters.

- Bias = a prejudice or favoritism toward a group [31].
- Fairness = the absence of prejudice or favoritism toward a group based on personal characteristics [31].
- Group membership = whether a person belongs to the privileged or unprivileged group, based on the protected attributes.

- Favorable label = the desirable outcome of a decision, receiving this label will benefit a person [23].
- Privileged group = the group that is put at an advantage by receiving the favorable outcome more often than others (the unprivileged group) [23].
- Protected attributes/sensitive features = features that are not allowed to discriminate on, often personal characteristics like gender, ethnicity, or age [9].
- G = Group membership based on the protected attributes, where $G = 1$ is the privileged group, and $G = 0$ is the unprivileged group.
- S = Set of sensitive features/protected attributes, where for every $s \in S, s = 1/0$ denotes the privileged/unprivileged groups according to s.
- X = The non-sensitive feature that are used for classification.
- Y = Actual label/outcome, according to the data. Where $Y = 1$ is the favorable label, and $Y = 0$ the unfavorable label.
- $\hat{Y}$ = Predicted label by the classifier.

1.5 Outline

The subsequent chapters of this thesis are structured as follows. Chapter 2 discusses the related work. This chapter starts by investigating the definition of bias, then it discusses different fairness measures, the impossibility theory, and different techniques for bias mitigation. Chapter 3 covers the theoretical background, in which the problem statement and the goals of this research are formulated. Furthermore, it explains how data profiling techniques can be used for fairness research purposes, and it presents the analysis of the bias mitigation algorithms. Chapter 4 introduces FairAST, the proposed tool that recommends the optimal bias mitigation algorithm to use on a given dataset. All different parts of the tool are explained in this chapter. The performance of FairAST is evaluated by performing several experiments. The set-up of these experiments as well as the results are discussed in Chapter 5. Finally, in Chapter 6, the conclusion, limitations, and directions for future work are discussed.

Chapter 2

Related Work

In the following sections, the related work will be discussed. It starts with section 2.1 about the definition of bias and how biases arise. In section 2.2, different ways of formulating and measuring fairness are explained. Section 2.3 discusses how some fairness measures are incompatible with each other. This chapter ends with section 2.4 discussing different methods of mitigating bias in machine learning algorithms.

2.1 Bias

In a machine learning context, an algorithm is biased when its decisions lead to unfair treatment or discrimination of certain individuals or groups [17]. Note that discrimination here is meant as unfair discrimination. All machine learning tasks 'discriminate' in the sense that they make a division of who will receive the favorable and the unfavorable outcome. However, this decision should be made on reasonable grounds. Personal characteristics like race, gender, age or nationality should not be discriminated on. In many countries it is even forbidden by law to discriminate people based on their personal characteristics. Unfair discrimination is thus about treating people differently based on unfounded reasons, like prejudices or stereotypes [17].

This does not mean that features like gender or age are never allowed to be used in a machine learning task. In some situations these features are actually useful and do not lead to unfair discrimination. For example in health care, men and women may need different treatments due to their biological differences. Treating men and women in the exact same way for a certain disease might be harmful, while treating them differently can benefit them both. Different decisions for these groups can thus be explained, which is why resulting disparities are considered 'explainable discrimination' [26].

Unfair discrimination on the other hand always harms the unprivileged group. The harms caused by biased algorithms can be divided into representational and allocative harms [5]. Representational harms appear in situations where stereotypes or social disparities are reinforced by the algorithm. For example, when an image search for 'CEO' shows more male than female CEO's, the disparities in the real world are reflected in the output. This can be harmful, since it can reinforce stereotypes about men and women. In situations where a biased algorithm directly withholds people a resource or an opportunity, it causes allocative harms. This is for example the case when an algorithm is used in hiring systems, deciding who will get a job. A biased system will systematically favor one group over the other, denying the unprivileged group the opportunity of getting a job.

Biases can exist or arise at different places in the machine learning pipeline [31]. First of all, data that is used for training the algorithm is based on the real world.

Therefore, any disparities in the world will be reflected in the data. These pre-existing biases are also called historical or societal biases, and they arise from social institutions, practices or attitudes and prejudices [17]. To illustrate, when people from a bank hold prejudices about a specific group and therefore (perhaps unconsciously) deny those people more often a credit loan, the data will show that some groups more often get a loan than this unprivileged group. The 'ground truth' labels are thus biased. Algorithms will learn this pattern and copy the discriminatory behavior of people in the real world.

Secondly, the way the data is collected or handled during the development of an algorithm can also lead to biases. An example is the representational bias, when the data sample is not a good representation of either the real world or the group the algorithm will be used for. Another example is the measurement bias, that arises by the way features are chosen to measure a particular aspect. For example, when arrest rates are used to measure crime rates, the measured number of arrests are used as a proxy to get an indication of the amount of crime, but perhaps not all crime is reported or has led to arrests [38].

Thirdly, biases can be the result of the algorithm that is being used. The inner workings of an algorithm or the design choices made by people influence the algorithm's behaviour, and may lead to biased outcomes [31]. Furthermore, emergent biases can arise when the user interacts with the algorithm [17]. Another interesting result is that the outcomes of an algorithm influence what happens in the world, which affects the data that later on will be collected. This process is called a feedback loop [31].

2.2 Fairness measures

Fairness is a concept that is hard to precisely define. People often have intuitions about whether something is fair, but these can differ per situation or per person. A core aspect of all notions of fairness is however a demand for equality, which means that biases and discrimination are deemed unfair [31]. Still, such a definition does not solve the problem for ML researchers on how to measure and quantify whether an algorithm is fair. Different fairness measures have been proposed that try to grasp the definition of fairness. Amongst the most well-known principles are group-based measures like statistical parity [15], equalized odds [21], and calibration [33], and notions of individual fairness [14].

Perhaps a counter-intuitive finding is that not all notions of fairness can be fulfilled at the same time. This will be further discussed in section 2.3. However, these results show that it matters which measure is adopted in evaluating the fairness of an algorithm. According to [21] it is therefore important to consider the underlying moral assumptions of each fairness measure, in order to decide which measure is most appropriate in a given situation.

In the following subsections we will discuss the differences between some of the most well-known fairness measures and investigate their underlying assumptions, advantages and limitations.

2.2.1 Parity-based measures

Parity-based measures define fairness as equality in the outcomes for both groups in the data. According to the parity-based measures of fairness, the predictions made by the classifier should thus be derived independently from the group membership, which is denoted by $\perp$ [9]:

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

For **statistical parity (SP)**, also called **demographic parity**, this is reflected in the idea that the percentage of favorable outcomes (the **base rate**) should be equal for all groups [9]:

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

Statistical parity is measured by calculating the **Disparate Impact (DI) ratio** or **difference**. The DI ratio can be calculated by dividing the base rate of the unprivileged group over the privileged group. Perfect equality would give a ratio of 1, but in American law, the 80%-rule has been adopted, meaning that the ratio should be above 0.8 in order to consider the data fair [15]. For the DI difference, the base rate from the privileged group is subtracted from the unprivileged group. Perfect equality would then give a difference of 0.

The assumption made here is that the groups to be compared are or should be equal, and therefore should have the same probability of getting a favorable outcome. Disparities between the base rates are therefore considered signs of unfairness. According to a study where participants were asked to assess the predictions made by a classifier, this fairness measures appeals most to people's perception of fairness. The given cases were about fairness in predicting recidivism and detecting skin cancer [37].

However, it is important to note that when SP is not met, it does not mean that one group is undoubtedly being discriminated. As mentioned before, sometimes there are differences between the groups that can explain the different rates of positive outcomes, that are not taken into account by measuring the disparate impact [6]. In [26], an example for the Adult dataset is given, which is used for predicting whether the annual income of people is below or above 50k. Women are more often predicted to receive a low income than men, meaning that SP is not satisfied. This could be interpreted as unfair, but actually the data also shows that women have a lower number of working hours. So in this case, the different levels of income for the two groups can be explained by an actual difference in the features of these groups. This example shows that the assumption that two groups are or should be equal is not true for all situations. Therefore, these measures do not always provide correct insight in the fairness of a classifier. As suggested by [6], SP is a suitable measure in cases of 'civil justice', where personal characteristics should not matter for the outcome.

2.2.2 Confusion matrix-based measures

The following measures are based on the confusion matrix of a classifier, which specifies the number of correct and incorrect (un)favorable predictions. These measures thus consider the correctness of the predictions made by the classifier compared to the actual labels as specified in the data. They adhere to the criterion of 'separation', which states that the prediction should be independent from the group membership conditioned on the actual labels [9].

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

In practice, this means that based on the confusion matrix specific rates are calculated, such as the True Positives Rate (TPR) or the False Negatives Rate (FNR). According to the following fairness measures, these rates should be similar for different groups. For **Equality of Opportunity (EoO)**, the TPR for both groups should be equal, which means that the same percentage of people with an actual favorable label were also predicted to receive the favorable label. This measure is in classification tasks also known as 'recall', but here the recall is calculated for the privileged and the unprivileged group, instead of over the complete dataset:

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

For **Equalized Odds (EO)**, both the TPR and the FPR should be equal across groups. This is similar to equality of opportunity, but now also the percentage of false positive predictions should be similar:

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

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

The **Positive Predictive Value (PPV)** measures how many of the positive predictions were actually correct. For fairness purposes, it can be investigated whether this score, also known as 'precision', is equal for the privileged and the unprivileged group:

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

The complement of the PPV is the **False Discovery Rate (FDR)**, that measures how many of the positive predictions were false. This rate can also be compared for the privileged and unprivileged group:

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

Treatment Equality measures the ratio of false positive predictions over false negative predictions for both groups, and compares them. This measure can help to reveal whether the misclassifications of a group are more often advantageous or disadvantageous for the group, and whether this is similar for both groups [9]:

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

Equal Accuracy measures the percentage of correct predictions in order to compare the accuracy for both groups [9].

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

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

Comparing the accuracy can be useful to initially investigate whether a classifier is able to make correct predictions for perhaps a minority group in the data. The overall accuracy might be very high, while for some groups the predictions can be very wrong. However, improving fairness comes at cost of the accuracy of a classifier, since it is actually assumed that the 'correct' classifications are not fair and should be changed [23]. Therefore, equally high accuracy is not necessarily a fairness measure to strive for, because a decreased accuracy might actually improve the fairness.

An advantage of these measures is that they do take into account the possibility that groups have different qualities, where parity-based measures did not. This is often considered fair in situations of ’economic justice’ [6]. On the other hand, an important assumption of these measures is that the actual label is present, correct and fair. Unfortunately, this can be problematic in real situations [21]. Imagine a situation where an algorithm is used for hiring people, and the training data exists of candidates from the past, and the labels show who was hired or not. In this case, the actual labels reflect what happened in practice, but the labels are not necessarily correct or fair. The hiring procedure could have been performed by people with biases against some groups, and we will never know whether the people that were not hired would have been a great choice for the job too.

2.2.3 Calibration-based measures

The third group of fairness measures is based on the criterion of ’sufficiency’. This criterion is met when the actual labels are independent from the group membership, conditioned on the predicted outcome. That is, people with the same predicted label should have similar probabilities of having the actual positive label [9].

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

According to **calibration** or **test fairness**, people with a similar probability score p at receiving the favorable prediction should also have the same probability of actually belonging to the favorable label. Similar to the confusion matrix-based measures, it is assumed that the actual label is present and correct.

$$P(Y = 1 | p \& G = 0) = P(Y = 1 | p \& G = 1)$$

2.2.4 Individual fairness measures

The three types of fairness measures discussed so far, all compared group statistics against each other. Arguably, these measures fail to treat individuals as individuals, since equal rates on a group level, do not ensure fair decisions for an individual [6]. Quite intuitively, we would want similar individuals to be treated similarly, so there is **consistency**. This means that individual outcomes are considered, and that similar individuals should get the same predicted label [14]. The consistency measure requires a specific metric for similarity between individuals, which can be hard to define [23]. The consistency measure from [41] is based on a K-Nearest Neighbors method applied over the entire dataset. Importantly, this means that even when individuals are being treated similarly, this does not necessarily imply they are being treated fairly. All ’nearest neighbours’ of the unprivileged group could be other unprivileged persons, all receiving the unfavorable outcome. Such outcomes would show high consistency, although they are still biased.

Another way of assessing individual fairness is by **counterfactual fairness** [9]. This method is based on a causal model, and it measures whether the prediction for a specific individual would have been the same if only the sensitive attribute would have changed.

2.2.5 Summary

In short, we have seen that not all fairness measures are suitable for any given situation. A decision needs to be made which measure is most appropriate given the bias in the data and the goal of the classification task. Statistical parity is useful for measuring biases in the label distribution, but it overlooks the possibility of explainable disparities. Other group statistics like equality of opportunity and equalized odds do allow for such explainable disparities, but these measures rely heavily on the assumption that the labels are fair. Beside group statistics, it is useful to look at individual fairness, for example to measure the consistency of the decisions being made.

2.3 Impossibility theory

Now that we have seen several ways of measuring fairness, it is interesting to investigate how these measures work together. As mentioned before, some measures are mathematically impossible to satisfy simultaneously, or only in special cases with strict assumptions.

For example, it has been proven that calibration is incompatible with EO. In [29], the authors consider calibration within groups, and a balance for the positive and negative class. These last two are in binary classification tasks similar to an equal FNR and FPR. Since FNR is the complement of TPR (because FNR = 1 - TPR), these balances for the positive and negative class together are the same as EO as explained in section 2.2.2. They show that satisfying these fairness measures would only be possible in two highly constrained cases. The first one being that the base rates of the privileged and unprivileged group are the same. This usually is not the case, since the the privileged group more often is predicted to have the positive outcome than the unprivileged group. Or, the classifier should make perfect predictions, causing the FNR and FPR of both groups to be 0, and the calibration to be satisfied, because the predicted scores exactly match the actual scores.

Following on from these results, the compatibility of calibration together with only one of the error rate constraints (FNR or FPR) is investigated in [33]. They find that only sometimes these two constraints can be satisfied simultaneously. However, the authors suggest that it is better to choose between calibration or equal error rates, as their experiments show that imposing both restrictions increases the disparity for all other error rates. In most practical cases, this will cause the overall decisions of the algorithm to be considered unfair.

Calibration can be satisfied together with SP, or with PPV, but not necessarily. Since calibration is based on the scores produced by the classifier, while SP and PPV are based on the classifications, it depends on the chosen threshold that transforms the scores into classifications whether both calibration and SP or PPV are satisfied [18].

Continuing with the PPV, it has been shown that this measure is incompatible with EO [11]. Again, these two measures could only be satisfied simultaneously if the base rates are equal. However, only fulfilling one of EO constraints, like equal TPR, is possible in combination with equal PPV across groups [18].

Finally, there are some remarks on the utility of satisfying two fairness criteria when the fulfillment influences other measures. According to [18], the TPR of one of the groups can be forced to be very low when both SP and PPV are satisfied. This would be the case when the ratio of the base rates is far from 1, meaning that the base rates are very different. This would be undesirable, since it would mean that for one group there are barely correctly positive classifications.

Furthermore, the authors mention that while SP and EO can be simultaneously fulfilled, this would only be possible when the TPR and FPR are equal to each other. This is undesirable, because the TPR is preferably higher than the FPR.

However, these observations are based on mathematical derivations and assumptions of perfectly satisfying a measure. In practice, there is more leniency in terms of fulfilling a fairness measure. As mentioned, a score above 0.8 is for SP already considered fair, and for other measures, a score ’close to’ 0 or 1 is often interpreted as fair. Nevertheless, it is important to know that there are always trade-offs being made when one optimizes for one or two fairness measures. Depending on the case, one has to decide to what extent these are acceptable and fair.

2.4 Bias mitigation algorithms

An initial thought in preventing unfairness might be to exclude sensitive features from the data, assuming that this will prevent the algorithm from learning biases against specific groups. Unfortunately, this idea of ’fairness through unawareness’ does not guarantee fair results. There can be correlated features in the data that work as proxies for the left out sensitive features. Zip codes, for example, can tell something about ethnicity. A classifier will thus still learn the same patterns, but now it is more difficult to see whether the biases are against a specific group of people. Keeping track of sensitive features can help to check for those biases, and find ways to mitigate them. In these cases the sensitive features are present but marked as ’protected’ in order to prevent discrimination based on these features.

In general, all these methods can be divided in three groups. The difference lies in where the alterations are made to create more fair outcomes. For the pre-processing techniques, the input data will be modified. This is based on the idea that a biased input will lead to biased outcomes, therefore the input has to be fair [16]. The second group is called in-processing techniques, meaning that the algorithm itself is discrimination-aware and handles the data in such a way that the outcomes are fair. The final group are post-processing techniques, that change the predictions made by a classifier in order to satisfy a particular fairness criterion [31]. In the following sections, examples of each of these techniques will be discussed.

2.4.1 Pre-processing

Disparate Impact Remover (DIR) This mitigation algorithm aims to achieve SP, so to remove the disparate impact. It does so by changing the feature values, in order to create more equality between the privileged and the unprivileged group. The in-group order is preserved, which means that for every single feature the ranking of the instances in the data will remain the same although the feature values are changed [15].

Learning Fair Representations (LFR) This method aims at achieving group fairness (SP) and individual fairness. It does so by mapping each individual, represented as a data point in a given input space, to a probability distribution in a new representation space. The goal is to lose any information that can identify whether the person belongs to the protected subgroup, without losing too much other information [41].

Optimized Pre-processing (OP) This method transforms the data (features and labels) in order to diminish the disparate impact. It does so by creating a mapping from the dataset to a transformed dataset. The transformation is constrained by three principles: 1) the outcome should be independent from the sensitive attribute (to satisfy SP), 2) distortion control must prevent the transformation to make very large changes in the data, and 3) utility preservation means that the overall distribution of the original and the transformed dataset must be similar [8].

Reweighing (Rew) This algorithm also aims to diminish disparate impact difference. The pre-processing method assigns different weights to different objects in the data, in order to compensate for biases. So the unprivileged group will get higher weights for the positive examples, and lower weights for the negative examples, while the privileged group will get lower weights for the positive examples and higher weights for the negative examples. The weights are obtained by first calculating the expected (or desired) probability that an instance from one group obtains the favorable class label. This probability is divided by the observed probability that an instance from that group gets the positive class label [24].

Correlation Remover (CR) The CorrelationRemover comes from Fairlearn's preprocessing package. This method transforms the data in order to remove the correlation between the sensitive and non-sensitive features. The user is given the hyperparameter $0 \leq \alpha \leq 1$ to specify how much filtering should be applied, as to what extent the correlations are removed. The column containing the sensitive feature will be dropped [7].

Capuchin The method called Capuchin repairs the data by removing the causal relationship between the sensitive features and the labels. The method aims at achieving justifiable fairness, which means that the decision is only influenced by 'admissable' features, and not by the 'inadmissable' (sensitive) features, [35]. The user specifies which features are admissable or inadmissable.

2.4.2 In-processing

Adversarial Debiasing (AD) In Adversarial Debiasing, two neural networks are used to mitigate the bias: one, the predictor, is trained in predicting the correct label based on the features, while the other, the discriminator, tries to predict the sensitive feature based on the predictions from the first network. The goal of the predictor is to make it difficult for the discriminator to identify the sensitive feature, because this shows that there is no longer a relation between the sensitive feature and the outcome [42].

Gerry Fair Classifier (GFC) The Gerry Fair Classifier [28] aims at achieving fairness for subgroups within the data. Subgroups are based on all the possible value combinations of the sensitive attributes. The method is based on 'fictitious play' game theory, where two opponents try to achieve their goal while playing against each other. Here, the game is to solve a cost sensitive classification problem, while the Learner aims to maximize accuracy and the Auditor aims to improve the fairness for the subgroups.

Meta Fair Classifier (MFC) This method is described as a flexible meta-algorithm, that is able to optimize for different fairness measures and can handle complex compositions of sensitive features [10]. The algorithm learns through gradient descent the optimal regularization parameter that constraints the classifier to make predictions that are fair with respect to the specified fairness measure.

Prejudice Remover (PR) The Prejudice Remover aims to remove the 'prejudice' in the data, which is defined as the statistical dependence of the sensitive features on the labels. A regularization term is added to the logistic regression classifier that reduces the influence of the sensitive features on the final prediction [27].

Exponentiated Gradient Reduction (EGR) This method transforms the problem to a cost-sensitive classification problem in order to optimize a classifier for the specified fairness constraints [2]. Similar as explained for GFC, the method involves two 'players', one optimizing for accuracy and one for fairness. An exponentiated gradient algorithm is used to find the Lagrange multipliers that result in the most fair and accurate outcomes.

Grid Search Reduction (GSR) This approach is similar to the previous one, except that it performs a grid search to find the optimal Lagrange multipliers in order to satisfy the specified fairness constraints, instead of using the exponentiated gradient algorithm [2].

Fairness Constraints (FC) The method as proposed in [40] introduces a notion of decision boundary (un)fairness that helps the algorithm for one or more sensitive attributes to ensure SP. The decision boundary (un)fairness is based on the covariance of the sensitive attributes and the distance to the decision boundary. The constraint classifier aims to reduce this covariance to zero, to satisfy SP.

2.4.3 Post-processing

Calibrated Equalized Odds Post-processing (CEOP) The Calibrated Equalized Odds technique optimizes for EO, but it also aims to preserve calibration. Since these two fairness measures are incompatible, the notion of EO is relaxed to a form where it is optimized for either the FPR, the FNR, or a weighted combination of these two rates [33]. The post-processing consists of perturbing the prediction scores from a randomly chosen subgroup in the data, in order for the scores to satisfy the fairness constraints.

Equalized Odds Post-processing (EOP) The Equalized Odds method requires to know which instances in the data belong to the privileged and the unprivileged group. As it is a post-processing technique, it uses the predictions made from a regular classifier to find the initial true and false positive classifications for the privileged and unprivileged groups. Then, it will learn the probabilities by which to randomly change the original predictions in order to satisfy EO [20].

Reject Option Classification (ROC) The Reject Option Classification changes the prediction based on the confidence band around the decision boundary. For the privileged group this means that the most uncertain positive predictions will be changed to a negative one, and for the unprivileged group the most uncertain negative predictions will be changed to positive outcomes [25].

Threshold Optimizer (TO) Fairlearn's ThresholdOptimizer is based on the EOP algorithm from [20]. This method aims to find the optimal thresholds for each group to predict the positive class, in order to meet the specified fairness and accuracy constraints. It is possible to optimize for SP or EO, and additionally optimize for different accuracy constraints [7].

Chapter 3

Theoretical Background

In this chapter the theoretical background is presented. It starts with a formulation of the problem statement in section 3.1, and the goals in section 3.2. Next, the purpose and methods of data profiling are discussed in section 3.3. Finally, in section 3.4 we provide an analysis of different bias mitigation algorithms in order to find out what kind of characteristics of the data tell us which algorithm can best be used.

3.1 Problem statement

In this study, a binary classification problem is considered for a given dataset $D = \{X, S, Y\}$, where X is the set of the non-sensitive features, S the set of sensitive features, and $Y = \{0, 1\}$ is the label. The sensitive features in S are combined into a single feature $s = \{0, 1\}$, where 1 denotes the privileged group.

We consider a set of bias mitigation algorithms $A = \{A_1, A_2, .., A_n\}$, and fairness measures $F = \{F_1, F_2, .., F_m\}$. The optimal result of a fairness measure is denoted by

$$opt(F_j), \text{ where } opt(F_j) = \begin{cases} 0, & \text{if } F_j \text{ measures a difference} \\ 1, & \text{if } F_j \text{ otherwise} \end{cases} .$$

The goal is to use data profiling techniques to find A_i such that

$$E = \arg\min_{A_i \in A}(|opt(F_j) - F_j(A_i(D))|)$$

where $F_j(A_i(D))$ stands for the fairness after applying A_i to D, measured by F_j.

3.2 Goals

The goal is to provide a framework that will help the user to:

1. perform a systematic data analysis
2. discover possible biases in the data
3. decide based on the data profiling which bias mitigation algorithm can best be used
4. create more fair outcomes

3.3 Data profiling

Data profiling is about investigating the properties of a dataset to get a better understanding of the data and its metadata. It should not be confused with creating profiles of people based on their data, which is also referred to as (personal) data profiling. Data profiling as it is meant here, is a systematic analysis of data, which will give insight into the structure, content and the quality of the data [4].

As mentioned in [32], data profiling serves multiple purposes. First of all, it can be the starting point in the data cleaning process. By creating the metadata of the columns, like information about missing values, or 'rules' of the data type, all kinds of information can be extracted. Based on these findings the data can be cleaned. Secondly, it can optimize the process of creating queries, because it is already clear how the data can be queried. Thirdly, it is important for data integration to know what the sources consist of, where they overlap or differ. Fourth, it is useful for (scientific) data management. And finally, it is useful for data analytical purposes, like data mining.

There are numerous methods, techniques and tools that can be used in the process of data profiling. The authors of [1] give an overview of the possible things to consider during data profiling. First, one can look at single columns and generate metadata for each of them. This metadata can consist of counts (number of values, null values, unique values), statistics (mean value, minimum and maximum), the data type or patterns in the data values. Second, one can look at multiple columns to discover correlations and dependencies. Third, if there are multiple sources, data profiling can be used to discover overlap and dependencies between columns of different tables.

In this thesis, data profiling will first be used for the data cleaning and pre-processing. Considering that the outcomes of a classifier are the result of the input data, it is useful to carefully inspect the data. For the data cleaning, the data profiling will ensure the identification of missing values, and the type of features in the data. For the data pre-processing, data profiling can help to identify which features are possible sensitive features and whether these features show disparities for different groups, by calculating statistics of the data. This will help to identify the possible biases and other characteristics of the dataset.

Furthermore, data profiling will be used to select a bias mitigation algorithm. Obtaining different statistics and characteristics of the data can help to see if a dataset is suitable for a specific bias mitigation algorithm. In the following section we will investigate what kind of statistics are important to obtain for this purpose, by analyzing a wide variety of bias mitigation techniques.

3.4 Analysis of bias mitigation algorithms

In order to find out what kind of characteristics of a dataset are important for selecting a bias mitigation algorithm, we need to know how the algorithms work. The inner workings of an algorithm will show whether the input data should meet certain requirements for the algorithm to perform well. Therefore, we will compose a diverse set of bias mitigation algorithms and analyze their operations. We selected sixteen algorithms, all of which the method of bias mitigation are already discussed in section 2.4. Only the method Capuchin was not included, because the code of this algorithm is not publicly available.

Whenever possible, we used the implementation of the algorithm from existing packages AIF360 and FairLearn. These are well-known Python libraries for fairness that entail multiple bias mitigation algorithms, but also several implementations to easily obtain different fairness measures on a dataset. We preferred to use these implementations over the original algorithms, because of their ease of use, their available documentation, and examples of working with the algorithms. The only drawback is that the implementation of a specific algorithm can differ from the original implementation discussed in the literature. Whenever needed, we will point out these differences. Only the method FC is not from either of these two packages.

For each algorithm, we inspected the code obtained from GitHub[1], and went through the different parts of each algorithm to understand how it works and how the data is handled. Here we will discuss which requirements are found to be of importance, and other remarks about the algorithms.

First we determine the type of classification the models are made for. Most of the algorithms we consider are made for binary classification tasks, therefore the label is only allowed to take two different values. For the AIF360 algorithms, the data even needs to be in a BinaryLabelDataset format, which is a data structure that stores the data as well as some characteristics of the data, including the label, the features and the protected attributes. As the name shows, this can only be used for datasets with binary labels. Only Fairlearn's EGR and GSR can also be used for regression problems, for which the label stores continuous values.

For the LFR algorithm, there is another requirement for the label. This algorithm creates prototypes of instances in the data, and assigns all instances to one of these prototypes, as if it clusters the data. The algorithm aims to satisfy three criteria: 1) the features of the prototypes should be of good quality, meaning that they resemble the original data well, 2) the privileged and unprivileged instances should be equally distributed over the prototypes, causing them to receive similar predictions, and 3) the predictions should be as accurate as possible [41]. These three criteria result in three losses that are taken together in the total loss formula that is to be minimized. Despite the possibility to put different weights to each of these losses, the algorithm might show the tendency to provide the trivial solution: predicting only one class label. This is because there is no guarantee that the prototypes resemble both favorable and unfavorable predictions. This is especially problematic for datasets with a great class imbalance, which means that there are much more favorable or unfavorable labels in the dataset. In this case, the loss of the algorithm can easily be minimized by predicting one class, since then the accuracy is still relatively high, the predictions are exactly equal for both groups, and it is still possible to create prototypes that closely resemble the data, just not the instances in the data that are related to the minority class. For more balanced datasets, the prototypes are more likely to resemble both favorable and unfavorable instances.

Regarding the sensitive features, there are a couple of differences between the bias mitigation algorithms. First of all, the allowed number of protected attributes is limited to one for some algorithms. This is clearly specified for DIR, MFC and PR, but for LFR and AD it appeared to be the case that only the first attribute from the list of protected attributes is actually considered to be the attribute on which the bias should be mitigated.

Secondly, for some algorithms the protected attributes should contain binary values only. This is the case for MFC, where the code shows that for the protected

[1] AIF360: https://github.com/Trusted-AI/AIF360,
Fairlearn: https://github.com/fairlearn/fairlearn,
FC: https://github.com/mbilalzafar/fair-classification

attribute the values for the privileged group are changed to 1, and the remaining values to 0. This is different from the original MFC implementation, that, according to the paper [10], should be able to handle multiple and non-binary protected attributes.

Another aspect is the number of privileged and unprivileged groups that are specified. A group is specified by a dictionary stating the protected attribute and the corresponding value for that group. For example: {'race':0, 'sex':1}. In case there are, for instance, two binary protected attributes, there will be four different groups in the data, showing all possible combinations of these two attributes. When two of these groups are privileged, it is possible to list both of these groups as privileged. Some algorithms, however, allow only a single privileged and unprivileged group to be specified. This means that there can either be only a single binary protected attribute, resulting in only two groups, or some groups are considered to be neither privileged or unprivileged. Both LFR and AD only allow a single privileged and unprivileged group to be specified.

By contrast, the GFC thrives by a larger set of protected attributes, possibly with multiple values (as long as they are categorical). This algorithm is made for achieving fairness across a large number of subgroups defined over the protected attributes. A subgroup consists of a combination of multiple protected attributes, such as the four groups mentioned earlier that arise when two binary protected attributes are specified. If there would be only a single binary attribute, there are only two groups, which is not in line with the idea of subgroups. Experiments will need to show whether this is problematic for the algorithm, or whether it is still able to achieve good fairness and performance results.

There can also be restrictions on the number of features, or unique values per feature. The OP algorithm aims to transform the data to reduce the disparate impact. However, in the process of doing so, dataframes are created that represent the total number of possible feature combinations in the data. For this purpose, the cartesian product over all features and unique feature values is taken. For datasets with a high number of unique feature values, this will result in an enormous dataframe, that is too big to be stored. Therefore, this algorithm is only suitable to use on datasets with a low number of features and unique feature values.

For LFR, there also is a restriction on the number of unique feature values for the categorical features. Since this algorithm requires categorical features to be one-hot-encoded, features with many different categories will create very large and sparse matrices. This complicates the calculations of the distance between different instances, and is therefore not recommended.

Finally, the data type of the original features is important. The DIR technique is built for transforming numerical features in the data, by creating a more equal distribution of the values in the data for the privileged and unprivileged group. It is meant for numerical features, because there is supposed to be a ranking in the data. For categorical features that are not ordinal, this is not the case. Thus, in order to apply DIR, there must be numerical features in the data, not only categorical features. By contrast, the OP works best for categorical features. As mentioned before, the features should have a low number of unique feature values, which often is not the case for continuous data. Moreover, the algorithm assigns costs to different values, to create a cost-sensitive classifier. Therefore, the numerical features should be categorized to assign these costs.

To summarize, the important characteristics to identify in a dataset are the type of label, the label balance in the data, the number of protected attributes, the type of protected attributes (binary or multi-valued), the number of privileged and unprivileged groups, the number of unique feature values in combination with the number of features, and the data type of the features. An overview of the algorithms and their requirements is presented in Table 3.1.

Besides these characteristics that can be obtained from the data, it is important to note that the algorithms aim at achieving different notions of fairness. It depends on the fairness measure the user deems right for the dataset to enhance which algorithms could be suitable. Furthermore, the algorithms are examples of pre-, in-, and post-processing techniques. If a user wishes to only use a specific type of technique, this should be considered too. Looking at the differences between the type of techniques, it is noticeable that the post-processing techniques impose less restrictions on the data than the pre- and in-processing techniques. This can be explained by the fact that the post-processing techniques perform their operations based on the actual and the predicted labels, and a representation of the privileged and unprivileged groups. The features of the data are not as important to these algorithms as for the pre- and in-processing techniques.

Method	Package	Type	Measure	Protected	Other
DIR[15]	AIF360	Pre	SP	Single	Numerical features
LFR[41]	AIF360	Pre	SP	Single	Single groups, balanced, small nr features/values
OP[8]	AIF360	Pre	SP		Categorical features, small nr of features/ values
Rew[24]	AIF360	Pre	SP		
CR[7]	Fairlearn	Pre	SP		
AD[42]	AIF360	In	EO	Single	Single groups
GFC[28]	AIF360	In	FPR/FNR	Multi	
MFC[10]	AIF360	In	SP/FDR	Single, binary	
PR[27]	AIF360	In	SP	Single	
EGR[2]	Fairlearn	In	SP/EO/ TPR/FPR/ TNR/FNR		
GSR[2]	Fairlearn	In	SP/EO/ TPR/FPR/ TNR/FNR		
FC[40]	-	In	SP		
CEOP[33]	AIF360	Post	Calibration, FPR/FNR		
EOP[20]	AIF360	Post	EO		
ROC[25]	AIF360	Post	SP/EO/TPR		
TO[7]	Fairlearn	Post	SP/EO/ TPR/FPR/ TNR/FNR		

TABLE 3.1: An overview of the discussed bias mitigation algorithms, showing their characteristics.

Chapter 4

FairAST

In this chapter, we introduce the Fair Algorithm Selection Tool (FairAST). The purpose of this tool is to help the user to find the best bias mitigation algorithm for a given dataset, while considering a specific fairness measure. The user provides a dataset and a specific fairness measure the results should be optimized for. Based on the user's input, the tool recommends a set of algorithms that give the highest improvements on the desired fairness measure. Our target is that the best performing algorithm that can be found using exhaustive search should be among the top three recommendations of our tool. FairAST is implemented in Python and all code is available on GitHub[1].

FairAST consists of four components: the Data Preparation module, the Profiling module, the Algorithms and the Recommendation. First, the data will be prepared to be in the correct format so the tool and the algorithms can handle the data properly. Next, a profile of the data will be generated using the profiling module. The third component stores all the requirements of the bias mitigation algorithms and the methods that enables the tool to run the algorithms. The final component makes the recommendation by checking the data profile against the algorithms' requirements and testing the suitable algorithms. The architecture of the tool is presented in Fig. 4.1. In the following sections the different components will be explained in more detail.

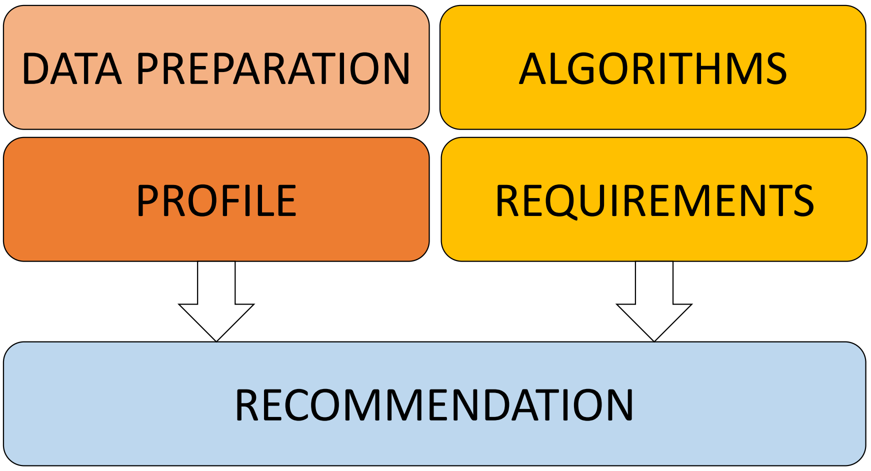

FIGURE 4.1: An overview of the architecture of FairAST: the dataset and its profile as well as the algorithms and their requirements are used to make a recommendation.

[1] https://github.com/qahtanaa/FairnessAlgSelection

4.1 Data preparation

The process of making a recommendation starts with the data. In order for the tool and the algorithms to process the data correctly, the data should be in the right format. Therefore, it is the first step to prepare the data in the DataPreparation module. As input, the module requires the data and a set of properties of the data that the user needs to specify. These requirements and the transformations performed during the data preparation are shown in Fig. 4.2.

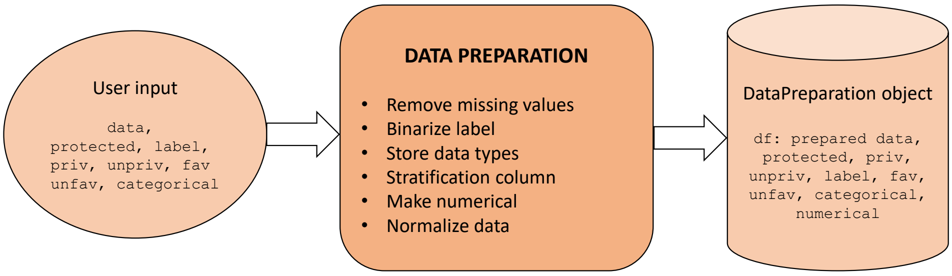

FIGURE 4.2: The tasks performed in the DataPreparation module, based on the user's input.

The input data should be a DataFrame, consisting of the label and the features the user would like to use for classification. The data should eventually be free of missing values, since the algorithms cannot handle these. Therefore, all rows containing missing values will be removed from the data during the data preparation. If the user wants to handle missing values differently, this should be done before passing the data to the tool.

In addition to the data itself, some properties need to be specified. For the label, the user needs to indicate which column in the data represents the class label, and which values are considered the favorable and the unfavorable outcome. This implementation is only suitable for binary classification tasks, so there should be only two different values in the column. Based on the provided information, the label values will be transformed into 0 for the unfavorable labels and 1 for the favorable labels.

The user also needs to decide which features will be considered the protected attributes, and which groups in the data are the privileged and the unprivileged groups. Here it is important for the user to decide whether there will be only one or more protected attributes, and whether the protected attributes are (turned into) a binary representation or multi-valued.

Next, a column is added to the data that will be used for stratification of the data when the data is split into a train and test set. The column represents the subgroups in the data in terms of whether an instance has been assigned to the (un)privileged group and whether it has the favorable or the unfavorable label. Using this information for stratification guarantees the base rates of the privileged and unprivileged groups to be the same across the train and test set.

Furthermore, the type of data of each feature is identified. A feature can be either categorical or numerical. All features with non-numerical data or only two different values are considered categorical. In case the input data has numerical features that nevertheless should be considered categorical, the user can specify this in the input. The categorical feature values are transformed into numerical values, because the algorithms cannot work with strings. Each different category is changed into a specific integer. The numerical data will be normalized by scaling the data between 0 and 1. Doing so prevents the data to be too influential when the numbers are big.

Calling the 'prepare' method performs all of the actions explained above consecutively. The result is a DataPreparation object that has multiple attributes, 'df' being the most important, because this is the ultimate data in the proper format that will be used in the following steps.

4.2 Profiling

The second step is to make a profile of the data that includes the important properties of the data and the preferences of the user (see Fig. 4.3). This profile will in the final step be used to see which algorithms are suitable for this dataset.

The profile stores the attributes that have shown to be important according to the analysis of the algorithms in section 3.4. These are the number of protected attributes, the number of unique feature values, the number of (un)privileged groups, the type of data, the kind of label, and the balance of the labels. The Profile class retrieves all of this information by inspecting the data.

The fairness measure that should be optimized must be specified by the user. In the current implementation, the user can choose one out of nine different fairness measures ('SP', 'EO', 'TPR', 'FPR', 'C', 'TNR', 'FNR', 'PPV', 'FDR'). Additionally, the user can restrict the tool to only consider specific bias mitigation techniques (pre-, in-, or post-processing). By default all techniques are allowed.

All this information is stored in a dictionary where the keys are the profile's attributes and the values are the corresponding properties of the data. Calling the 'create_profile' function immediately creates and returns the profile.

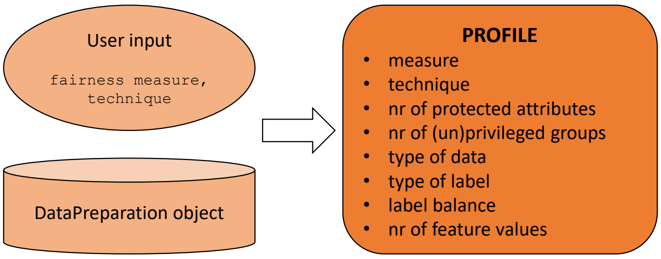

FIGURE 4.3: Based on the data and the user input, the profile is created.

4.3 Algorithms

The third components consists of two main elements: the requirements and the algorithms. An overview of the elements in these two parts is presented in Fig. 4.4. As input, this component only requires the previously specified fairness measure the user wants to optimize for, and whether the algorithm should be optimizing for this measure specifically or not. This will be further explained in the following section.

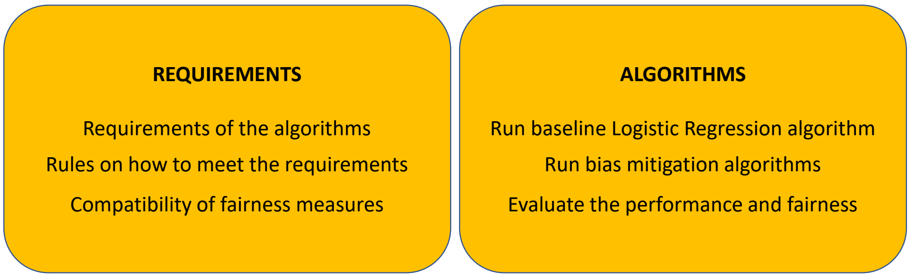
REQUIREMENTS
Requirements of the algorithms
Rules on how to meet the requirements
Compatibility of fairness measures
ALGORITHMS
Run baseline Logistic Regression algorithm
Run bias mitigation algorithms
Evaluate the performance and fairness

FIGURE 4.4: The Algorithms component stores the requirements of the algorithms and the methods that are used to run the algorithms.

4.3.1 Requirements

Based on the analysis in section 3.4, the algorithms have been found to have certain requirements. For example, some algorithms can only handle one protected attribute, where others can deal with multiple protected attributes. For each bias mitigation algorithm, the restrictive requirements are stored in a dictionary, where the keys correspond to the keys of the data profile. Additionally, rules are specified that tell in which way the requirements should be met. For example, for the maximum number of protected attributes, the value of the profile should be smaller or equal to the requirement of the algorithm, while for the type of label the values should match.

A special requirement is the fairness measure the algorithm optimizes for. We know from the impossibility theory explained in section 2.3 that some fairness measures cannot be fulfilled at the same time. In these cases, optimizing for one measure implies that the other is impossible to simultaneously improve. However, many fairness measures are not necessarily incompatible with each other. So, if an algorithm optimizes for one specific measure, other measures might as well improve. Therefore, the user can allow to not only select algorithms that are built to optimize the desired fairness measure, but also the algorithms that optimize for other compatible measures. If the user does not want this, the setting 'strict' can be switched to only choose algorithms that specifically optimize the desired fairness measure.

4.3.2 Algorithms

Besides all the rules and requirements, the section of Algorithms stores the functions that enable the execution of all bias mitigation algorithms. For algorithms that can choose between different fairness measures to optimize for, it makes sure the algorithm chooses the one that matches the user's desired fairness measure, or a compatible measure if that is allowed. It can also run a logistic regression classifier without bias mitigation in order to see the difference/progress made by the mitigation algorithm. Moreover, the logistic regression classifier is used to make the predictions after the data has been transformed by the pre-processing algorithms, and the original classifications of the logistic regression classifier are used by the post-processing algorithms. After each run, the results of the performance and fairness measures on the predictions are returned. The implemented performance measures are the balanced accuracy score for the privileged and the unprivileged group, precision, recall and the F1-score.

4.4 Recommendation

The final component of the tool incorporates the data, the profile and the algorithms to find the best algorithm to use on the given dataset. The process of making the recommendation is visualized in Fig. 4.5.

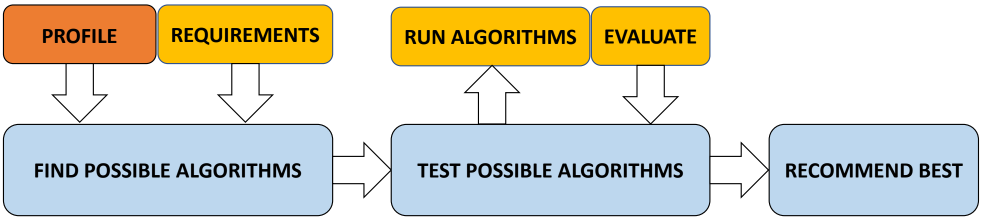

FIGURE 4.5: The final recommendation is based on the profile and the results obtained from testing the algorithms.

First, the tool collects all the algorithms that are possible to use on the given dataset. To determine which algorithms are suitable, it needs to receive the data profile and the requirements of the algorithms that are created in the previous stages. This enables the recommendation algorithm to check for each profile feature of the data whether it is in line with the algorithm's requirements. If all requirements are met, the algorithm will be added to the list of possible algorithms. Pseudocode for this step is provided in line 1 - 13 of Algorithm 1.

After checking which algorithms are suitable to use for the given dataset, there are three options: 1) there are no suitable algorithms, 2) there are only one or two suitable algorithms, or 3) there are multiple suitable algorithms. In the first case, there can no recommendation be made, other then that the user could try to make different decisions in the data preprocessing. In the second case, the one or two algorithms will both be trained and tested on the dataset to see which one performs better. In case there are multiple suitable algorithms, the algorithms will first be trained and tested on a sample of the data (see also line 14 - 22 of Algorithm 1).

It is up to the user to decide how many times the algorithms should be tested on a sample, by specifying the n number of sample runs. For the sample, 5000 examples of the dataset will be taken, of which 70 percent will be used for training, and the rest for testing. The advantage of testing on a sample is that it will be faster than using the entire dataset to train and test the model. A disadvantage is that the sample might not be fully representative of the whole dataset, causing the results to be different from the results after training on a bigger portion of the dataset. However, using stratification at least makes sure that the distribution of groups (privileged/unprivileged with a favorable/unfavorable label) is similar as in the complete dataset.

For each algorithm, the specified fairness measure is calculated on the predictions made on the test set after training the bias mitigation algorithm on the sample of the data (line 17). After all algorithms have been tested on the sample, the algorithms are ranked from best to worst based on their performance on the fairness measure. Each algorithm receives a score based on their rank in the list, where the best performing algorithm receives the highest score (line 19). After each sample run, their received score is accumulated to their total score (line 20). When all sample runs have been performed, the k number of algorithms with the highest scores are in the set of recommended algorithms. These algorithms are now tested once on the complete dataset to see their results.

Algorithm 1 FairAST recommendation algorithm

Input: A dataset D, a data profile $\mathcal{P}$, a list of algorithms $\mathcal{A}$, Algorithms requirements $\mathcal{R}$.
Output: A set of recommended algorithms $A_r \subseteq \mathcal{A}$.

```
$A_p \leftarrow []$
for $A \in \mathcal{A}$ do
    $S(A) \leftarrow 1$
    for $att \in \mathcal{P}$ do
        if $att$ does not satisfy $\mathcal{R}(A)$ then
            $S(A) \leftarrow 0$
            break
        end if
    end for
    if $S(A) == 1$ then
        $A_p \leftarrow A_p \cup \{A\}$
    end if
end for
for $i \leftarrow [1, ..n]$ do
    for $A \in A_p$ do
        $D_s \leftarrow sample(D)$
        $m[A] \leftarrow fairness(D_s, A, \mathcal{P}.measure)$
    end for
    $Score[i] \leftarrow Sort(A_p)$ based on $m$
    $TotalScore \leftarrow Accumulate(Score)$
end for
$A_r \leftarrow top_k(Sort(TotalScore))$
```

4.5 Final remarks on FairAST

FairAST is meant to help researchers and practicioners in the field of ML to quickly find a bias mitigation algorithm that suits their dataset. Although the user should perform a couple of pre-processing steps to clean the data and identify the biases, all other steps are completely automated. This saves a lot of time, as the pre-processing, profiling and ensuring the correct implementation of all different kinds of bias mitigation algorithms can be very time consuming. This framework adjusts the data for every algorithm to be in the correct format, and enables the user to execute the algorithms for a specific fairness measure. By showing not only the name of the recommended algorithm, but also the results of the performance and fairness measures, the user can understand how the recommendation was made, and see the performance differences for the algorithms. This can help the user to make an informed decision on the algorithm to use.

The user can choose to use the different parts of the recommendation tool independently, or to perform all steps (from data preparation, profiling and recommending) directly by running a single command. For example, when the user is only interested in the profile of the data, it is possible to only execute the data preparation and the profiler.

Furthermore, new bias mitigation algorithms can easily be included to the recommendation tool, by following a couple of steps. The requirements of the algorithm, and the method of executing the algorithm can be added to the Algorithms module. Only in case the algorithm requires to know a new characteristic of the dataset, this has to be added to the profiler as well.

Chapter 5

Evaluation

In this chapter the proposed recommendation tool FairAST will be evaluated. This will be done by comparing FairAST's recommendations to the best performing algorithms found through exhaustive search. For the experiments, three datasets are used, these will be discussed in section 5.1. In section 5.2 the different measures for evaluating the performance of the algorithms and the recommendation tool are explained. Next, the experimental set-up for both the evaluation of the algorithms and FairAST are discussed. Finally, the results are presented and discussed in section 5.4.

5.1 Datasets

For the evaluation, three different datasets are used: the Compas dataset from ProPublica [34], the Adult census income data from the UCI Machine Learning Repository [13], and the Hospital admission data from [22]. In the following sections the characteristics of the datasets are discussed, as well as the cleaning process and the decisions made regarding the protected attributes. In section 5.3.1 will be further clarified which pre-processing steps are applied. An overview of the important characteristics of these datasets is presented in Table 5.1.

	Compas	**Adult**	**Hospital**
Domain	justice	income	healthcare
Instances	7185	30162	528820
Favorable	didn't recidivate	>50K	admit
Unfavorable	did recidivate	<=50K	discharge
Class balance	1.22	0.33	0.45
Categorical	4	8	9
Numerical	5	4	6
Protected	race, sex	race, sex	race
DI ratio	0.73	0.38	0.62

TABLE 5.1: The characteristics of the datasets Compas, Adult and Hospital.

Compas: The first dataset is ProPublica's Compas recidivism dataset [34]. This dataset is used to predict whether a criminal defendant is likely to re-offend within two years, based on the person's demographical features, criminal history and current charge. This dataset is widely used in fairness research, since the data shows disparities for race and gender.

The data used in this study comes from the ’compas-scores-two-years.csv’ file from ProPublica [34]. First, the data is cleaned by removing features that are not useful for the classification task. Also, rows with missing values are removed.

After cleaning, the data consists of 7185 instances, representing individual people charged for a crime. The label ’two_year_recid’ describes whether a person did or did not recidivate within two years, where the latter is the favorable label. There are 8 features (4 categorical and 5 numerical features). The categorical features are ’race’, ’sex’, ’charge_degree’, and ’charge description’. The numerical features are counting the number of juvenile felonies, misdemeanors or other types of crimes. Similarly, ’priors_count’ denotes the number of prior crimes. The ’age’ of the person is stored as a continuous value.

Data profiling shows that the set consists of 1387 females and 5798 males, with base rates of respectively 0.64 and 0.53. For race, the largest groups are the African-American (3684, base rate 0.485) and the Caucasian (2445, base rate 0.607). When race and sex are inspected together, it shows that the African-American males have a much lower base rate than all other groups (disregarding the Asian and Native America females, since these groups are too small with respectively 2 and 4 instances). These base rates are presented in Table 5.2. Furthermore, the labels in the dataset are fairly balanced, with a ratio of 1.22.

Since there are multiple sensitive attributes (race and sex), a feature is added that combines the two sensitive features into one, anticipating algorithms that can only handle one sensitive attribute. The separate race and sex features are removed from the data. The sensitive feature is binarized by singling out the group with a notably different base rate and taking together all other subgroups. In this case, the unprivileged group consists of African-American men, with a base rate of 0.456. All other groups (race-sex combinations) taken together are the privileged group, with a base rate of 0.618.

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

TABLE 5.2: The base rates for all possible race-sex groups in the Compas dataset.

Adult: The Adult dataset consists of census income data and is used to predict whether a person will make a high or a low income. The prediction is based on a person’s demographical features, education and work situation. This dataset is also widely for fairness research on the race and sex features in the data.

After removing the uninformative features and rows with missing values, the data consists of 30.162 instances. The label ′income_per_year′ describes whether a person receives a high (>50K) or a low (<=50K) income per year. There are 8 categorical features (′race′, ′sex′, ′education′, ′work class′, ′relationship′, ′marital status′, ′occupation′, and ′native country′). The 4 numerical features are ′age′, ′capital gain′, ′capital loss′, and ′work hours′.

The sensitive features are race and sex. Again, these two features are combined and turned into a binary value with 0 for the unprivileged groups and 1 for the privileged group. Similar as for the Compas dataset, the base rates of all subgroups in the data are inspected to see which groups more often receive the favorable outcome than other groups, as shown in Table 5.3. Here, the privileged group consists of the White males and the Asian/Pac Islander males, with a joint base rate of 0.33. The other subgroups together form the unprivileged group, with a base rate of 0.12. As also can be seen from these baserates, the number of favorable labels in this dataset is low in comparison to the number of unfavorable labels. The ratio of the number of favorable over the unfavorable labels is 0.33, which shows a great imbalance.

Race	Sex	Base rate
White	Female	0.12
	Male	**0.33**
Asian-Pac-Islander	Female	0.14
	Male	**0.34**
Black	Female	0.06
	Male	0.20
Amer-Indian-Eskimo	Female	0.10
	Male	0.13
Other	Female	0.05
	Male	0.12

TABLE 5.3: The base rates for all possible race-sex groups in the Adult dataset.

Hospital: The third dataset is used in the context of hospital admission for people at the emergency department (ED). According to research into the racial differences in the American healthcare system, there are significant disparities in the way people of different ethnicities are treated and admitted to hospitals [43]. Therefore, training a classifier on historic data of hospital admissions might lead to biased classifications. Inspecting the data shows in fact racial disparities, proving that this dataset is interesting to use for fairness purposes.

The original dataset consisted of more than 900 features. Based on the paper describing this dataset [22], only the most informative features for classification are used, leaving us with 60 features. These are demographical features, history of hospital visits and treatments, and the use of medicines. These features are used to determine whether the person should be admitted to the hospital or not.

After cleaning, the data consists of 528.820 observations. The label ′disposition′ describes whether the patient was Admitted or Discharged. The categorical features consist of ′gender′, ′race′, ′marital status′, ′employment status′, ′insurance status′, ′esi′, ′arrival mode′, ′previous disposition′. The numerical features consist of the number of ′surgeries′, ′ED visits′, ′hospital admissions′, and ′age′.

The final 48 features all represent a different category of medicines people can take, counting the number of medicines they take in that category. These features are transformed into three different features: one counting the total number of medicines, one specifying in how many categories a person takes medicines, and 'top meds' specifies the category the most medicines are taken from.

The sensitive feature is 'race', where 'White or Caucasian' with a base rate of 0.375 is the privileged group. The other groups together are considered the unprivileged group, with a base rate of 0.233. Similar as for the Adult dataset, the data shows an imbalanced distribution of favorable and unfavorable labels, with a ratio of 0.45.

5.2 Measures

The performance of the algorithms will be evaluated by both performance and fairness measures. The performance measures consist of the balanced accuracy conditioned on the privileged and unprivileged group, and the F1-score.

- Accuracy is a measure that reflects the number of correct classifications. To adjust for imbalanced datasets, the balanced accuracy score is used. This score is calculated as follows [3]:

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

  It is split into a balanced accuracy for the privileged and the unprivileged group to compare the accuracy between these groups.

- The F1-score reflects the harmonic mean of the precision and recall. Precision tells how many of the items that are predicted to belong to the favorable class actually belong to that class, calculated by $TP/(TP+FP)$. Recall tells how many of the items that actually belong to the favorable class are also predicted to belong to that class, which is calculated by $TP/(TP+FN)$. These scores together are used to calculate the F1-score[3]:

  $$F1 = 2 * \frac{precision * recall}{precision + recall}$$

The fairness of the models will be evaluated by multiple fairness measures. To measure SP, the disparate impact ratio will be used. For EoO, the (absolute) TPR difference will be calculated. For EO, the average absolute odds difference is taken. Consistency is calculated to measure individual fairness.

- the DI ratio is the percentage of favorable outcomes of the unprivileged group divided by the percentage of favorable outcomes of the privileged group:

  $$DI = Pr(Y=1|unprivileged)/Pr(Y=1|privileged)$$

  A score of 1 means that the base rates of the two groups are exactly equal, but a score between 0.8 and 1.25 is usually already considered to be fair.

- The TPR difference shows the difference in the True Positive Rates between the privileged and the unprivileged group. The TPR is calculated by $TP/(TP+$

$FN)$. The absolute difference is calculated by:

$$|TPR(unpriv) - TPR(priv)|$$

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

- For EO, the TPR as well as the FPR differences should be equal for the privileged and the unprivileged group. The FPR is calculated by $FP/(FP + TN)$. To calculate EO, the average of the absolute TPR and FPR differences is calculated:

$$\frac{1}{2}|FPR(unpriv) - FPR(priv)| + |TPR(unpriv) - TPR(priv)|$$

- Consistency measures the individual fairness by calculating the similarity of the labels regarding the similarity of the features. A score close to 1 means the data is more consistent.

5.3 Experimental set-up

The experimental evaluation consists of two kinds of experiments. First, all the bias mitigation algorithms will be tested to see how well they perform and reduce bias. Second, the recommendation tool will be used to give recommendations for the three datasets and different fairness measures. The quality of these recommendations will be evaluated by comparing them to the best performing algorithms found in the initial experiments, that represent the results one would obtain by performing an exhaustive search for the best performing algorithm. In the following sections the set-up of the experiments will be explained.

5.3.1 Data pre-processing

For the recommendation algorithm, the data does not have to be preprocessed any further, because this will be done by the tool itself. However, for the first experiments these steps do need to be taken. So for all datasets, the favorable label will be denoted with 1 and the unfavorable label with 0. All categorical features will be transformed to numerical values by changing each category into a different integer. The numerical values are scaled between 0 and 1 by using the MinMaxScaler from Sklearn.

5.3.2 Evaluating the algorithms

For the experiments on the algorithms, we start with a baseline algorithm that shows the classifications being made without interventions for fairness. The baseline algorithm is Sklearn's Logistic Regression classifier with as only setting "solver='liblinear'". This baseline is used to see to what extent the bias mitigation algorithms improve the fairness relative to a regular classifier. The logistic regression classifier is also used for making the predictions in the pre-processing and post-processing methods.

We perform 10 runs on each (bias mitigation) algorithm for all three datasets. In each run, the dataset is split into a train (70%) and a test (30%) set. Stratification is used to ensure an equal ratio of favorable and unfavorable outcomes of the privileged and unprivileged group. This means that the disparate impact ratio is similar in the train and test set. After each run, the performance and fairness measures are calculated. After all 10 runs, the average and the standard deviation of these results are taken.

Some algorithms are able to optimize for different fairness measures. Since we will evaluate the performance of the algorithms on SP, EO and TPR, the algorithms are set to optimize for these measures. This means that EGR, GSR, ROC and TO will be tested three times; once for each measure. For algorithms that allow to specify the extent to which the algorithm should try to optimize fairness, the maximum value was chosen (DIR, CR, PR), since we aim to improve fairness as much as possible. Also for FC, it was chosen to set 'apply fairness constraints' to 1, since we want to optimize fairness rather than accuracy. Other hyperparameters were set to the values in accordance with the recommended settings in the documentation.

Besides, based on the requirements of the algorithms, we know that GFC, LFR and OP might not operate well on our chosen datasets and protected attributes. Nevertheless, these algorithms will be tested to confirm these expectations. Additional experiments will be performed on these algorithms with different settings, to establish the performance results of these algorithms when the data is suitable for the algorithm. Since these results are not relevant for the evaluation of FairAST, these experiments and their results are presented in Appendix A.

5.3.3 Evaluating FairAST

As explained in Chapter 4, FairAST recommends a set of bias mitigation algorithms that give the best results on a specific fairness measure. We test two different settings of the recommendation tool: one where the tool runs the possible algorithms only once on a sample of the data (FairAST (1)), and one where the tool runs the possible algorithms three times on a sample of the data (FairAST (3)). The parameter settings of the algorithms in FairAST are the same as explained in the previous section. We will show the performance of the top three recommended algorithms in the results.

For each dataset we try three different fairness measures: SP, EO and TPR, and we allow FairAST to also test algorithms that are optimizing for compatible measures. So, for both experiments taken together this will give 18 recommendations. The quality of these recommendations is evaluated by comparing them to the top three best performing algorithms according to the results obtained by the exhaustive search.

5.4 Results

In this section, the results of the experiments will be presented and discussed. First, we will investigate how well the bias mitigation algorithms perform on the three datasets. Next, the quality of the recommendations given by FairAST will be discussed.

5.4.1 Performance of the algorithms

The results of the bias mitigation algorithms on the Compas, Adult and Hospital dataset are presented in respectively Table 5.4, Table 5.5, and Table 5.6. The first row of each algorithm shows the mean, and the second row the standard deviation for each measure based on 10 runs.

Starting with the Logistic Regression classifier, we see that the SP, thus the disparate impact ratio, is now even lower than originally in the data. For Compas, the ratio has decreased from 0.73 to 0.49, for Adult from 0.38 to 0.15, and for Hospital from 0.62 to 0.50. This shows that the regular classifier exacerbates the disparities in the data, which underlines the importance of bias mitigation in classifiers.

Important to note is that for LFR and OP there were no results obtained on the given datasets, as could be expected from the analysis of the algorithms. For LFR, the algorithm would transform the data in such a way that there was only one class left in the data, making it impossible to test the data on the logistic regression classifier, since this algorithm requires both classes to be present in the data. For OP, the number of features and unique feature values was too big, resulting in a memory error. The additional experiments can be found in Appendix A.

The GFC was also expected to not perform well on these datasets, because it is not built for optimizing fairness on a single binary attribute. The results show that the algorithm indeed does not perform well: the accuracy is much lower than for the other algorithms, and the F1 score shows a high deviation. The TPR and EO score seem good, but for SP the deviation is again very high (0.369). Looking at the results for the single runs (see Appendix B), shows that the classifier often predicts only or mostly one class, leading to these results. Since the algorithm is meant to optimize fairness constraints for more complex groups, it might be the case that a single protected attribute results in an overfitting classifier. The extra experiments on GFC with multiple protected attributes are presented in Appendix A.

There is another bias mitigation algorithm that stands out from the others, which is CEOP. This is the only algorithm for which the results are worse from the baseline algorithm. However, this is in line with the findings in the paper [33]. CEOP is made to satisfy calibration, and then tries to improve on either FPR or TPR. As mentioned in the paper, the algorithm does not always succeed in fulfilling these two measures, and moreover, it always comes at cost of other fairness and performance measures.

For all other bias mitigation algorithms, we see that the TPR, EO and SP are improved from the Logistic Regression algorithm. The consistency remains approximately equal on each dataset. The standard deviations for most of the results are also very small, often smaller than 0.05. This shows that the results are reasonably consistent over each run. Only for MFC we see some deviation of more than 0.1, especially for SP, although this is the measure the classifier optimizes for. The deviations could be explained by the algorithm's use of gradient descent to find the optimal model. In some cases, the model might find a local minimum that does not give the optimal results. Unfortunately, the settings for the number of runs or the step size are not changeable by the user.

The performance results (accuracy and F1) stay relatively similar to the Logistic Regression classifier, which means that the algorithms are, despite improving fairness, able to maintain the ability to make good classifications. This is important, because it shows that the classifier is not randomly making predictions that satisfy the fairness criteria, or giving the trivial solution of predicting only one class.

For DIR, we see some differences in the performance on the three datasets. The SP obtained on the Compas dataset (0.782) is much higher than for Adult and Hospital (0.567 and 0.557). This can be explained by the fact that the bias mitigation in DIR is only applied on the numerical features. For Compas, there are only two categorical features, while Adult and Hospital have respectively 6 and 8 categorical features. Since these features are not transformed by the algorithm, these features can still carry some differences for the privileged and the unprivileged group. Therefore, the predictions of the classifier can be influenced by these differences, and the SP is not satisfied.

A similar effect can be found by the PR algorithm, where the SP is for all datasets the lowest compared to the other algorithms. This algorithm aims to reduce the influence of the specified protected attributes on the decisions of the classifier. In fact, training a logistic regression classifier based on fairness through unawareness, in which the protected attributes are removed from the data, yields almost exactly the same results (see Appendix C). Unfortunately, fairness through unawareness has been shown to be problematic, since other features can correlate with the protected attributes, and therefore still produce biased outcomes.

Another interesting results is that algorithms optimizing for measure X can perform better on a measure Y than algorithms specifically optimizing for measure Y. For example, the EOP optimizing for EO, obtains an average SP of 0.963 on the Compas dataset, while the DIR, despite optimizing for SP, shows an average of 0.782. Results like these show that it is interesting for a user to not only consider to use algorithms that optimize for the desired fairness measure, but also to allow algorithms that optimize for compatible measures, as this can result in even more improvements.

Moreover, this can even be the case within one algorithm. For example, the ROC optimized for EO performs better on the TPR measure, than the same algorithm optimizing for TPR, although the differences are relatively small. Similar observations can be made for results on EO and TPR for EGR, GSR and TO.

Method	Acc p	Acc up	F1	TPR	EO	SP	C
Log.Reg.	0.619	0.659	0.729	0.333	0.374	0.489	0.855
	0.011	0.014	0.007	0.020	0.023	0.023	0.007
AD	0.626	0.650	0.723	0.093	0.102	0.840	0.877
	0.022	0.021	0.018	0.067	0.094	0.133	0.024
CEOP	0.583	0.659	0.727	0.363	0.439	0.458	0.852
	0.011	0.014	0.007	0.020	0.028	0.024	0.007
CR	0.649	0.636	0.706	0.113	0.125	1.127	0.848
	0.011	0.017	0.010	0.021	0.015	0.029	0.003
DIR	0.637	0.661	0.730	0.090	0.114	0.782	0.941
	0.011	0.015	0.008	0.023	0.023	0.033	0.007
EGR (EO)	0.643	0.651	0.718	0.016	0.023	0.932	0.836
	0.011	0.015	0.007	0.014	0.013	0.032	0.009
EGR (SP)	0.646	0.646	0.716	0.042	0.041	0.990	0.841
	0.012	0.012	0.008	0.020	0.021	0.033	0.006
EGR (TPR)	0.644	0.657	0.723	0.022	0.031	0.890	0.838
	0.011	0.015	0.008	0.012	0.015	0.025	0.007
EOP	0.597	0.591	0.677	0.018	0.021	0.963	0.718
	0.012	0.009	0.008	0.010	0.007	0.023	0.012
FC	0.621	0.655	0.713	0.124	0.158	0.722	0.864
	0.009	0.008	0.004	0.020	0.012	0.016	0.006
GFC	0.524	0.510	0.525	0.056	0.042	0.714	0.964
	0.031	0.015	0.230	0.062	0.042	0.369	0.022
GSR (EO)	0.611	0.645	0.728	0.129	0.164	0.739	0.854
	0.031	0.032	0.010	0.049	0.046	0.081	0.014
GSR (SP)	0.554	0.589	0.727	0.070	0.104	0.859	0.921
	0.036	0.038	0.006	0.050	0.054	0.083	0.040
GSR (TPR)	0.612	0.659	0.744	0.115	0.162	0.756	0.866
	0.010	0.013	0.005	0.012	0.014	0.016	0.008
MFC	0.622	0.660	0.735	0.144	0.182	0.725	0.856
	0.013	0.025	0.012	0.079	0.102	0.135	0.016
PR	0.634	0.668	0.731	0.161	0.195	0.679	0.843
	0.011	0.019	0.009	0.016	0.012	0.015	0.005
Rew	0.647	0.649	0.713	0.073	0.071	1.036	0.846
	0.011	0.014	0.010	0.021	0.019	0.032	0.004
ROC (EO)	0.647	0.667	0.699	0.026	0.035	0.880	0.833
	0.011	0.018	0.010	0.020	0.019	0.039	0.006
ROC (SP)	0.652	0.667	0.682	0.045	0.038	0.956	0.832
	0.010	0.019	0.012	0.026	0.011	0.050	0.004
ROC (TPR)	0.651	0.670	0.692	0.043	0.051	0.847	0.831
	0.011	0.017	0.014	0.034	0.020	0.069	0.005
TO (EO)	0.641	0.639	0.714	0.016	0.026	0.947	0.839
	0.011	0.022	0.020	0.013	0.010	0.031	0.011
TO (SP)	0.652	0.660	0.700	0.058	0.050	1.001	0.824
	0.011	0.015	0.012	0.020	0.016	0.027	0.007
TO (TPR)	0.650	0.661	0.710	0.022	0.027	0.897	0.826
	0.013	0.016	0.009	0.011	0.014	0.030	0.010

TABLE 5.4: Results of the experiments on the Compas dataset. Shows the mean and standard deviation over 10 runs.

Method	Acc p	Acc up	F1	TPR	EO	SP	C
Log.Reg.	0.695	0.590	0.535	0.292	0.187	0.152	0.932
	0.007	0.006	0.009	0.017	0.008	0.007	0.002
AD	0.712	0.789	0.607	0.153	0.084	0.655	0.936
	0.014	0.025	0.021	0.066	0.034	0.123	0.004
CEOP	0.695	0.500	0.506	0.483	0.287	0.000	0.941
	0.007	0.000	0.010	0.012	0.007	0.000	0.002
CR	0.641	0.676	0.455	0.095	0.059	0.815	0.934
	0.007	0.012	0.011	0.032	0.016	0.052	0.002
DIR	0.652	0.654	0.472	0.027	0.017	0.567	0.961
	0.011	0.009	0.019	0.019	0.009	0.052	0.003
EGR (EO)	0.661	0.659	0.490	0.029	0.019	0.531	0.930
	0.007	0.012	0.010	0.020	0.011	0.041	0.003
EGR (SP)	0.639	0.676	0.451	0.102	0.065	0.848	0.932
	0.008	0.015	0.014	0.036	0.019	0.069	0.003
EGR (TPR)	0.661	0.660	0.490	0.026	0.018	0.526	0.930
	0.006	0.011	0.009	0.020	0.011	0.038	0.003
EOP	0.585	0.587	0.315	0.016	0.010	0.642	0.913
	0.007	0.006	0.011	0.012	0.007	0.021	0.002
FC	0.725	0.773	0.619	0.148	0.100	0.596	0.937
	0.034	0.019	0.048	0.079	0.028	0.270	0.005
GFC	0.614	0.583	0.458	0.085	0.054	0.473	0.979
	0.079	0.058	0.042	0.062	0.038	0.364	0.014
GSR (EO)	0.640	0.636	0.440	0.021	0.012	0.558	0.944
	0.026	0.034	0.063	0.011	0.005	0.026	0.010
GSR (SP)	0.640	0.634	0.441	0.019	0.012	0.563	0.943
	0.006	0.007	0.011	0.015	0.007	0.049	0.001
GSR (TPR)	0.646	0.644	0.455	0.020	0.015	0.610	0.945
	0.017	0.017	0.037	0.016	0.008	0.107	0.008
MFC	0.707	0.711	0.548	0.059	0.070	0.764	0.892
	0.034	0.011	0.022	0.038	0.048	0.100	0.008
PR	0.667	0.644	0.498	0.075	0.052	0.420	0.927
	0.007	0.009	0.010	0.023	0.010	0.019	0.002
Rew	0.659	0.667	0.489	0.025	0.015	0.569	0.931
	0.006	0.011	0.010	0.014	0.008	0.034	0.003
ROC (EO)	0.739	0.721	0.582	0.057	0.039	0.725	0.893
	0.007	0.008	0.006	0.019	0.012	0.020	0.003
ROC (SP)	0.738	0.718	0.568	0.019	0.038	0.896	0.891
	0.007	0.008	0.007	0.012	0.007	0.021	0.003
ROC (TPR)	0.737	0.718	0.573	0.026	0.021	0.800	0.891
	0.007	0.007	0.007	0.026	0.014	0.041	0.002
TO (EO)	0.669	0.669	0.505	0.027	0.016	0.588	0.925
	0.011	0.017	0.020	0.021	0.011	0.033	0.005
TO (SP)	0.639	0.689	0.454	0.140	0.090	0.972	0.929
	0.007	0.013	0.013	0.030	0.016	0.062	0.003
TO (TPR)	0.673	0.675	0.513	0.028	0.017	0.580	0.926
	0.011	0.014	0.017	0.021	0.011	0.041	0.005

TABLE 5.5: Results of the experiments on the Adult dataset. Shows the mean and standard deviation over 10 runs.

Method	Acc p	Acc up	F1	TPR	EO	SP	C
Log.Reg.	0.797	0.758	0.718	0.132	0.093	0.498	0.954
	0.002	0.002	0.002	0.005	0.003	0.004	0.000
AD	0.800	0.789	0.733	0.043	0.031	0.622	0.958
	0.002	0.003	0.002	0.005	0.003	0.007	0.001
CEOP	0.797	0.677	0.678	0.305	0.185	0.341	0.939
	0.002	0.002	0.002	0.007	0.004	0.005	0.000
CR	0.768	0.792	0.697	0.087	0.063	0.894	0.951
	0.001	0.002	0.002	0.006	0.003	0.005	0.000
DIR	0.792	0.764	0.715	0.092	0.063	0.557	0.973
	0.002	0.002	0.002	0.006	0.004	0.006	0.001
EGR (EO)	0.785	0.780	0.713	0.019	0.013	0.672	0.952
	0.002	0.003	0.002	0.008	0.004	0.008	0.001
EGR (SP)	0.768	0.794	0.696	0.096	0.070	0.922	0.948
	0.001	0.002	0.002	0.006	0.003	0.006	0.002
EGR (TPR)	0.785	0.780	0.712	0.018	0.013	0.673	0.952
	0.002	0.003	0.002	0.008	0.004	0.008	0.001
EOP	0.749	0.747	0.657	0.006	0.003	0.727	0.903
	0.002	0.003	0.002	0.005	0.002	0.007	0.000
FC	0.796	0.780	0.726	0.056	0.040	0.598	0.964
	0.002	0.001	0.002	0.005	0.003	0.009	0.000
GFC	0.695	0.683	0.538	0.074	0.062	0.527	0.970
	0.050	0.086	0.110	0.005	0.035	0.073	0.002
GSR (EO)	0.751	0.744	0.660	0.024	0.015	0.672	0.950
	0.018	0.010	0.025	0.010	0.006	0.024	0.002
GSR (SP)	0.764	0.751	0.680	0.041	0.027	0.622	0.951
	0.004	0.002	0.005	0.009	0.005	0.013	0.001
GSR (TPR)	0.747	0.743	0.656	0.019	0.016	0.653	0.949
	0.004	0.003	0.006	0.006	0.003	0.006	0.001
MFC	0.757	0.756	0.664	0.052	0.092	0.829	0.959
	0.057	0.065	0.060	0.027	0.067	0.251	0.008
PR	0.795	0.761	0.717	0.114	0.080	0.521	0.951
	0.002	0.003	0.003	0.006	0.003	0.005	0.001
Rew	0.782	0.783	0.710	0.007	0.005	0.717	0.952
	0.002	0.002	0.002	0.004	0.002	0.005	0.000
ROC (EO)	0.807	0.802	0.731	0.049	0.044	0.676	0.950
	0.002	0.002	0.002	0.004	0.002	0.002	0.001
ROC (SP)	0.807	0.804	0.719	0.036	0.039	0.878	0.951
	0.002	0.003	0.003	0.008	0.005	0.006	0.001
ROC (TPR)	0.807	0.802	0.731	0.049	0.044	0.676	0.950
	0.002	0.002	0.002	0.004	0.002	0.002	0.001
TO (EO)	0.786	0.784	0.714	0.006	0.004	0.709	0.951
	0.003	0.003	0.003	0.005	0.003	0.004	0.001
TO (SP)	0.765	0.797	0.692	0.123	0.091	0.997	0.949
	0.002	0.002	0.002	0.005	0.003	0.007	0.001
TO (TPR)	0.785	0.784	0.713	0.006	0.004	0.705	0.951
	0.003	0.003	0.003	0.005	0.003	0.009	0.001

TABLE 5.6: Results of the experiments on the Hospital dataset. Shows the mean and standard deviation over 10 runs.

5.4.2 Performance of FairAST

The results of the recommendations made by FairAST for the three different fairness measures are presented in Table 5.7. Here, the recommendations are compared to the results obtained in the experiments on the algorithms. For each dataset/fairness measure combination, the top three algorithms are selected. It is important to note that in case an algorithm is able to optimize for different fairness measures, only the results of the specified fairness measure are included in the list of best performing algorithms. So, for example, if TO (SP) and TO (EO) would actually give the best results for optimizing EO, only TO (EO) will be included in the top three. This seemed fair, because the recommendation tool can only run an algorithm for one fairness measure.

The results in the table are colour-coded to help interpreting the results. The three best performing algorithms from the previous experiments are considered to be the actual top three algorithms. These are presented in the left column and coloured from dark gray (first) to light gray (third). The results show the average score over 10 runs on the specified fairness measure. For the algorithms recommended by FairAST, any algorithm matching the top three received a corresponding color. For the recommendations made by FairAST, the results over a single run on the entire dataset are shown.

The results show us that for all 18 recommendations, the number one recommendation corresponds to at least one of the top three best algorithms. For the experiments on a single sample run, 5 of the number one recommendations were the same as the actual number one from the top three. The other 4 number one recommendations all corresponded to the third best algorithm. In the experiments with three sample runs, the number one recommendation corresponds 6 times to the number one algorithm. Once, the recommendation corresponds to the second best, and twice to the third best algorithm. Comparing the two settings (one or three sample runs) thus shows that the recommendations made after three sample runs are slightly better, since the results better match the top three. This is in line with the expectations, because after multiple runs the results level out any single run deviations. Yet, some algorithms are recommended that are not in the top three. The size of the sample plays a role in this problem.

The sample size used in FairAST was chosen to be 5000, of which 3500 instances are used for training the algorithms. The bigger a sample, the more the results will resemble the results obtained on the entire dataset, but it will also take more training time. Looking at the sizes of the datasets (Compas 7185, Adult 30162, Hospital 528820), using only 3500 instances for training, will mostly benefit the training time for Adult and Hospital. For Compas, this benefit is a bit smaller, since Compas already consists of less instances. Looking at the accuracy of the results, we do not see big differences between the datasets, indicating that the chosen sample size works as well for the small as for the bigger datasets. We do see differences for the fairness measures, since the recommendations for SP are much more in line with the top three than the recommendations for EO and TPR. For SP, the sample might be more equal to the entire dataset, since stratification is used to ensure the same disparate impact ratio in the sample as in the complete dataset. Therefore, the algorithms will have to perform very similar operations to satisfy SP. For EO and TPR, the results depend on the number of true and false positives. Here, it might be the case that for some algorithms it might be easier to optimize the EO and TPR for this sample, then for the entire dataset. Using a smaller sample size will therefore probably not benefit the accuracy of the recommendations for EO and TPR.

The running time of FairAST is not only influenced by the sample size, but of course also by the number of algorithms to test. The given datasets (with only a single binary protected attribute) and settings of FairAST (allowing to optimize for compatible fairness measures and all types of pre-, in- and post-processing) allow most algorithms to work on these datasets. Only GFC, LFR and OP are excluded for all recommendations, and CEOP is excluded for optimizing EO. This means that in these experiments, 12 or 13 algorithms were considered suitable algorithms. Using FairAST instead of performing an exhaustive search thus prevents the user from testing algorithms that are not going to perform well (or not at all) on their dataset, and it enables the user to quickly check the performance of all suitable algorithms, without the need to figure out how each algorithm should be implemented.

Overall, the performance of FairAST can be considered fairly accurate, as all final recommendations correspond to one of the best performing algorithms found through exhaustive search. Also, the recommended algorithms that were not the number one best algorithm, still improve the desired fairness measure. This confirms that the recommended algorithms are suitable for the given dataset. For better results, it is advisable to use FairAST with multiple sample runs instead of one, to level out the performance over multiple runs, and to perform a final run on at least the top three algorithms, to see how they perform on the entire dataset.

	TOP THREE	SP	FairAST (1)	SP	FairAST (3)	SP
Compas	TO (SP)	1.001	Rew	0.998	Rew	0.998
	EGR (SP)	0.990	EGR	0.975	EGR	0.976
	Rew	1.036	TO	0.974	TO	0.974
Adult	TO (SP)	0.972	TO	0.976	TO	0.976
	ROC (SP)	0.896	GSR	1.101	ROC	0.935
	EGR (SP)	0.848	CR	0.876	CR	0.876
Hospital	TO (SP)	0.997	TO	1.004	TO	1.004
	EGR (SP)	0.922	EGR	0.926	EGR	0.927
	CR	0.894	CR	0.877	CR	0.877

	TOP THREE	EO	FairAST (1)	EO	FairAST (3)	EO
Compas	EOP	0.021	TO	0.008	EGR	0.018
	EGR (EO)	0.023	EOP	0.030	EOP	0.022
	TO (EO)	0.026	AD	0.178	ROC	0.068
Adult	EOP	0.010	EOP	0.016	Rew	0.009
	GSR (EO)	0.012	GSR	0.033	EOP	0.013
	Rew	0.015	CR	0.049	CR	0.049
Hospital	EOP	0.003	Rew	0.002	EOP	0.004
	TO (EO)	0.004	GSR	0.022	GSR	0.022
	Rew	0.005	CR	0.057	CR	0.057

	TOP THREE	TPR	FairAST (1)	TPR	FairAST (3)	TPR
Compas	EOP	0.018	EOP	0.014	EOP	0.002
	EGR (TPR)	0.022	ROC	0.052	TO	0.007
	TO (TPR)	0.022	Rew	0.054	ROC	0.052
Adult	EOP	0.016	EOP	0.018	EOP	0.023
	GSR (TPR)	0.020	ROC	0.026	ROC	0.026
	Rew	0.025	CR	0.058	CR	0.058
Hospital	EOP	0.006	Rew	0.000	EOP	0.005
	TO (TPR)	0.006	MFC	0.016	MFC	0.052
	Rew	0.007	CR	0.076	CR	0.076

TABLE 5.7: The recommendations of FairAST after one (1) or three (3) sample runs, compared to the best baseline results.

Chapter 6

Conclusion

The final chapter consist of a summary and the answers to the research questions of this thesis. Furthermore, the limitations and directions for future work are discussed, and the final ethical considerations are presented.

6.1 Summary

In this thesis we have investigated the problem of how to decide which bias mitigation technique can best be applied to improve the fairness on a dataset. We compared sixteen different bias mitigation algorithms and showed that data profiling techniques provide useful insights in the data that contribute to the selection of the best algorithm to use.

Data profiling techniques showed to be important for different purposes. First, inspecting the data and gathering statistics is important to properly clean and preprocess the data. The results are used to identify sensitive features and biases in the data. Secondly, the characteristics of the data tell which bias mitigation algorithms are suitable to use on the dataset.

The analysis of the algorithms provided a clear overview of the requirements and capabilities of each of these techniques. Although many algorithms are presented and tested in the literature, a comparative study on all these algorithms has not been performed before. Testing the algorithms on the same datasets, for the same protected attributes and fairness measures, enabled an equal comparison of their effects on fairness.

Furthermore, the recommendation tool FairAST was presented, which automates the process of selecting the best bias mitigation algorithm by partially pre-processing the data, creating a profile of the data, and testing the suitable algorithms. The tool has proven to give fairly trustworthy recommendations, and can truly speed up the process of selecting the algorithm that will improve fairness the most.

By providing this recommendation tool, we achieved the goal of creating a framework that will help a user to perform data profiling, helps to discover biases, choose an algorithm and create fair outcomes. In the following section we will discuss the answers to the research questions that were posed in this thesis.

6.2 Answer to research questions

In section 1.3, the research questions of this thesis project were stated. We will start by formulating the answers to the subquestions, before turning to the main research question.

Subquestions:

1. How does the choice of fairness measure influence the choice of bias mitigation algorithm?

   We investigated different fairness measures and the (im)possibility to satisfy multiple fairness measures simultaneously. After all, whether a strategy is the 'optimal' bias mitigation strategy, depends on the measure that is chosen to assess the fairness. Although algorithms are built to satisfy some specific measure(s), we found in section 5.4.1 that not only the specified measure, but also compatible measures can be optimized simultaneously. The results show that it is beneficial to not only focus on algorithms that specifically optimize for the desired fairness measure, because optimizing for compatible measures can also improve the desired measure. However, algorithms that optimize for an incompatible measure can be discarded, since these will not be able to improve both measures. So, the choice of a bias mitigation algorithm is influenced by the compatibility of a fairness measure with the optimization goals of the algorithm.

2. How do different bias mitigation algorithms perform compared to each other?

   In section 5.4.1 the results were presented that showed how each bias mitigation algorithm performed for multiple fairness measures on three different datasets. These results provided insight in how the algorithms perform compared to each other.

3. To what extent are the bias mitigation algorithms robust enough to ensure fairness for any given dataset?

   By analyzing the inner workings of the algorithms in section 3.4, we found that some algorithms have very specific requirements for the datasets in order to function properly. This shows that the algorithms are not capable of mitigating biases on any dataset. The most robust algorithms are from the post-processing technique, because these algorithms are mostly based on the actual and predicted labels, and not so much on the features in the data.

4. Which statistics obtained by data profiling are important for the selection of a bias mitigation technique?

   The analysis of the algorithms in 3.4 provided several requirements, that show which properties of a dataset are important for selecting a technique. The type of label, the label balance, the type of features, the number of protected attributes and (un)privileged groups all showed to be important. Furthermore, the number of unique feature values for all features and for the protected attributes are characteristics that need to be inspected. All these properties can be identified through data profiling techniques.

The main research question was: how can data profiling techniques contribute to selecting the optimal bias mitigation strategy for any given dataset? By answering the subquestions, we also find the answer to the main question.

We have shown that data profiling techniques provide useful information about the dataset that help to find a suitable bias mitigation algorithm. Data profiling helps in the pre-processing phase by finding the biases in the data, so the user can select the protected attributes and specify the groups in the data. Moreover, data profiling techniques gathering the counts and data types of the features are of great importance for selecting a mitigation technique. However, not all selection criteria can be obtained purely from the data. For the fairness measure, the user will have to decide which measure is most suitable for the given dataset.

6.3 Limitations and future work

There are a couple of improvements that could be made to the provided recommendation tool FairAST. First of all, it is a limitation that the tool only works for binary classification problems. Although most of the algorithms are also meant for binary classifications, there are techniques that allow for multi-class classification or regression problems. However, the existing fairness measures are only suitable for binary classifications problems. Future research could investigate the possibilities for fairness on other types of ML problems. FairAST could then be improved by including these different types of algorithms. In general, adding more bias mitigation algorithms or allowed fairness measures will increase the usability of the tool.

Furthermore, the comparative performance results of the bias mitigation algorithms are based on datasets with a single binary protected attribute. For the sake of good comparability, we wanted to be able to test the datasets on as many of the algorithms as possible, which was with a single binary protected attribute. A lot of previous fairness research also concentrates on a single binary protected attribute. However, since several algorithms should be able to handle multiple or non-binary protected attributes, it would be interesting to analyze the performance and improvements regarding the fairness on such datasets too. Future work could perform experiments on the algorithms with multiple protected attributes, and investigate to what extent the algorithms are capable of enhancing fairness for multiple attributes together. In fact, the tool could be of help for this, since it already enables the user to execute the algorithms on datasets with multiple protected attributes.

Moreover, performing experiments on a more wide variety of datasets could help to find interesting results on the performance of the algorithms, that could provide more insight into which algorithms will perform best on a given dataset. Additional requirements based on these findings could reduce the number of suitable algorithms that will be tested in FairAST. This will reduce the running time of FairAST.

6.4 Ethical considerations

Research into the fairness of ML algorithms is of great importance for the trustworthiness of algorithmic decisions. However, we have seen that fairness is a complex concept, that differs per person, or per problem. There are multiple mathematical definitions established, but these notions do not capture all aspects of fairness. Which means that even if a fairness measure is satisfied, we can still wonder whether the outcomes are completely fair. For example, when an algorithm aims at SP, for some people in the privileged group the favorable outcome might be changed in an unfavorable outcome. Depending on the situation, it could be argued that on an individual level, this might be unfair for this person. Also, the method used can be considered fair or unfair. For example, whether the changes made in the prediction to create fairness are based on the certainty of the prediction, or just randomly. Therefore, it is always important to consider the implications of using bias mitigation algorithms on a dataset, especially when it will be used in practice.

Finally, it is important to keep in mind that the biases shown in the data distributions are not the only existing biases, as discussed previously. The way the data is collected, but also the way a model is used, can perpetuate or generate biases. Therefore, taking a holistic perspective on fairness is essential for employing ML models in the real world.

Appendix A

Additional experiments

Additional experiments have been performed to show the performance of LFR, OP and GFC on datasets that are suitable for these algorithms.

A.1 OP

For OP, we used the Compas dataset as preprocessed in the AIF360 library. Now there are only a small number of categorical features: 'race', 'sex', age categorized in <25, 25-45, >45, the priors categorized in 0, 1-3, >3, and the charge degree being either F or M. 'Race' is chosen as the protected attribute. The label 'two year recid' represents whether someone will reoffend within two years.

Using the distortion and optimization functions specified in the helper functions of OP, we transform the data and run a logistic regression classifier. The results now show that the algorithm works properly and the disparate impact is improved to 0.884.

A.2 LFR

For LFR, we used the Compas dataset without the feature 'c_charge_desc', since this feature has many different unique feature values. Setting the number of prototypes to 10, and Ax, Ay, Az respectively to 0.1, 10 and 10, provided the results presented in Table A.1.

LFR	Acc p	Acc up	F1	TPR	EO	SP	C
mean	0.604	0.602	0.716	0.056	0.056	1.025	0.984694
std	0.022	0.026	0.015	0.045	0.042	0.062	0.002

TABLE A.1: Results for the Compas dataset.

We also tested the Adult dataset. Only removing the feature 'native country', because this feature has many unique feature values, was not enough to make the algorithm function properly. The Adult dataset is imbalanced regarding the class labels, therefore, the LFR learns to predict the majority class in order to minimize its error. By downsampling the number of unfavorable labels, we find that the algorithm succeeds finding both favorable and unfavorable predictions. For the privileged group, we randomly chose 3000 favorable, and 1400 unfavorable instances. For the unprivileged group, we chose 1400 favorable, and 3000 unfavorable instances. By doing so, the imbalance between the privileged and the unprivileged group is preserved, but the distribution of class labels in the dataset is equal. Now, the LFR was able to make more accurate predictions. The results are shown in Table A.2

LFR	Acc p	Acc up	F1	TPR	EO	SP	C
mean	0.701	0.743	0.780	0.280	0.321	0.353	0.964
std	0.015	0.067	0.052	0.122	0.072	0.102	0.007

TABLE A.2: Results for the Adult dataset.

A.3 GFC

For GFC, we used the Adult dataset and considered ′race′ and ′sex′ as two separate features, where sex is a binary feature, and race can take 5 different values. With these settings, there are 10 subgroups in the data. The results are now more accurate and consistent than for the experiments on the single binary protected attribute, as can be seen in Table A.3.

GFC	Acc p	Acc up	F1	TPR	EO	SP	C
mean	0.726	0.696	0.610	0.126	0.096	0.302	0.897
std	0.005	0.007	0.008	0.014	0.007	0.012	0.003

TABLE A.3: Results for the Adult dataset with two protected attributes.

Appendix B

Additional results GFC

Run	Acc p	Acc up	Precision	Recall	TPR	EO	SP	C
0	0.501	0.499	0.550	0.970	0.016	0.014	0.986	0.976
1	0.502	0.502	0.551	0.974	0.003	0.003	1.003	0.981
2	0.490	0.493	0.546	0.965	0.016	0.019	0.984	0.980
3	0.565	0.521	0.814	0.148	0.131	0.087	0.275	0.947
4	0.549	0.538	0.725	0.189	0.091	0.079	0.486	0.931
5	0.568	0.521	0.812	0.157	0.156	0.109	0.206	0.943
6	0.505	0.494	0.550	0.966	0.015	0.010	0.995	0.973
7	0.553	0.519	0.812	0.127	0.122	0.088	0.212	0.943
8	0.499	0.510	0.552	0.962	0.007	0.012	0.994	0.971
9	0.502	0.499	0.550	0.999	0.002	0.003	1.001	0.999

TABLE B.1: Single run results on the Compas dataset for GFC with a single binary protected attribute. The precision and recall scores show that mostly one class is predicted.

Run	Acc p	Acc up	Precision	Recall	TPR	EO	SP	C
0	0.660	0.623	0.811	0.340	0.107	0.071	0.258	0.968
1	0.666	0.639	0.811	0.359	0.093	0.067	0.260	0.968
2	0.660	0.606	0.858	0.324	0.130	0.076	0.245	0.971
3	0.659	0.609	0.815	0.332	0.132	0.081	0.246	0.971
4	0.670	0.601	0.842	0.343	0.168	0.098	0.209	0.972
5	0.500	0.500	0.249	1.000	0.000	0.000	1.000	1.000
6	0.665	0.629	0.822	0.351	0.106	0.070	0.260	0.969
7	0.664	0.619	0.839	0.339	0.117	0.073	0.252	0.971
8	0.500	0.500	0.249	1.000	0.000	0.000	1.000	1.000
9	0.500	0.500	0.249	1.000	0.000	0.000	1.000	1.000

TABLE B.2: Single run results on the Adult dataset for GFC with a single binary protected attribute. The precision and recall scores show that in some runs only one class is predicted.

Appendix C

Fairness through unawareness

	Method	Acc p	Acc up	F1	TPR	EO	SP	C
Compas	PR	0.634	0.668	0.731	0.161	0.195	0.679	0.843
		0.011	0.019	0.009	0.016	0.012	0.015	0.005
	Unaware	0.635	0.668	0.732	0.160	0.194	0.680	0.843
		0.010	0.018	0.008	0.014	0.013	0.017	0.005
Adult	PR	0.667	0.644	0.498	0.075	0.052	0.420	0.927
		0.007	0.009	0.010	0.023	0.010	0.019	0.002
	Unaware	0.667	0.644	0.499	0.075	0.052	0.420	0.927
		0.007	0.009	0.010	0.022	0.010	0.019	0.002
Hospital	PR	0.795	0.761	0.717	0.114	0.080	0.521	0.951
		0.002	0.003	0.003	0.006	0.003	0.005	0.001
	Unaware	0.795	0.762	0.718	0.113	0.080	0.523	0.954
		0.002	0.002	0.002	0.005	0.003	0.004	0.000

TABLE C.1: Results for the PR and a logistic regression classifier trained without the protected attributes (Unaware).

Bibliography

[1] Ziawasch Abedjan, Lukasz Golab, and Felix Naumann. "Data profiling: A tutorial". In: *Proceedings of the 2017 ACM International Conference on Management of Data (SIGMOD'17)*. 2017, pp. 1747–1751.

[2] Alekh Agarwal et al. "A reductions approach to fair classification". In: *International Conference on Machine Learning (ICML'18)*. 2018, pp. 60–69.

[3] *API reference*. URL: https://scikit-learn.org/stable/modules/classes.html#module-sklearn.metrics.

[4] Otmane Azeroual, Gunter Saake, and Eike Schallehn. "Analyzing data quality issues in research information systems via data profiling". In: *International Journal of Information Management* 41 (2018), pp. 50–56.

[5] Solon Barocas, Moritz Hardt, and Arvind Narayanan. "Fairness in machine learning". In: *Nips tutorial* 1 (2017), p. 2.

[6] Reuben Binns. "Fairness in machine learning: Lessons from political philosophy". In: *Conference on Fairness, Accountability and Transparency*. PMLR. 2018, pp. 149–159.

[7] Sarah Bird et al. *Fairlearn: A toolkit for assessing and improving fairness in AI*. Tech. rep. MSR-TR-2020-32. Microsoft, 2020. URL: https://www.microsoft.com/en-us/research/publication/fairlearn-a-toolkit-for-assessing-and-improving-fairness-in-ai/.

[8] Flavio P Calmon et al. "Optimized pre-processing for discrimination prevention". In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. 2017, pp. 3995–4004.

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

[10] L Elisa Celis et al. "Classification with fairness constraints: A meta-algorithm with provable guarantees". In: *Proceedings of the conference on fairness, accountability, and transparency*. 2019, pp. 319–328.

[11] Alexandra Chouldechova. "Fair prediction with disparate impact: A study of bias in recidivism prediction instruments". In: *Big data* 5.2 (2017), pp. 153–163.

[12] Jeffrey Dastin. *Amazon scraps secret AI recruiting tool that showed bias against women*. 2018. URL: https://www.reuters.com/article/us-amazon-com-jobs-automation-insight/amazon-scraps-secret-ai-recruiting-tool-that-showed-bias-against-women-idUSKCN1MK08G.

[13] Dheeru Dua and Casey Graff. *UCI Machine Learning Repository*. 2017. URL: http://archive.ics.uci.edu/ml.

[14] Cynthia Dwork et al. "Fairness through awareness". In: *Proceedings of the 3rd innovations in theoretical computer science conference*. 2012, pp. 214–226.

[15] Michael Feldman et al. “Certifying and removing disparate impact”. In: *proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining (KDD’15)*. 2015, pp. 259–268.

[16] Sorelle A Friedler et al. “A comparative study of fairness-enhancing interventions in machine learning”. In: *Proceedings of the conference on fairness, accountability, and transparency*. 2019, pp. 329–338.

[17] Batya Friedman and Helen Nissenbaum. “Bias in computer systems”. In: *ACM Transactions on Information Systems (TOIS)* 14.3 (1996), pp. 330–347.

[18] Pratyush Garg, John Villasenor, and Virginia Foggo. “Fairness metrics: A comparative analysis”. In: *2020 IEEE International Conference on Big Data (Big Data)*. IEEE. 2020, pp. 3662–3666.

[19] *General Data Protection Regulation*. The European Parliament and the Council of the European Union, May 25, 2018. URL: https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32016R0679 (visited on 03/03/2021).

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

[21] Hoda Heidari et al. “A moral framework for understanding fair ml through economic models of equality of opportunity”. In: *Proceedings of the conference on fairness, accountability, and transparency*. 2019, pp. 181–190.

[22] Woo Suk Hong, Adrian Daniel Haimovich, and R Andrew Taylor. “Predicting hospital admission at emergency department triage using machine learning”. In: *PloS one* 13.7 (2018), e0201016.

[23] Knut T Hufthammer et al. “Bias mitigation with AIF360: A comparative study”. In: *Norsk IKT-konferanse for forskning og utdanning*. 1. 2020.

[24] Faisal Kamiran and Toon Calders. “Data preprocessing techniques for classification without discrimination”. In: *Knowledge and Information Systems* 33.1 (2012), pp. 1–33.

[25] Faisal Kamiran, Asim Karim, and Xiangliang Zhang. “Decision theory for discrimination aware classification”. In: *2012 IEEE 12th International Conference on Data Mining (ICDM’12)*. IEEE. 2012, pp. 924–929.

[26] Faisal Kamiran and Indrė Žliobaitė. “Explainable and non-explainable discrimination in classification”. In: *Discrimination and Privacy in the Information Society*. Springer, 2013, pp. 155–170.

[27] Toshihiro Kamishima et al. “Fairness-aware classifier with prejudice remover regularizer”. In: *Joint European conference on machine learning and knowledge discovery in databases*. Springer. 2012, pp. 35–50.

[28] Michael Kearns et al. “Preventing fairness gerrymandering: Auditing and learning for subgroup fairness”. In: *International Conference on Machine Learning (ICML’18)*. 2018, pp. 2564–2572.

[29] Jon Kleinberg, Sendhil Mullainathan, and Manish Raghavan. “Inherent tradeoffs in the fair determination of risk scores”. In: *arXiv preprint arXiv:1609.05807* (2016).

[30] Bruno Lepri et al. “Fair, transparent, and accountable algorithmic decision-making processes”. In: *Philosophy & Technology* 31.4 (2018), pp. 611–627.

[31] Ninareh Mehrabi et al. “A survey on bias and fairness in machine learning”. In: *ACM Computing Surveys (CSUR)* 54.6 (2021), pp. 1–35.

[32] Felix Naumann. "Data profiling revisited". In: *ACM SIGMOD Record* 42.4 (2014), pp. 40–49.

[33] Geoff Pleiss et al. "On fairness and calibration". In: *Advances in neural information processing systems* 30 (2017).

[34] ProPublica. *Compas recidivism risk score data and analysis*. URL: `https://www.propublica.org/datastore/dataset/compas-recidivism-risk-score-data-and-analysis`.

[35] Babak Salimi et al. "Interventional fairness: Causal database repair for algorithmic fairness". In: (2019), pp. 793–810.

[36] Robik Shrestha, Kushal Kafle, and Christopher Kanan. "An investigation of critical issues in bias mitigation techniques". In: (2022), pp. 1943–1954.

[37] Megha Srivastava, Hoda Heidari, and Andreas Krause. "Mathematical notions vs. human perception of fairness: A descriptive approach to fairness for machine learning". In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD'19)*. 2019, pp. 2459–2468.

[38] Harini Suresh and John V Guttag. "A framework for understanding unintended consequences of machine learning". In: *arXiv preprint arXiv:1901.10002* 2 (2019).

[39] Sahil Verma and Julia Rubin. "Fairness definitions explained". In: *2018 ieee/acm international workshop on software fairness (fairware)*. IEEE. 2018, pp. 1–7.

[40] Muhammad Bilal Zafar et al. "Fairness constraints: Mechanisms for fair classification". In: *Artificial Intelligence and Statistics*. 2017, pp. 962–970.

[41] Rich Zemel et al. "Learning Fair Representations". In: *Proceedings of the 30th International Conference on Machine Learning (ICML'13)*. 2013, pp. 325–333.

[42] 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, pp. 335–340.

[43] Xingyu Zhang et al. "Trends of racial/ethnic differences in emergency department care outcomes among adults in the United States from 2005 to 2016". In: *Frontiers in medicine* 7 (2020), p. 300.