UCLouvain

École polytechnique de Louvain

Increasing fairness in supervised classification: a study of simple decorrelation methods applied to the logistic regression model.

Author: **Constantin DE SCHAETZEN**
Supervisor: **Marco SAERENS**
Readers: **Jean-Charles DELVENNE, Pierre LELEUX, Marco SAERENS**
Academic year 2020–2021
Master [120] in Mathematical Engineering

Abstract

Nowadays, classifications algorithms perform tasks such as filtering college and loan applications or assessing the risk that an inmate reoffend when released from prison. As our society becomes more and more driven by data and as machine learning takes increasingly more place in everyday life's decision making processes, it becomes urgent that we find classification algorithms that ensure fairness and equity between individuals. For instance, unfairness or discrimination may happen in classification when the data is generated by a biased decision process. In such cases, classification models could not only repeat the bias but also introduce new ones. To tackle this issue, we look for accurate models for which the predictions are uncorrelated with a protected sensible attribute (e.g. race, gender,...). In particular, we propose decorrelation methods operating before, during and after the learning phase of classification models. We show that our methods are able to remove the correlation between the sensible attribute and the predictions while maintaining a high level of accuracy. By limiting the biases present in the predictions made by the classification algorithms, we reinforce the equality between individuals.

Acknowledgements

I would like to thank Prof. Marco Saerens, my thesis supervisor, who helped me a lot in writing this paper. His advice and guidance were very helpful.

Contents

Chapter 1

Introduction

Machine learning is one of the most popular field of computer science and mathematics nowadays. As our society becomes more and more driven by data and as data collection is now a billion dollar market, research on this subject is constantly evolving bringing new questions and new considerations to the table. One of the most common task in machine learning is classification. Given a certain number of parameters observed on a sample, one would like to be able to assign a category to it. This is usually done by training a model on a data set and then using the knowledge that the model has learned to be able to make predictions on new samples.

In particular, supervised classification is very popular. A model is said to be based on supervised classification when it uses the category label of the training samples to acquire knowledge. In contrast, unsupervised classification will not use the labels but just other features that have been observed. There exists a tremendous number of applications using supervised learning in different domains such has finance, healthcare, advertisement, etc. For example, Google is developing artificial intelligence techniques to improve cancer detection [17]. To detect cancer, doctors often use X-ray screening. However, detecting a cancer based on this is quite a challenge and it is even more the case when the disease is still at an early stage. Unfortunately, this means that the diagnosis made by doctors can result in false negative. To lower the chance of false negative happening, google researchers rely on machine learning and classification algorithm to try to categorize X-ray screening in the positive or in the negative group. As [17] shows, in some cases Google's technology can detect breast cancer better than the radiologists.

Although machine learning has clear benefit, we can and should ask ourselves if we can always rely on data and on algorithms for any tasks. Over the last decade, machine learning has found new domains of applications such as filtering loan

applicants, deploying police officers in the streets of a city or even assessing the risk that a criminal reoffend a crime. In such areas, machine learning techniques raise important concerns for unfair treatment and potential discriminatory practices. For instance would you trust a method based on historical justice data to assess a prison sentence to be fair between individuals of different races? These concerns are supported by numerous research on the subject which leads to the need of algorithms that ensure equal treatment for all [3, 5, 18].

In the last 10 years, people have been talking more and more about fairness and what was originally a niche topic of research is now an important subject in machine learning. Despite of this, the task is still quite challenging as it is at the boundary of different fields such as computer science, mathematics, statistics but also ethics and morality. Moreover, it brings a lot of questions. What does it mean not to discriminate? Where does unfairness find its sources? How can we detect unfair treatment and how should we adapt our methods to tackle this issue? At what cost ?

A straightforward idea that one can think of when considering fairness problems is to simply remove a sensible variable (e.g. race or gender) from the data set. It turns out that this solution would not work as it is very likely that the sensible variable and some other attributes in the data set are not independent [8, 29, 30]. The methods found in the literature concerning fairness can be grouped into three categories. Methods that attempt to modify or change the representation of the data (pre-processing), methods that try to impose equal treatment during the learning phase and methods that try to modify the predictions a posteriori to match some fairness criteria (post-processing). In the vast majority of cases, improving fairness will result in a degradation of the classification performances. We will therefore need to find a trade-off between the two [23].

In this master thesis, attempt to mitigate bias with all three kinds of methods have been imagined. Pre-processing approaches has been imagined through projections to remove the linear correlation between a protected group and other data that will be used during training. Approaches that operate during the learning phase were based on an optimisation problem known as maximum entropy (MaxEnt). Finally, post-processing approaches using least square optimization combined with fairness constraints have been implemented to try to mitigate bias. The methods that were tried are non exclusive, they can be combined and one objective of this master thesis will be to investigate whether or not a method working on "more than one level" of the process can give interesting results.

The objective of this master thesis are thus the following :

- *Does the maximum entropy formulation allows to easily introduce fairness constraint ?*
- *Can we show empirically the equivalence between logistic regression and maximum entropy?*
- *What method offer the best trade-off between fairness and classification performances ?*
- *Are pre-processing methods able to fully decorrelate predictions with sensitive variables ?*
- *Can post-processing achieve a good trade-off between classification performance and fairness? How does supervised method compare with semi-supervised approaches ?*
- *Does a combination of pre-processing, post-processing and learning methods improve the results?*
- *How do our methods compare with approaches from literature ?*

This master thesis will be divider into two parts. The first one will be theoretical (Chapter 1 & 2), it will consist of a state of the art and some basic theory (notably about fairness) and then in some theory about maximum entropy and about the pre- and post-processing techniques that we will apply later on (Chapter 3 & 4). The second part (Chapter 5,6 & 7) of the thesis will cover the experimental methodology, the empirical evaluation of our models, a discussion about the results and an assessment of the work that could be further developed based on our findings.

Chapter 2

Theoretical basis

As introductory example, suppose that we want to implement a classification method to determine whether a certain person is a serious candidate for employment in a company. To this end, the company built a dataset over the years by taking note of many parameters about the people who applied for the job and whether or not these people were good candidates. With this dataset, the company implemented a machine learning-based classifier. It turns out that the data is biased: in recent years, the company has favoured white men over other genders and races. The model that learned to classify based on historical data learned to reproduce this discrimination. What we need is to find a way for the model to learn not to reproduce this discrimination while maintaining some kind of accuracy.

In particular, the problem of fairness in classification algorithms arises from a dependence that is considered negative between certain attributes observed in the dataset and the target class to which the observations belong. The dependency is deemed negative because the decision process that determines the class is biased and the attributes are attributes that are deemed sensitive (e.g. gender, race,...). The dependency will have an undesirable effect on the predictions because they will reproduce the bias present in the dataset. Not only can the classifier reproduce biases but it can also introduce new ones [5, 6]. A straightforward solution would be to simply remove the sensitive attributes from the dataset. It turns out that this solution is generally not sufficient to make the classifier unbiased [8, 29, 30]. Indeed, other attributes might themselves be related in some way to the sensitive attributes. For example, it is reasonable to think that home address and race may not be independent. Removing race from the dataset would therefore generally not

be sufficient to remove discrimination. This phenomenon is called *redlining* and originated in the United States where banks coloured maps in different colours [31], with red representing areas where they would not invest. This colouring was done without direct observation of race, which appeared to be race-neutral but was not, as the bank was observing attributes correlated with race. In addition, removing the sensitive attribute as well as the attributes that are correlated with it usually does not work either, because either too much dependency remains or the accuracy decreases too much [8].

2.1 Sources of unfairness

Before going further and defining mathematically what unfairness is, let's look at the different sources that can lead a classifier to make discriminatory decisions.

bias encoded in the data set A first cause of unfairness is a bias encoded in the data. It is indeed very common to find human biases in the training data. In the example above, the company has hired white men in priority and therefore this population will often be favoured by a classifier that has been trained to fit the data. As there is no incentive to remove the biases, machine learning techniques will just replicate them in order to minimize the error [22, 23].

unbalanced data set A second phenomenon that can lead to unfairness is when there is a majority and a minority population. If the classifier does not know which individual belongs to which group and if it cannot fit both population at the same time in an optimal way, then, by minimizing the overall error, the classifier will fit the majority population. The distribution of the error will be different for the two groups as it will be higher for the minority group. [6, 7]

need to explore A third scenario that can potentially lead to unfairness is the so-called "need to explore" [23]. In many problems, the data available depends on a decision the algorithm has made in the past. Suppose the goal of our classifier is to determine whether a prisoner will be a recidivist if released from prison. The dataset we have can only indicate that a criminal is a recidivist if he has been released before. In this type of problem, for the learning phase to be effective, sub-optimal decisions should be made in order to have more varied data. Of course, ethical issues emerge from such problems and sometimes prevent exploration.

This is non-exhaustive and there are other sources of unfairness.

2.2 Problem definition

To give rigorous definitions of fairness, let us first define the classification problem. In this thesis we will focus on binary classification, i.e. the target variable will have two different values. Suppose we have at our disposal n observations with m features in a $n \times m$ data matrix X.

$$X = [x^1, x^2, \ldots, x^m] \tag{2.1}$$

where x^j represents the observations of feature j. In addition we have p sensible variables (e.g. gender, race,...) in a $n \times p$ data matrix Z.

$$Z = [z^1, z^2, \ldots, z^p] \tag{2.2}$$

Our goal is to predict target variable Y which is binary and such that $\text{dom}(Y) = \{+, -\}$.

2.3 Definitions of fairness

Literature most often relies on a fairness definition that can be categorized in one of the three following groups : statistical definitions, individual definitions and causal definition [23]. At this point, there is not yet a consensus on which definition is the most appropriate and it seems that depending on the application one can be preferred over the other. This discussion is quite complex as it does not only include mathematical and statistical arguments but must also be subject to ethical considerations. What does fairness even mean ? Webster's dictionary gives the following definition [19].

> The quality of state of being fair. Lack of favoritism toward one side or another.

In this section, we briefly review the two first categories (statistical and individual), compare them and introduce the notion of fairness that will be used throughout this work.

2.3.1 Statistical measures

Statistical measures of fairness are the most used in literature. In general, these definitions select a number of protected variables and then impose a semblance of parity based on some statistical measure between the different protected attributes. The notion of statistical independence is the one that most often emerges from these discussions but there are many more (see [1]). For the sake of simplicity, we will restrain ourselves to a binary sensible variable encoded such that its domain is $\{0, 1\}$. We will see later that the definitions below can be easily extended to a case where there a several protected variables. Correspondingly, we will refer to the 0-class and the 1-class for the class.

Demographic or Statistical Parity (SP) A first notion based on this is the *demographic parity* of the prediction $\hat{Y}$ which relies on a measure called the *disparate impact* criterion (DI) [10, 22]

$$\text{DI} = \frac{\mathbb{P}(\hat{Y} = +|Z = 0)}{\mathbb{P}(\hat{Y} = +|Z = 1)} \tag{2.3}$$

This is also measured by the *discrimination score* which is defined as the difference $\mathbb{P}(\hat{Y} = +|Z = 0) - \mathbb{P}(\hat{Y} = +|Z = 1)$ in [33]. Demographic parity is achieved when DI is close to 1. This measure even has a legal value since it is part of the employment rules in the United States which require that it be no less than 0.8 [32]. The definition above considers that there is one sensible attribute Z that is binary (e.g. gender, race, ...). The generalization of this requires that $\hat{Y}$ and Z are independent, i.e. $\hat{Y} \perp\!\!\!\perp Z$.

Equal opportunity (EO) However this measure is considered not good enough for the following reason. A classifier that would assign 1 to the top 20% of the 0-class and randomly to 20% of the 1-class would be considered as fair when it is clear that it is not. For instance, suppose that a company hire people based on such a method. The best male candidate are preferred over the other male candidate. The best female candidate however, have the same chances that any other women. Would you consider this as fair treatment? To tackle this issue, the notion of *Equal opportunity (EO)* was introduced [10]. This requires that the True Positive Rates are equal across the protected groups and it is measured by the *Difference of Equal Opportunity* (DEO).

$$\text{DEO} = \mathbb{P}(\hat{Y} = +|Z = 1, Y = +) - \mathbb{P}(\hat{Y} = +|Z = 0, Y = +) \tag{2.4}$$

Equalized odds (EOdds) In addition to requiring EO, equalized odds require that False Positive Rates are equal across protected groups

$$\mathbb{P}(\hat{Y} = +|Z = 1, Y = -) - \mathbb{P}(\hat{Y} = +|Z = 0, Y = -) \tag{2.5}$$

In practice, we will never impose that one of these criteria be exactly zero (too restrictive), but rather that the model be what is called ϵ-fair [10]. A model is ϵ-fair if the fairness constraint it uses is violated by at most ϵ. For instance, a model using EO as fairness criteria will be ϵ-fair if the absolute value of DEO is less or equal to ϵ, i.e.

$$|\mathbb{P}(\hat{Y} = +|Z = 1, Y = +) - \mathbb{P}(\hat{Y} = +|Z = 0, Y = +)| \leq \epsilon \tag{2.6}$$

Statistical measures are easy to handle, to understand and to verify. On the other hand, even when satisfied, these fairness criterion won't on their own achieve extensive fairness in the sense that a particular individual or a particular subgroup of the protected group might still be discriminated. Those definitions give guarantees "on average" on the protected groups but not more than that. [2] extends on numerous cases where statistical measures fail to ensure fairness at an individual level.

2.3.2 Individual measure of fairness

To tackle this issue, some papers have focused on other kind of fairness criterion that are more centered on individuals. For instance, [2] works with a task-specific metric and used a criterion that ensures that "similar people should be considered similarly by the classifier". Another example is [3] where researchers have worked with the idea that "less qualified people should not be advantaged over better qualified people". Compared to statistical approaches, these criteria are more difficult to manage because they often require making assumptions that can be quite significant about the problem to be solved. For this reason, these criteria are less popular in the literature than statistical approaches, although they are still an important research topic.

2.3.3 Covariance as indicator of fairness

In this paper, we will use a slightly different yet straightforward measure for fairness but we will see that it can be reduced to a very similar form of the statistical measures discussed above. We will use the notion of covariance and correlation to establish the degree of fairness of a classifier. Mathematically, the sample covariance between n measurements of two variables X, Y on the elements of a sample set $\mathcal{T}$ is given by

$$\begin{aligned}
\text{cov}(\mathbf{x}, \mathbf{y}) &= \tfrac{1}{n-1} \sum_{i \in \mathcal{T}} (x_i - \bar{x})(y_i - \bar{y}) \\
&= \tfrac{1}{n-1} \sum_{i \in \mathcal{T}} [\mathbf{Hx}]_i [\mathbf{Hy}]_i \\
&= \tfrac{1}{n-1} (\mathbf{Hx}) \cdot (\mathbf{Hy}) \\
&= \tfrac{1}{n-1} (\mathbf{Hx})^\mathsf{T} (\mathbf{Hy}) \\
&= \tfrac{1}{n-1} \mathbf{x}^\mathsf{T} \mathbf{Hy}
\end{aligned} \tag{2.7}$$

where $\mathbf{H} = (\mathbf{I} - \frac{1}{n}\mathbf{e}\mathbf{e}^\mathsf{T})$ is the centering operator and $\mathbf{e}$ is a $n \times 1$ column vector containing 1's.
The variance is given by

$$\text{var}(\mathbf{x}) = \text{cov}(\mathbf{x}, \mathbf{x}) = \tfrac{1}{n-1} \|\mathbf{Hx}\|^2 \tag{2.8}$$

And finally the correlation is expressed by

$$\begin{aligned}
\text{cor}(\mathbf{x}, \mathbf{y}) &= \frac{\text{cov}(\mathbf{x}, \mathbf{y})}{\sqrt{\text{var}(\mathbf{x}) \text{var}(\mathbf{y})}} \\
&= \frac{\mathbf{Hx}}{\|\mathbf{Hx}\|} \cdot \frac{\mathbf{Hy}}{\|\mathbf{Hy}\|} \\
&= \cos(\mathbf{Hx}, \mathbf{Hy})
\end{aligned} \tag{2.9}$$

2.3.4 Covariance between a categorical variable and a numerical variable

Let us suppose that $Z \in \{0, 1\}$ is a binary discrete sensible variable and Y is a numerical target to be predicted. Accordingly, let $\mathbf{z}$ and $\mathbf{y}$ be two vectors of n realisations of those random variables. The class Z_0 will regroup indices of observations exhibiting $z = 0$ and Z_1 those that are such that $z = 1$. We will use $n_0 = |Z_0|$ and $n_1 = |Z_1|$ to refer to the number of instances of $\mathcal{T}$ that are in the class Z_0 and Z_1 respectively. Let us further denote by π_1 and π_0 the prior probabilities of the corresponding classes, i.e. $\mathbb{P}(x = 1) \simeq \pi_1 = \frac{n_1}{n}$ and $\mathbb{P}(x = 0) \simeq \pi_0 = \frac{n_0}{n}$. We

start with our previous definition of the sample covariance that we can decompose by regrouping the terms that belong to the same classes and then by noticing that $\bar{z} = \frac{1}{n}\sum_{i\in\mathcal{T}} z_i = \pi_1$

$$\begin{aligned}
\text{cov}(\mathbf{z},\mathbf{y}) &= \tfrac{1}{n-1}\sum_{i\in\mathcal{T}}(z_i-\bar{z})(y_i-\bar{y}) && (2.10)\\
&= \tfrac{1}{n-1}\sum_{i\in Z_1}(z_i-\bar{z})(y_i-\bar{y}) + \tfrac{1}{n-1}\sum_{i\in Z_0}(z_i-\bar{z})(y_i-\bar{y}) && (2.11)\\
&= \tfrac{1}{n-1}\sum_{i\in Z_1}(1-\bar{z})(y_i-\bar{y}) + \tfrac{1}{n-1}\sum_{i\in Z_0}(-\bar{z})(y_i-\bar{y}) && (2.12)\\
&= \tfrac{1}{n-1}\sum_{i\in Z_1}(1-\pi_1)(y_i-\bar{y}) + \tfrac{1}{n-1}\sum_{i\in Z_0}(-\pi_1)(y_i-\bar{y}) && (2.13)\\
&= \tfrac{1}{n-1}\sum_{i\in Z_1}(\pi_0)(y_i-\bar{y}) + \tfrac{1}{n-1}\sum_{i\in Z_0}(-\pi_1)(y_i-\bar{y}) && (2.14)\\
&= \pi_0\tfrac{n_1}{n-1}\tfrac{1}{n_1}\sum_{i\in Z_1}(y_i-\bar{y}) - \pi_1\tfrac{n_0}{n-1}\tfrac{1}{n_0}\sum_{i\in Z_0}(y_i-\bar{y}) && (2.15)\\
&= \tfrac{n}{n-1}\left(\pi_0\tfrac{n_1}{n}\tfrac{1}{n_1}\sum_{i\in Z_1}(y_i-\bar{y}) - \pi_1\tfrac{n_0}{n}\tfrac{1}{n_0}\sum_{i\in Z_0}(y_i-\bar{y})\right) && (2.16)\\
&= \tfrac{n}{n-1}\pi_0\pi_1\left(\tfrac{1}{n_1}\sum_{i\in Z_1}(y_i-\bar{y}) - \tfrac{1}{n_0}\sum_{i\in Z_0}(y_i-\bar{y})\right) && (2.17)
\end{aligned}$$

Now we can define $\Delta\bar{y}_1 = \frac{1}{n_1}\sum_{i\in Z_1}(y_i-\bar{y})$ and $\Delta\bar{y}_0 = \frac{1}{n_0}\sum_{i\in Z_0}(y_i-\bar{y})$. This represents the average excess or deficit in the numerical target with respect to the global average for each class of the sensible variable. Substituting those variables in the equation yields

$$\text{cov}(\mathbf{z},\mathbf{y}) = \tfrac{n}{n-1}\pi_0\pi_1(\Delta\bar{y}_1 - \Delta\bar{y}_0) \tag{2.18}$$

We can give the following interpretation. Suppose that the binary sensible variable Z represents the gender with $z=0$ for men and $z=1$ for women. The y numerical value represents the probability of getting hired for a job. If we wand similar treatment between men and women, it is intuitive to require $\Delta\bar{y}_1 = \Delta\bar{y}_0$, i.e. the two classes should have the same excess/deficit with respect to the global average. This is very similar to the *DI* measure that we introduced earlier in this chapter (and therefore also close to the discrimination score from [33]). Indeed we can write that $\text{cov}(\mathbf{z},\hat{\mathbf{y}}) \propto \hat{E}(Y|Z=1) - \hat{E}(Y|X=0)$ where $\hat{E}$ denotes the estimator of the mean which is the empirical expectation.

The result is similar if the binary sensible variable $Z \in \{-1,1\}$. We consider that Z_{1^-} is the class regrouping observations exhibiting $z=-1$ and correspond-

ingly Z_{1^+} for those such that $z = 1$. We have

$$\bar{z} = \frac{1}{n}\sum_{i\in\mathcal{T}} z_i \tag{2.19}$$
$$= \frac{1}{n}\sum_{i\in Z_{1^+}} z_i + \frac{1}{n}\sum_{i\in Z_{1^-}} z_i \tag{2.20}$$
$$= \frac{1}{n}n_{1^+} - \frac{1}{n}n_{1^-} \tag{2.21}$$
$$= \pi_{1^+} - \pi_{1^-} \tag{2.22}$$

And thus we can compute

$$\mathrm{cov}(\mathbf{z},\mathbf{y}) = \frac{1}{n}\sum_{i\in Z_{1^+}}(1-\bar{z})(y_i-\bar{y}) + \frac{1}{n}\sum_{i\in Z_{1^-}}(-1-\bar{z})(y_i-\bar{y}) \tag{2.23}$$
$$= \frac{1}{n}(1-\pi_{1^+}+\pi_{1^-})\sum_{i\in Z_{1^+}}(y_i-\bar{y}) - \frac{1}{n}(1+\pi_{1^+}-\pi_{1^-})\sum_{i\in Z_{1^-}}(y_i-\bar{y}) \tag{2.24}$$
$$= 2\pi_{1^-}\frac{n_{1^+}}{n}\frac{1}{n_{1^+}}\sum_{i\in Z_{1^+}}(y_i-\bar{y}) - 2\pi_{1^+}\frac{n_{1^-}}{n}\frac{1}{n_{1^-}}\sum_{i\in Z_{1^-}}(y_i-\bar{y}) \tag{2.25}$$
$$= 2\pi_{1^+}\pi_{1^-}(\Delta\bar{y}_{1^+} - \Delta\bar{y}_{1^-}) \tag{2.26}$$

We find the same requirement to equal treatment across population of the protected attribute $\Delta\bar{y}_{1^+} - \Delta\bar{y}_{1^-}$.

2.3.5 Covariance between two categorical variables

When both Z and Y are categorical variable, we find

$$\mathrm{cov}(\mathbf{z},\mathbf{y}) = \frac{1}{n-1}\sum_{i\in\mathcal{T}}(x_i-\bar{x})(y_i-\bar{y}) \tag{2.27}$$
$$= \sum_{i\in\mathcal{T}}(x_iy_i + \bar{x}\bar{y} - x_i\bar{y} - y_i\bar{x}) \tag{2.28}$$
$$= \sum_{i\in\mathcal{T}}(x_iy_i) + n\bar{x}\bar{y} - \bar{y}\sum_{i\in\mathcal{T}}x_i - \bar{x}\sum_{i\in\mathcal{T}}y_i \tag{2.29}$$
$$= \sum_{i\in\mathcal{T}}x_iy_i + n\bar{x}\bar{y} - 2n\bar{x}\bar{y} \tag{2.30}$$
$$= \frac{n}{n-1}\left(\frac{1}{n}\sum_{i\in\mathcal{T}}x_iy_i - \bar{x}\bar{y}\right) \tag{2.31}$$
$$= \frac{n}{n-1}(\pi_{xy} - \pi_x\pi_y) \tag{2.32}$$

where $\pi_{xy} = \sum_{i\in\mathcal{T}} x_iy_i$ is the proportion of observations for which both $x_i = 1$ and $y_i = 1$, π_x is the proportion of observations for which $x = 1$ and correspondingly

for π_y which is the proportion of observation that exhibit $y = 1$. Recall that one way of stating that two random variable X and Y are independent is by writing $\mathbb{P}[X \cap Y] = \mathbb{P}[X]\mathbb{P}[Y]$. Thus the covariance quantifies in some way the departure from independence between X and Y observed in the sample. For similar treatment, we find $\pi_{xy} = \pi_x \pi_y$. If we take the sample example as previously (job selection) this means that requiring the covariance to be zero is essentially saying that being a man/woman should be independent from being selected for the job.

2.4 Achieving fairness

Generally speaking, there are three main categories of methods that can be used to achieve fairness.

Pre-processing techniques The first category includes methods that work on the dataset and attempt to remove bias by preprocessing the data in order to learn unbiased representations of the data. A first group of methods works directly on the samples, for example by modifying them directly [8] or by adding observations while respecting the observed distribution [7]. Other preprocessing techniques use for instant projection and basis changes to remove some kind of dependency in the data set.

During learning The second category includes algorithms that attempt to force fairness during the learning phase. Those approaches rely solely on tuning the hyper parameters to obtain accurate and fair solutions. [9] is an example of such approach, achieving what they call a fair principal component analysis through multi-objective optimization where the aim is not to minimize the error and then to maximize a fairness criteria but rather to optimize both at the same time. [10] provides a framework that can handle different definitions of fairness at the same time and that can apply to a large family of machine learning techniques which is often not the case in other fairness techniques proposed in literature.

Post-processing techniques Finally, the last category is that of methods that work on predictions and attempt to correct the bias a posteriori. Those methods often work as black-box methods that adjust a classifier predictions in order to be ϵ-fair according to a certain fairness measure. A popular example of such approach is [13] where the idea is to optimally adapt an already fitted classifier using randomized positive discrimination. Another way to think about it is to imagine that a biased coin is flipped to apply a positive discrimination in order

to reduce the bias produced by the classifier. The advantages of post-processing techniques is that they are black box methods which means that we can use it with any classifier that already exist without the need of adapting algorithms. The main problem with post-processing techniques is that they are inherently sub-optimal. Once the model is fitted, these techniques do not account for the data anymore and only act on the previously learned information.

2.5 Measure for classification performance

Obviously, when dealing with fair classifier, one does not only care about fairness but also about the performance of the classifier. There are several ways to assess the performance of a classification method such as recall, precision or f-score [24]. In this paper as in the vast majority of the paper that focus on fair classification we will use the accuracy criteria, i.e.

$$accuracy = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \tag{2.33}$$

where

- TP is the true positive predictions, i.e. the predictions that were correctly assigned to the + class.
- TN is the true negative predictions, i.e. the predictions that were correctly assigned to the - class.
- FP is the false positive predictions, i.e. the predictions that were incorrectly assigned to the + class.
- FN is the false negative predictions, i.e. the predictions that were incorrectly assigned to the - class.

Chapter 3

Supervised classification and maximum entropy

A common task in machine learning is to predict a target variable Y based on observation X of several attributes. For instance, we could try to predict the number of views of an online article given observations on the date of publication, the subject of the article, the number of words in the article etc. In this case, the target variable Y would be numerical and we call this task a regression problem. When, on the other hand, Y is a categorical variable we talk about classification problems. As we said earlier, the focus of this paper is on classification problem. There exists plenty of different algorithms to solve such problems, among which logistic regression.

Logistic regression is a very popular classifier as it is efficient, easy to implement and easy to interpret [34]. In this section, we will first present a simple version of logistic regression. Then, we will introduce the maximum entropy principle and show that we can derive logistic regression from a maximum entropy argument. The model that we will have will be equivalent to logistic regression and will allow to introduce new (linear) constraints to enforce fairness. As [27, 28] shows, logistic regression formulated from a maximum entropy argument cope well with linear constraints.

3.1 Logistic regression

Instead of predicting the binary discrete target variable Y directly, logistic regression will model the probabilities that Y belong to a certain category. Let us consider $\mathbf{x} \in \mathbb{R}^m$ be a vector containing observations of m features including an extra column

of 1's for the bias term and for the sake of simplicity, we will restrict ourselves to a binary target variable, i.e. dom$y = \{-, +\}$. We will need a function $p : \mathbb{R}^m \to [0, 1]$, i.e. a function that maps any input $\mathbf{x}$ to a probability between 0 an 1. There are numerous such functions but logistic regression uses the following :

$$p(\mathbf{x})_+ = \frac{e^{\lambda \cdot \mathbf{x}}}{1 + e^{\lambda \cdot \mathbf{x}}} \tag{3.1}$$

$$p(\mathbf{x})_- = 1 - p(\mathbf{x})_+ = \frac{1}{1 + e^{\lambda \cdot \mathbf{x}}} \tag{3.2}$$

with $\lambda \in \mathbb{R}^m$ a vector of m unknown parameters. Notice that $p(\mathbf{x})_+$ is the sigmoid function, which is the inverse of the logit function after which logistic regression is named. Equivalently we can write the probability function as

$$p(\mathbf{x})_k = \frac{e^{\lambda_k \cdot \mathbf{x}}}{\sum_{l \in \text{dom}y} e^{\lambda_l \cdot \mathbf{x}}} \tag{3.3}$$

where $k \in \text{dom}y$, $\lambda \in \mathbb{R}^{2 \times m}$ is now a matrix with one row for each category that Y can take (in this case 2). Equivalence with Equation 3.1 can be found by requiring that the values in the second row of λ be 0.

To estimate λ, logistic regression uses the *maximum likelihood* method. The intuition behind this method is the following. Across the training set, we would like that the probabilities $p(\mathbf{x})_+$ be close to 1 when the observations were made on an instance of the + class and conversely that $p(\mathbf{x})_-$ be close to 1 for instances of the $-$ class. Considering that we have a dataset $\mathbf{X} = (x_{if}) \in \mathbb{R}^{n \times m}$ of n samples where $\mathbf{x_i}$ denotes the i^{th} sample and y_i the category of the i^{th} sample, this transposes to the maximization of the likelihood function

$$l(\lambda) = \prod_{i=1}^{n} p(\mathbf{x}_i)_{y_i} \tag{3.4}$$

Equivalently, we can maximize the log-likelihood function which will have a different optimal value but the same optimum. The log-likelihood function is

$$ll(\lambda) = \log\left(\prod_{i=1}^{n} p(\mathbf{x}_i)_{y_i}\right) \tag{3.5}$$

$$= \sum_{i=1}^{n} \log p(\mathbf{x}_i)_{y_i} \tag{3.6}$$

The log-likelihood function is concave as it it the sum of log functions which are concave themselves. We know that the derivatives with respect to λ_{kf} must vanish

at the optimal value. Let us first compute the derivative of the probability function with respect to λ_{kf}

$$\frac{\partial}{\partial \lambda_{kf}} p(\mathbf{x_i})_k = x_{if} p(\mathbf{x}_i)_k (1 - p(\mathbf{x}_i)_k) \tag{3.7}$$

$$\frac{\partial}{\partial \lambda_{kf}} p(\mathbf{x_i})_l = -x_{if} p(\mathbf{x}_i)_k p(\mathbf{x}_i)_l \quad \text{for } l \neq k \tag{3.8}$$

Now we can compute the derivative of the log-likelihood function with respect to λ_{kf} using the chain-rule

$$\frac{\partial}{\partial \lambda_{kf}} ll(\lambda) = \frac{\partial}{\partial \lambda_{kf}} \sum_{i=1}^{n} \log p(\mathbf{x}_i)_{y_i} \tag{3.9}$$

$$= \sum_{i=1}^{n} \frac{1}{p(\mathbf{x}_i)_{y_i}} \frac{\partial}{\partial \lambda_{kf}} p(\mathbf{x})_{y_i} \tag{3.10}$$

$$= \sum_{\substack{i=1 \\ y(i)=k}}^{n} \frac{1}{p(\mathbf{x}_i)_k} \frac{\partial}{\partial \lambda_{kf}} p(\mathbf{x}_i)_k + \sum_{\substack{i=1 \\ y(i)\neq k}}^{n} \frac{1}{p(\mathbf{x}_i)_{y_i}} \frac{\partial}{\partial \lambda_{kf}} p(\mathbf{x}_i)_{y_i} \tag{3.11}$$

$$= \sum_{\substack{i=1 \\ y(i)=k}}^{n} \frac{1}{p(\mathbf{x}_i)_k} x_{if} p(\mathbf{x}_i)_k (1 - p(\mathbf{x}_i)_k) - \sum_{\substack{i=1 \\ y(i)\neq k}}^{n} \frac{1}{p(\mathbf{x}_i)_{y_i}} x_{if} p(\mathbf{x}_i)_{y_i} p(\mathbf{x})_k \tag{3.12}$$

$$= \sum_{\substack{i=1 \\ y(i)=k}}^{n} x_{if} (1 - p(\mathbf{x}_i)_k) - \sum_{\substack{i=1 \\ y(i)\neq k}}^{n} x_{if} p(\mathbf{x}_i)_k \tag{3.13}$$

$$= \sum_{\substack{i=1 \\ y_i=k}}^{n} x_{if} - \sum_{i=1}^{n} x_{if} p(\mathbf{x}_i)_k \tag{3.14}$$

We can set this derivative to 0 and we find

$$\sum_{\substack{i=1 \\ y_i=k}}^{n} x_{if} = \sum_{i=1}^{n} x_{if} p(\mathbf{x}_i)_k \quad \text{for all } k, f \tag{3.15}$$

This is a set of $m \times 2$ non-linear equations that we can solve to find the estimate $\hat{\lambda}$'s that maximize the log-likelihood function and now we can make predictions on new observations by computing

$$\hat{p}(\mathbf{x}) = \frac{e^{\hat{\lambda} \cdot \mathbf{x}}}{1 + e^{\hat{\lambda} \cdot \mathbf{x}}} \tag{3.16}$$

3.2 Maximum entropy

In this section we introduce the maximum entropy principle and we will see later that logistic regression can be derived from a maximum entropy argument.

The maximum entropy principle was first developed by E.T. Jaynes in 1957 [4]. In his work, he shows the link between mechanics and information theory, and in particular he shows that statistical mechanics should be seen as a special case of logical inference and information theory.

From this idea comes the principle of maximum entropy, a method used to estimate the probability distribution of a random variable. Let us suppose that this variable can take n distinct values with probabilities $p_1, p_2, \ldots, p_n$ such that

$$\begin{array}{ll} p_i & \geq 0 \quad \forall i \\ \sum_{i=1}^{n} p_i & = 1 \end{array} \tag{3.17}$$

Some additional knowledge, for instance the expected value, might be known about this random variable. With this additional information, we can write linear equations that must be satisfied by our random variable. Those equations have the form

$$\sum_{i=1}^{n} a_i p_i = b \tag{3.18}$$

With each new equation, the distribution is further constrained and the imprecision on the probabilities is reduced. Nevertheless, usually there is still an infinite number of distributions that verify these constraints. How to choose among them? The principle of maximum entropy answers this question. It states that the probability distribution that best describes a certain system is the one that maximises entropy. In other words, the one that best describes the system is the one that makes no additional assumptions about the data, the one that uses only information that is known for sure. This approach is conservative and quite intuitive. We do not want to impose constraints on the distribution beyond those that we know it must satisfy.

3.2.1 Shannon's entropy

Claude Shannon proposed in 1948 [14] a mathematical expression to quantify the uncertainty of a random variable and referred to it as the notion of entropy information. Given a discrete random variable y taking n distinct values with probabilities $p_1, p_2, \ldots, p_n$, the entropy of y is defined as:

$$H(p_1, \ldots, p_n) = -k \sum_{i=1}^{n} p_i \log p_i \tag{3.19}$$

There also exist a continuous analogy for continuous random variable. The random variable y with probability density function $f(y)$ with support Ω has the following Shannon's entropy:

$$H(y) = -k \int_{\Omega} f(y) \log(f(y)) dy \tag{3.20}$$

In both definition, k is an arbitrary positive constant that is in many definition set to 1 and thus omitted.

This measure can be interpreted as the average number of bits needed to describe the random variable. The choice of the logarithm base corresponds to the choice of the base of the binary representation. The higher the uncertainty on the random variable, the higher the function H and thus the higher the number of bits needed to represent it will be.

Shannon gives an example of this concept in [14]. Suppose we have a random variable that takes two distinct values with non-negative probability p and $1 - p$. If we try to maximise the entropy of the random variable we find that $p = \frac{1}{2}$. This means that the uncertainty is the maximal when both event happen with the same probability. In this case we find that Shannon's entropy with a logarithm in base 2 is equal to 1 and thus that the random variable has an entropy of 1 bit. If, on the contrary, we try to minimize the entropy (with the constraint that the probabilities should be non-negative and sum to 1) we find that the uncertainty is minimal when $p = 0$ or $p = 1$. In this case, we are sure about the outcome, the entropy is 0 (perfect information).

The function from Equation 3.19 has interesting properties when it comes to optimization. The first one is that it is a strongly concave function. Let us prove it by showing that $-H(y)$ is convex. We start with a probability mass function $p : \Omega \to [0, 1]$ such that $\sum_{y \in \Omega} p(y) = 1$. We have

$$-H(p) = \sum_{y \in \Omega} p(y) \log p(y) \tag{3.21}$$

where we consider that $H(0) = \lim_{p \to 0} p \log p = 0$.[1]

We denote by $\mathcal{P} \subseteq \mathbb{R}^{\Omega}$ the set of all probability mass functions on Ω. This set is

[1]This is necessary because p denotes a probability and a probability of 0 is possible in some cases.

compact and convex. Furthermore, let us show that the function $f : p \to p \log p$ is strongly convex. By definition[2], it is when we can find a constant $\mu > 0$ such that

$$\langle \nabla f(y) - \nabla f(x), y - x \rangle \geq \mu ||y - x||^2 \quad , \quad \forall x, y \in [0, 1] \tag{3.22}$$

in our case we have $\nabla f(y) = \log y + 1$ and the previous equation transposes to

$$\log y - \log x \geq \mu(y - x) \quad \forall x, y \in [0, 1] \tag{3.23}$$

We can start with the fact that the mapping $p \to \log p$ is concave and thus it holds that [20]

$$\log x \leq \log y + \frac{1}{y}(x - y) \quad , \quad \forall x, y \in [0, 1] \tag{3.24}$$

$$\Leftrightarrow \log y - \log x \geq \frac{1}{y}(y - x) \quad , \quad \forall x, y \in [0, 1] \tag{3.25}$$

$$\Leftrightarrow \log y - \log x \geq \frac{1}{\mu}(y - x) \quad , \quad \forall x, y \in [0, 1], \mu \in (0, 1) \tag{3.26}$$

$$\tag{3.27}$$

where the use of parenthesis in the interval denote the open interval between 0 and 1. The first inequality can be loosely explained by the fact that the tangent line computed at one point of a concave function is always above its graph. The mapping $f : p \to p \log p$ is thus strongly convex. Moreover, the sum of convex functions with at least one strongly convex function is strongly convex and so we conclude that $-H(p)$ is indeed strongly convex and conversely that $H(p)$ is strongly concave. From an optimization point of view, this is a nice property because it means that when optimized the function $H(p)$ will admit at most one (global) maximizer [20]. This only required that the function be strictly concave but strong concavity implies strict concavity. In addition, some very popular methods require the objective function to be convex (interior-point method for instance) and methods usually converge faster when the objective is strongly convex/concave. For instance, gradient method used to maximize (minimize) strongly concave (convex) functions has a linear convergence [20].

A second nice property of Equation 3.19 is that the logarithm ensures that the probabilities are non-negative. The case where the probabilities are zero is handled by the remark above stating that we should consider $H(0) = \lim_{p \to 0} p \log p = 0$. The importance of this property will become apparent in what follows.

[2]The reader might be more familiar with the definition $f(y) \geq f(x) + \nabla f(x)^T (y - x) + \frac{\mu}{2} \|y - x\|^2$. The definition above comes from this and the requirement that $f(y) \geq f(x) + \nabla f(x)^T (y - x)$, $\forall x, y$ which is roughly saying that the graph of a convex function is above its tangent (in 2 dimensions).

Finally, a last thing we need to emphasize is that there are other forms of entropy than the one defined by Shannon. Authors of [15] draw a comparison between Shannon, Tsalis and Renyi entropy. In a nutshell, the two later use measures that depend on powers of probability $\sum_{i=1}^{n} p_i^{\alpha}$ which allows to increase (decrease) the impact of outliers when α is a large negative (positive) value.

Using Shannon's definition with the entropy principle, we end up with the following optimization program

$$\left|\begin{array}{ll} \underset{\{p_i\}}{\text{maximize}} & -\sum_{i=1}^{n} p_i \log p_i \\ \text{subject to} & \sum_{i=1}^{n} a_{im} p_i = b_{im} \quad \text{for all } m \\ & \sum_{p_i=1} p_i = 1 \end{array}\right. \tag{3.28}$$

assuming m linear constraints that reflect knowledge on the probability distribution. Notice that the non-negativity does not appear explicitly in this optimization program. The problem is convex since every constraint is linear and we are maximizing (strongly) concave objective function. This optimization program can be solved by very efficient algorithm such as interior-point method or using conic formulation. [20]

3.3 Maximum entropy for classification

Let us now have a look on how the theory that we have discussed above transpose in a classification setting. Our goal is to assign a probability for each of the possible target classes. Here, we will focus mainly on the binary classification task, i.e. there will only be two possible target classes. Moreover, for the sake of simplicity we will derive methods for problem considering only one sensible variable. The extension to multiple sensible variables is fairly easy and straightforward.

3.3.1 General formulation

Let us consider a $n \times m$ data matrix $\mathbf{X} = (x_{if})$ whose elements x_{if} contain the measurement of feature f on sample i of the training set. There are m such features, including an extra feature for the intercept or bias term of the regression. This extra column only contains 1's. Categorical variables are represented by binary indicator variables in a one-hot encoding fashion as usual.

The observed target values are recorded in the $n \times q$ target matrix $\mathbf{Y} = (y_{ik})$ where q is the number of different classes and y_{ik} encodes the class membership of each individual i. As it is assumed that classes are mutually exclusive, the y_{ik} are binary indicator variables, $y_{ik} = 1$ if sample i is categorized in class k and $y_{il} = 0$ for all $l \neq k$. The predicted a posteriori probabilities of belonging to each class, provided by the supervised classification model, are $\hat{y}_{ik}$ with $\sum_{k=1}^{q} \hat{y}_{ik} = 1$, and are encoded in $\hat{\mathbf{Y}}$.

The maximum entropy (MaxEnt) problem for classification can be stated as follows,

$$\left| \begin{array}{lll} \underset{\{\hat{y}_{ik}\}}{\text{maximize}} & -\sum_{i=1}^{n}\sum_{k=1}^{q} \hat{y}_{ik} \log \hat{y}_{ik} & \\ \text{subject to} & \sum_{i=1}^{n} \hat{y}_{ik} x_{if} = \sum_{i=1}^{n} y_{ik} x_{if} & \text{for all } f, k \\ & \sum_{k=1}^{q} \hat{y}_{ik} = 1 & \text{for all } i \end{array} \right. \tag{3.29}$$

The objective function is simply the maximisation of Shannon's entropy. The first constraint is very important, we will refer to it as the co-moment features-predictions constraint. It states that, within any class, the sum of all the observed values of any feature should be equal to the sum of probability mass multiplied by the observation value of the same feature across all data. The second constraint simply states that the sum of probabilities of one observation across all classes must sum to 1.

3.3.2 Optimality conditions

Let us compute the Lagrangian function of the optimization problem 3.29. With some arbitrary choice, the Lagrangian can be written as follows

$$\begin{aligned} \mathscr{L} = &- \sum_{i=1}^{n}\sum_{k=1}^{q} \hat{y}_{ik} \log \hat{y}_{ik} + \sum_{f=1}^{m}\sum_{k=1}^{q} \lambda_{kf} \Big(\sum_{i=1}^{n} \hat{y}_{ik} x_{if} - y_{ik} x_{if} \Big) \\ &+ \sum_{i=1}^{n} \mu_i \Big(\sum_{k=1}^{q} \hat{y}_{ik} - 1 \Big) \end{aligned} \tag{3.30}$$

where λ's and μ's are the free dual variable of the first and the second constraint in problem 3.29 respectively. As a consequence of Karush-Kuhn-Tucker theorem, the derivative of this Lagrangian function with respect to $\hat{y}_{ik}$ will vanish at the optimum. We can thus compute the partial derivative with respect to the predicted

values, $\partial \mathscr{L} / \partial \hat{y}_{ik}$, and set the result to zero. We readily obtain

$$0 = -\log \hat{y}_{ik} - 1 + \sum_{f=1}^{m} \lambda_{kf} x_{if} + \mu_i$$

$$\hat{y}_{ik} = \exp\left(\sum_{f=1}^{m} \lambda_{kf} x_{if} + \mu_i - 1 \right) \tag{3.31}$$

We can further develop this equation by using the fact that the sum of the probabilities for a given observation across all categories must add to 1.

$$\sum_{k=1}^{q} \exp\left(\sum_{f=1}^{m} \lambda_{kf} x_{if} + \mu_i - 1 \right) = 1 \tag{3.32}$$

This yields,

$$\exp(\mu_i - 1) = \left(\sum_{k=1}^{q} \exp\left(\sum_{f=1}^{m} \lambda_{kf} x_{if} \right) \right)^{-1} \tag{3.33}$$

which we can plug into equation 3.31 to obtain the form

$$\hat{y}_{ik} = \frac{\exp\left(\sum_{f=1}^{m} \lambda_{kf} x_{if} \right)}{\sum_{l=1}^{q} \exp\left(\sum_{f=1}^{m} \lambda_{lf} x_{if} \right)} \tag{3.34}$$

which is the form of the probability function in Equation 3.3 ! This shows the equivalence between the two formulations, the dual variable in the maximum entropy formulation are the coefficients fitted by logistic regression. Notice that this form does only depend on the λ's, the dual variable of the first constraint of the optimization problem 3.29. This form is thus suitable to make predictions on the basis of the training set, which is what classification is all about. Starting with an optimization program and a training set, we have been able to compute an equation that can be used to estimate the probability that an observation belong to a certain class. At this point, we did not take fairness into account, the next section will incorporate fairness into the model that we have just seen.

Chapter 4

Managing fairness constraints in classification

In the previous chapter, we have seen that a very popular classification model, logistic regression, can be derived from a maximum entropy argument under the form of a convex optimization program with linear constraints [27, 28]. We know that our MaxEnt model copes well with linear constraints. Hence, we propose to integrate linear constraints to our formulation from Equation 3.29 to enforce fairness.

4.1 Complete decorrelation with respect to the sensible variables

First, we propose a model that will impose that the correlation between the sensible variable and the predicted class probabilities $\hat{y}_{ik}$ is zero. Let $\mathbf{z}$ be the $n \times 1$ column vector of measurements of a *sensible* variable. We want to impose that each $n \times 1$ prediction vector $\hat{\mathbf{y}}_k$ (column k of $\hat{\mathbf{Y}}$) is uncorrelated with this sensitive variable $\mathbf{z}$. Thus the covariance between the prediction vector and the sensitive variable should be zero, $\text{cov}(\hat{\mathbf{y}}_k, \mathbf{z}) = 0$. Recall that we have

$$\text{cov}(\mathbf{x}, \mathbf{y}) = \frac{1}{n-1} \sum_{i \in \mathcal{T}} (x_i - \bar{x})(y_i - \bar{y}) \tag{4.1}$$

$$= \frac{1}{n-1} \mathbf{x}^\mathsf{T} \mathbf{H} \mathbf{y} \tag{4.2}$$

where $\mathbf{H} = (\mathbf{I} - \frac{1}{n}\mathbf{e}\mathbf{e}^\mathsf{T})$ is the centering operator introduced earlier.

Therefore, the null covariance constaint can be written $\mathbf{z}^\mathsf{T}\mathbf{H}\hat{\mathbf{y}}_k = 0$ and after defining $\tilde{\mathbf{z}} = \mathbf{H}\mathbf{z}$, the constraint becomes $\tilde{\mathbf{z}}^\mathsf{T}\hat{\mathbf{y}}_k = 0$.

Integrating these new constraints into the maximal entropy problem yields

$$\left|\begin{array}{lll} \underset{\{\hat{y}_{ik}\}}{\text{maximize}} & -\sum_{i=1}^{n}\sum_{k=1}^{q}\hat{y}_{ik}\log\hat{y}_{ik} & \\ \text{subject to} & \sum_{i=1}^{n}\hat{y}_{ik}x_{if} = \sum_{i=1}^{n}y_{ik}x_{if} & \text{for all } f,k \\ & \sum_{i=1}^{n}\hat{y}_{ik}\tilde{z}_i = 0 & \text{for all } k \\ & \sum_{k=1}^{q}\hat{y}_{ik} = 1 & \text{for all } i \end{array}\right. \tag{4.3}$$

Assuming the problem is feasible, we proceed as before and the Lagrange function becomes

$$\begin{aligned}\mathscr{L} = &-\sum_{i=1}^{n}\sum_{k=1}^{q}\hat{y}_{ik}\log\hat{y}_{ik} + \sum_{f=1}^{m}\sum_{k=1}^{q}\lambda_{kf}\left(\sum_{i=1}^{n}(\hat{y}_{ik}x_{if} - y_{ik}x_{if})\right) \\ &+\sum_{k=1}^{q}\nu_k\left(\sum_{i=1}^{n}\hat{y}_{ik}\tilde{z}_i - 0\right) + \sum_{i=1}^{n}\mu_i\left(\sum_{k=1}^{q}\hat{y}_{ik} - 1\right)\end{aligned} \tag{4.4}$$

By taking the partial derivative and setting the result to 0, we obtain

$$-\log\hat{y}_{ik} - 1 + \sum_{f=1}^{m}\lambda_{kf}x_{if} + \nu_k\tilde{z}_i + \mu_i = 0 \tag{4.5}$$

which leads to

$$\hat{y}_{ik} = \frac{\exp\left(\sum_{f=1}^{m}\lambda_{kf}x_{if}\right)\exp\left(\nu_k\tilde{z}_i\right)}{\sum_{l=1}^{q}\exp\left(\sum_{f=1}^{m}\lambda_{lf}x_{if}\right)\exp\left(\nu_l\tilde{z}_i\right)} \tag{4.6}$$

This form is proportional to equation 3.34 by a factor that depends on the dual variable of the fairness condition constraining complete decorrelation between the probabilities $\hat{y}_{ik}$ and the sensible variable. Again it only depends on dual variables of the optimization program and this form is suitable to make predictions on new samples. We must emphasize that $\tilde{z}_i$ represents the centered observation of the sensible variable of the sample we are trying to make predictions for. When making predictions on new sample, we should use the **same centering** that we used during training. More specifically, centering the variable amounts to subtract the mean of the sensible variable across the **training set** and not the test set.

4.2 Partial decorrelation with respect to the sensible variables

Now, we propose to loosen the fairness constraint a little bit. A partial decorrelation can also be achieved by using inequality constraints, stating that the absolute value of the covariance is no more that some predefined (small) value $\epsilon > 0$. Of course we want our constraint to remain linear so the absolute value cannot appear in the optimization program, we split this constraint into two linear constraints.

$$\left|\begin{array}{lll} \underset{\{\hat{y}_{ik}\}}{\text{maximize}} & -\sum_{i=1}^{n}\sum_{k=1}^{q} \hat{y}_{ik}\log\hat{y}_{ik} & \\ \text{subject to} & \sum_{i=1}^{n}\hat{y}_{ik}x_{if} = \sum_{i=1}^{n} y_{ik}x_{if} & \text{for all } f,k \\ & \frac{1}{n-1}\sum_{i=1}^{n}\hat{y}_{ik}\tilde{z}_i \leq +\epsilon & \text{for all } k \\ & \frac{1}{n-1}\sum_{i=1}^{n}\hat{y}_{ik}\tilde{z}_i \geq -\epsilon & \text{for all } k \\ & \sum_{k=1}^{q}\hat{y}_{ik} = 1 & \text{for all } i \end{array}\right. \tag{4.7}$$

We compute a Lagrangian function

$$\begin{aligned} \mathscr{L} = &-\sum_{i=1}^{n}\sum_{k=1}^{q}\hat{y}_{ik}\log\hat{y}_{ik} + \sum_{f=1}^{m}\sum_{k=1}^{q}\lambda_{kf}\left(\sum_{i=1}^{n}(\hat{y}_{ik}x_{if} - y_{ik}x_{if})\right) \\ &+ \sum_{k=1}^{q}\alpha_k\left(\epsilon - \frac{1}{n-1}\sum_{i=1}^{n}\tilde{z}_i\hat{y}_{ik}\right) + \sum_{k=1}^{q}\beta_k\left(\epsilon + \frac{1}{n-1}\sum_{i=1}^{n}\tilde{z}_i\hat{y}_{ik}\right) \\ &+ \sum_{i=1}^{n}\mu_i\left(\sum_{k=1}^{q}\hat{y}_{ik} - 1\right) \end{aligned} \tag{4.8}$$

$$\tag{4.9}$$

We compute the derivative of $\mathscr{L}$ with respect to $\hat{y}_{ik}$ and set it to 0. We find

$$\frac{\partial\mathscr{L}}{\partial\hat{y}_{ik}} = -\log\hat{y}_{ik} - 1 + \sum_{f=1}^{m}\lambda_{kf}x_{if} + \mu_i + \frac{\tilde{z}_i}{n-1}(\beta_k - \alpha_k) = 0 \tag{4.10}$$

and in the same way as before to remove μ_i from the equation, this yields

$$\hat{y}_{ik} = \frac{\exp\left(\sum_{f=1}^{m}\lambda_{kf}x_{if}\right)\exp\left(\frac{\tilde{z}_i}{n-1}(\beta_k - \alpha_k)\right)}{\sum_{l=1}^{q}\exp\left(\sum_{f=1}^{m}\lambda_{lf}x_{if}\right)\exp\left(\frac{\tilde{z}_i}{n-1}(\beta_l - \alpha_l)\right)} \tag{4.11}$$

This form is again proportional to equation 3.34 by a factor that depends on the dual variables of the fairness constraints. This form is again suitable for predictions

on new samples. The same remark that we made in the previous section about the centering of the sensible variable can be made here.

4.3 Pre-processing techniques

As discussed in the second section of this paper, pre-processing techniques have shown interesting result in fair machine learning. In this section, we propose a preprocessing technique to mitigate the effect of sensible variables in our binary classification task. The idea is to use projection to remove **linear correlation** between the data set and the sensible variable and then use this uncorrelated (with respect to the sensible variables) data matrix to make predictions. This technique is closely related to the technique of *partialling out* in statistics when controlling with respect to some control variables, in our case the protected variables (see [25, 26] for instance).

Assume as before that we have recorded the values of m features recorded in a $n \times m$ data set $\mathbf{X}$.

$$\mathbf{X} = [\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^m] \tag{4.12}$$

where $\mathbf{x}^f$ represents column f of the data matrix and thus the observations of feature f. Let $\mathbf{H} = (\mathbf{I} - \frac{1}{n}\mathbf{e}\mathbf{e}^\mathsf{T})$ be the centering operator with $\mathbf{e}$ a $n \times 1$ column vector containing 1's. Notice that $\mathbf{HX}$ has the effect of centering each column vector and thus each variable,

$$\mathbf{HX} = [\mathbf{H}\mathbf{x}^1, \mathbf{H}\mathbf{x}^2, \dots, \mathbf{H}\mathbf{x}^p] \tag{4.13}$$

It thus centers the data matrix.

A number of p sensible variables have also been recorded together with the data matrix for each sensible variable. The matrix containing the observations of the sensible variables is

$$\mathbf{Z} = [\mathbf{z}^1, \mathbf{z}^2, \dots, \mathbf{z}^p] \tag{4.14}$$

Now, it is well-known that the orthogonal projection operator on the column space of $\mathbf{Z}$ is

$$\mathbf{\Pi}_\mathbf{Z} = \mathbf{Z}(\mathbf{Z}^\mathsf{T}\mathbf{Z})^{-1}\mathbf{Z}^\mathsf{T} \tag{4.15}$$

Therefore, because for each column vector $\mathbf{v}$, $\mathbf{\Pi}_\mathbf{Z}\,\mathbf{v} + \mathbf{\Pi}_{\perp\mathbf{Z}}\,\mathbf{v} = \mathbf{v}$, the projection on the column subspace orthogonal to $\mathbf{Z}$ is

$$\mathbf{\Pi}_{\perp\mathbf{Z}} = \mathbf{I} - \mathbf{\Pi}_\mathbf{Z} \tag{4.16}$$

Also note that $\mathbf{\Pi_Z}\,\mathbf{Z} = \mathbf{Z}$, $\mathbf{\Pi_{\perp Z}}\,\mathbf{Z} = \mathbf{0}$ and $\mathbf{z} \cdot \mathbf{\Pi_{\perp z}}\,\mathbf{v} = 0$ where $\cdot$ is the inner product. Moreover, the projection applies on each column of the matrix,

$$\mathbf{\Pi_{\perp Z}}\,\mathbf{X} = [\mathbf{\Pi_{\perp Z}}\,\mathbf{x}^1, \mathbf{\Pi_{\perp Z}}\,\mathbf{x}^2, \ldots, \mathbf{\Pi_{\perp Z}}\,\mathbf{x}^m] \tag{4.17}$$

4.3.1 Un-correlating the data matrix wrt the sensible variables

Let us further define the projection of $\mathbf{x}$ on the subspace orthogonal to the centered matrix $\tilde{\mathbf{Z}} = \mathbf{HZ}$ as $\mathbf{x}_{\perp\tilde{\mathbf{Z}}} = \mathbf{\Pi}_{\perp\tilde{\mathbf{Z}}}\,\mathbf{x}$, for any column $\mathbf{x}$ of data matrix $\mathbf{X}$. Thus, Equation (4.15) is used with $\tilde{\mathbf{Z}}$ instead of $\mathbf{Z}$.

In this context, the empirical covariance between any column $\mathbf{z}$ of matrix $\mathbf{Z}$ and $\mathbf{x}_{\perp\tilde{\mathbf{Z}}}$ is

$$\begin{aligned}\operatorname{cov}(\mathbf{z}, \mathbf{x}_{\perp\tilde{\mathbf{Z}}}) &= \tfrac{1}{n-1}\mathbf{z}^\mathsf{T}\mathbf{H}\,\mathbf{x}_{\perp\tilde{\mathbf{Z}}} = \tfrac{1}{n-1}(\tilde{\mathbf{z}})^\mathsf{T}\mathbf{x}_{\perp\tilde{\mathbf{Z}}} \\ &= \tfrac{1}{n-1}(\tilde{\mathbf{z}})^\mathsf{T}\mathbf{\Pi}_{\perp\tilde{\mathbf{Z}}}\,\mathbf{x} = \tfrac{1}{n-1}(\tilde{\mathbf{z}}) \cdot (\mathbf{\Pi}_{\perp\tilde{\mathbf{Z}}}\,\mathbf{x}) \\ &= 0\end{aligned} \tag{4.18}$$

By using the same argument and $\mathbf{HH} = \mathbf{H}$, it follows that the covariance with $\mathbf{H}\,\mathbf{x}_{\perp\tilde{\mathbf{Z}}}$ (the centered vector) is also equal to 0. Therefore, the projected data matrix $\mathbf{X}_{\perp\tilde{\mathbf{Z}}}$ and the centered projected data matrix are both uncorrelated with the sensible variables.

4.4 Post-processing techniques

Post-processing techniques are applied after the predictions are made. It works in two steps. First, we only care about classification performance and fairness is put aside. Then, in a second round, we constrain the predictions to satisfy some fairness constraint. The second step is obviously going to constrain the original problem even more and the solution will often be less accurate.

Two approaches are proposed. In the second step, both of these consist of a least square optimization program with a constraint on the covariance between the sensible variables and the predictions. However, the two models will not make predictions on the same data set nor will they apply the fairness constraint on the same set.

First approach In the first step of this approach, the model will be fitted over the training set and then predictions are going to be made both on the prediction set and the test set. After that, we optimize a least square problem considering all the predictions with a fairness constraint on all the predictions as well. Finally, to assess performances, we only care about the predictions made on the test set.

With our maximum entropy model, we would keep the same model illustrated by the optimization program 3.29 and we would have the same expression as Equation 3.34 for the predictions but instead of making predictions on new samples only, we would also make predictions on the training set that we used to fit the MaxEnt model. Assume that d is the number of samples in the test set and that as previously, n denotes the number of samples in the training set. Then, we would optimize the following least square problem for the variable $\tilde{y}_{ik}$

$$\left|\begin{array}{lll} \underset{\{\tilde{y}_{ik}\}}{\text{minimize}} & \sum_{i=1}^{n+d}\sum_{k=1}^{q}(\hat{y}_{ik}-\tilde{y}_{ik})^2 & \\ \text{subject to} & \sum_{i=1}^{n+d}\tilde{y}_{ik}\tilde{z}_i \leq +\epsilon & \text{for all } k \\ & \sum_{i=1}^{n+d}\tilde{y}_{ik}\tilde{z}_i \geq -\epsilon & \text{for all } k \\ & \sum_{k=1}^{q}\tilde{y}_{ik} = 1 & \text{for all } i \\ & \tilde{y}_{ik} >= 0 & \text{for all } i,k \end{array}\right. \tag{4.19}$$

where *for all* i should be understood as $\forall i \in \{1,\ldots,n+d\}$ and not until n as before.

Second approach The second approach is more standard. We fit the model on the training set, then we make predictions on the test set and we only optimize the least square problem for the predictions that we made, i.e. the predictions on the test set.

The first step is thus the same as before since we also fit on the training set. The expression for the prediction is still the same as Equation 3.34. The least square optimization problem is the following

$$\left|\begin{array}{lll} \underset{\{\tilde{y}_{ik}\}}{\text{minimize}} & \sum_{i=1}^{d}\sum_{k=1}^{q}(\hat{y}_{ik}-\tilde{y}_{ik})^2 & \\ \text{subject to} & \sum_{i=1}^{d}\tilde{y}_{ik}\tilde{z}_i \leq +\epsilon & \text{for all } k \\ & \sum_{i=1}^{d}\tilde{y}_{ik}\tilde{z}_i \geq -\epsilon & \text{for all } k \\ & \sum_{k=1}^{q}\tilde{y}_{ik} = 1 & \text{for all } i \\ & \tilde{y}_{ik} >= 0 & \text{for all } i,k \end{array}\right. \tag{4.20}$$

where *for all* i should now be considered as $\forall i \in \{1, \ldots, d\}$ and not n or $n + d$.

Chapter 5

Implementation and experimental methodology

5.1 Implementation

To test the performance of our models, each of them have been implemented in Python. The MaxEnt classifier was written as a class following the same structure as other classifiers in the sklearn[1] library. The first part is the training part where the model fits the data solving an optimization problem as discussed in the previous section. Then, one can provide the model with new data and the model will make predictions on this sample.

Learning phase This process happens in the *fit* method. It takes two arguments

- $\mathbf{X}$, the $n \times p$ data matrix with n samples and p features.
- $\mathbf{Y}$, the $n \times 1$ label vector of the training set. MaxEnt is fitted with supervised learning, this means that it requires the labels of all the samples in the training set.

The optimization part of the training is performed using the CVXPY[2] package and Mosek[3] as solver. First, we build the optimization problem, defines the objective function as Shannon's entropy, create the constraints etc. Then the optimization program is solved and the dual variables of the constraints are stored.

[1] `https://scikit-learn.org/`
[2] `https://www.cvxpy.org/`
[3] `https://www.mosek.com/`

Making predictions The two functions responsible for making predictions on new samples once the model is fitted are the *predict* and *predict_proba* methods. Both take only one argument

- **X**, the data matrix of the test set. The number of samples is not important but there should be the same number of features, i.e. the same number of columns as the data matrix of the train set.

The *predict_proba* procedure outputs a vector $\hat{\mathbf{y}}$ with values in [0,1] that give the probability of belonging to the + class for each sample. The *predict* procedure is the decision function of the classifier that outputs the predicted class for each sample, i.e. the output is in $\{0, 1\}$ The decision function is simply to output 0 when the probability of belonging to the + class is below .5 and to output 1 in the other case.

5.1.1 Datasets

Three datasets were used for experimentation. They all come from real data and they are often used when it comes to fairness classification.

COMPAS COMPAS stands for Correctional Offender Management Profiling for Alternative Sanctions dataset, it is a commercial algorithm provided for judges and parole officers to give a score to a criminal's likelihood of re-offending. This algorithm was used in several U.S. states such as New York, Wisconsin and California. A few years ago, Propublica, a non-profit organisation specialising in the investigation of issues of public interest, published an analysis stating that the algorithm was biased in favor of white defendants. This dataset contains observations of variables used by the COMPAS algorithm and the re-offending outcome within 2 years. It contains 5278 samples of 12 features, 5 of which are categorical and 7 continuous. The sensible variable is race which is either Caucasian or African-American and the target is binary and is either 0 (no recidivism within 2 years) or 1 (recidivism within 2 years).

Default This dataset contains 30,000 instances and 24 attributes. The classification task is to determine whether or not a person will be in default of payment. The protected attribute is gender. Other attributes include amount of the credit, education, marital status, age, history of past payment etc. To keep low computational cost, 30% of the data were sampled at random. The protected variable is a little bit unbalance (2 against 1). Earlier, we stated that one cause of unfairness is that sometimes, classifiers optimize for the majority class at the expense of the minority class because it can not optimize for both at the same time. It will be interesting to see if accuracy is significantly different for the two protected groups.

Student The prediction task of this dataset is to decide if a student will pass or fail a math course. It contains 395 samples and 30 features. Other features are for instance age, school (binary between two schools), address (urban or rural), father education, mother education, romantic status, number of past failures, etc. The target attribute is called *G1* in the dataset and represents the final grade. It is replaced by a binary variable with value 1 when then grade is ≥ 10 and 0 otherwise. The protected attribute is gender.

5.1.2 Experimental tests

Different models were implemented and compared during the experimentation.

- **baseline.** The first model (acting as baseline to get reference accuracy and correlation score) is a simple MaxEnt as introduced in Section 3.3 which predicts the target class on the basis of the data matrix (that does not include the sensible variable).
- **projection.** Then the same model is used but rather than learning on the data matrix X, we learn on the projected data matrix that is uncorrelated with the sensible variable (pojection technique introduced in Section 4.3). A choice has to be made about whether to apply projection on the training set and on the data set combined or if we use projections on those set separately. The first approach is more heavy computationally but projections are relatively cheap in that respect. Both approaches will be tested and compared.
- **with threshold constraint.** This is the model introduced in Section 4.2.
- **post-processing.** The two approaches in Section 4.4 that we proposed to achieve post-processing techniques that enforce fairness.
- **pre-processing + post-processing.** A combination of pre-processing (projection) and then post-processing.

The model with the threshold constraint and the two post-processing approaches were tested for different value of the parameter ϵ in the interval $\{0\} \cup [0.001, 0.01]$. The choice of the interval was made empirically. We also need to choose if we will standardize the data before fitting. This is not required by logistic regression but optimization algorithm converge faster when the data are scaled. The models have been tested both with scaled and raw data. However, only the results for the scaled data will appear in the paper as they are better and depending on the dataset *Mosek* sometimes has a hard time converging with the raw data.

5.1.3 Cross validation

For every model, 5-fold cross-validation was used while computing the values for accuracy and correlations. The principle is that the training set gets split into 5 folds of approximately even size. Sequentially, each of the 5 folds is used as a test set while the other 4 folds are used for training. In the end, each fold has been used exactly once as a test set and the model has been fitted on 5 different training set.

Each time, computations were recorded and the end result is the average of the 5 simulations. Averaging the computations over different training and test sets distribution has the benefit of mitigating randomness and yields more reliable outputs.

Chapter 6

Experimental results and discussion

In this chapter we evaluate the different models that we have presented in the previous sections. To do so, we compare them to each other and to a baseline for reference. The criteria are accuracy, which measures the classifier performances, and the correlation between the predictions (binaries and probabilities) and the sensitive variables, which will measure the extent to which the proposed models are able to mitigate the biases from the data sets. The results for the correlation with the binary prediction will be more presented than those of the probability predictions because the purpose of classification is primarily to categorize. All results omitted in the paper can be found in the Appendix.)

We answer several research questions of this thesis in this chapter:

- *Can we show empirically the equivalence between logistic regression and maximum entropy?*
- *What method offer the best trade-off between fairness and classification performances ?*
- *Are pre-processing methods able to fully decorrelate predictions with sensitive variables ?*
- *Can post-processing achieve a good trade-off between classification performance and fairness? How does supervised method compare with semi-supervised approaches ?*
- *Does a combination of pre-processing, post-processing and learning methods improve the results?*

- *How do our methods compare with approaches from literature ?*

6.1 Baseline results

The baseline model is just the maximum entropy formulation of logistic regression that we introduced in Section 3.3. The data matrix X does not contain the sensible attribute. Table 6.1 shows the result observed for each dataset. On *Student* and *Default* we can see that the correlation is already quite small (.02). It will be interesting to see if we can improve the fairness even when the baseline already has a very low correlation.

	Accuracy	Correlation binary	Correlation probability
COMPAS	0.6789	0.2576	0.3131
Student	0.6789	-0.0273	-0.0396
Default	0.8087	-0.0247	-0.0442

Table 6.1: Results of the baseline on all datasets, correlation binary is the correlation between the sensible variable and the binary prediction and correlation probability is the correlation between the sensible variable and the continuous probability of belonging to the + class.

6.2 Projection results

Using projection as a pre-processing technique that will remove linear correlation between the sensible attribute and the data matrix X has been introduced in section 4.3. The model that we used is the same model as in the previous section, i.e. the maximum entropy formulation of logistic regression.

As we mentioned earlier, projections can be applied either on the training set and the test combined or separately. Both have been tried and the results are reported below on Table 6.2 and 6.3. Both approach yield similar results on the *COMPAS* and *Default* datasets but the combined projection seems to work better for the *Student* one.

We can see that correlation has decreased a lot on the *COMPAS* dataset (from 0.25 to about -0.04). On the *Student* dataset the correlation slightly increases and on *Default*, it is the opposite as it slightly decreases. Overall, accuracy has been traded for fairness when fairness improved.

	Accuracy	Correlation binary	Correlation probability
COMPAS	0.6614	-0.0384	-0.0106
Student	0.6688	0.0295	0.0147
Default	0.8087	0.0173	0.0176

Table 6.2: Results obtained with the projected data matrix for all datasets when projection is applied on the training and test set combined.

	Accuracy	Correlation binary	Correlation probability
COMPAS	0.6599	-0.0427	-0.0127
Student	0.6789	0.0895	0.0308
Default	0.8087	0.0173	0.0105

Table 6.3: Results obtained with the projected data matrix for all datasets when projection is applied on the training and test set separately.

6.3 Results for the model with threshold constraints

The model with threshold constraint was introduced in Section 4.2. This model is characterized by the value of the threshold ϵ. We will test different value of ϵ and plot the evolution of the accuracy and of the correlations with respect to the value of the threshold. For visualization, we will also plot the accuracy-correlation pairs for the different value of the threshold to see the different trade-off that we can find.

COMPAS In Figure 6.1 we can see the evolution of the accuracy and of the (absolute value of the) correlation between the sensible variable and the binary predictions when the threshold of the inequality constraint changes. We clearly see that the model trades accuracy for fairness. As the threshold ϵ lowers the correlation lowers as well but at the expense of accuracy that is also diminishing. Here we see that when the threshold gets from 0.01 to 0.005 the correlation with the binary prediction decreases and almost drops to zero but after that value, the correlation is going up again. The accuracy on the other hand, decreases from 0.6789 (Baseline value) to 0.663 which is roughly a 1% decrease in accuracy for a huge fairness gain as we went from 0.2576 to almost 0 (on the binary predictions).

The trade-off between accuracy and correlation is also visible on Figure 6.2. On the

Figure on the right, we see that the correlation with probabilities is monotonically decreasing while it is not the case with the correlation with binary predictions. In our model, we impose a constraint on the covariance with the probabilities and not with the binary predictions. It is interesting to see that the two correlations do not always follow the same trend.

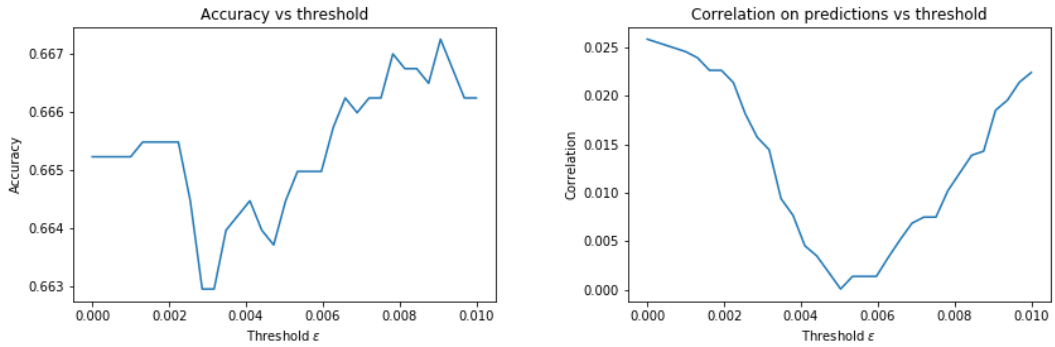


Figure 6.1: COMPAS - Evolution of the predictions accuracy and of the correlation (in absolute value) between the predictions and the sensible variable as a function of the threshold ϵ on inequality constraints.

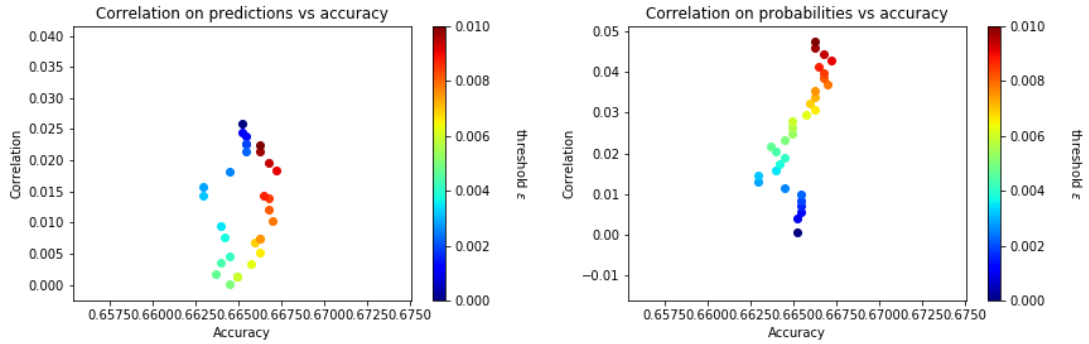


Figure 6.2: COMPAS - Correlation between predictions and sensible variable (left) and between probabilities and sensible variable (right) versus accuracy for different value of the threshold ϵ.

Student The results are a bit different on the Student dataset. On Figure 6.3, it can be seen that accuracy varies in steps. The covariance constraint is on the probabilities, not on the binary predictions and therefore binary predictions do not always vary for different threshold. Imposing the constraints on the binary predictions would not allow linear constraints, which explains this choice. In

this case, it seems that although the probabilities vary to satisfy the threshold constraint, the predictions do not necessarily change as well. It is possible that the probabilities do not vary enough to induce a difference in the binary predictions for successive thresholds. In the COMPAS dataset results, the binary predictions vary with each new threshold but here they seem to do so in intervals (Figure 6.4). Hence, there are different levels of accuracy and different level of correlation with the binary predictions. Again we see that after a threshold value of 0.005, the correlation on binary predictions goes up again.

Figure 6.4 shows the different trade-off. We can see that correlation on the probabilities vary for each threshold and that again we were able to almost completely remove the correlation with binary predictions.

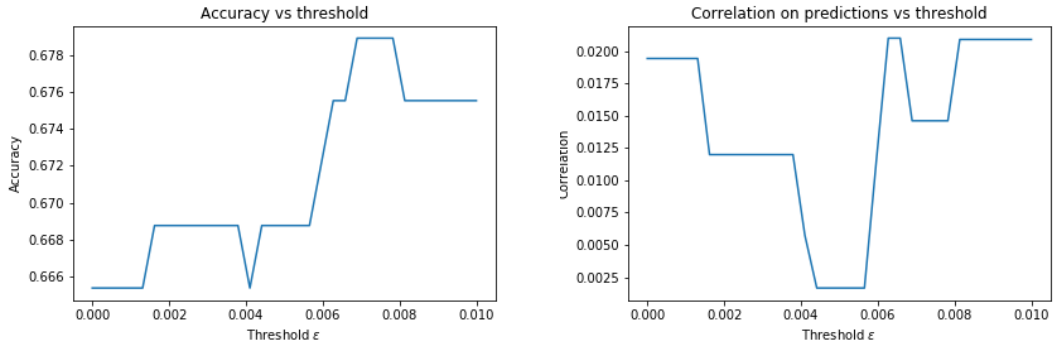


Figure 6.3: STUDENT - Evolution of the predictions accuracy and of the correlation (in absolute value) between the predictions and the sensible variable as a function of the threshold ϵ on inequality constraints.

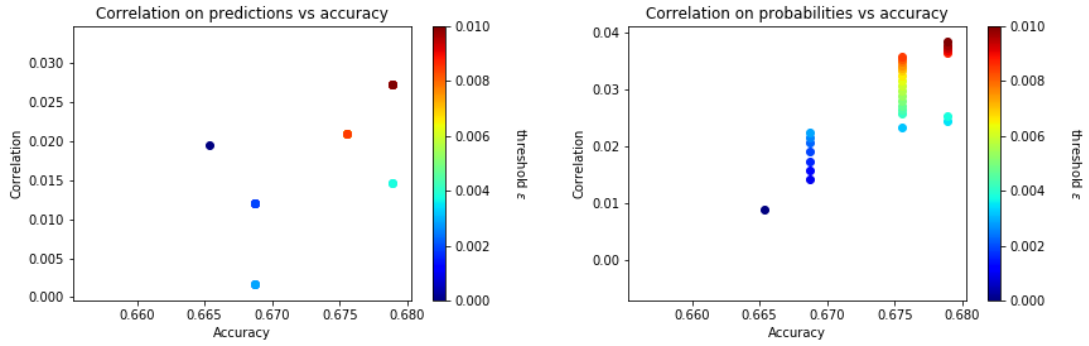


Figure 6.4: STUDENT - Correlation between predictions and sensible variable (left) and between probabilities and sensible variable (right) versus accuracy for different value of the threshold ϵ.

Default On this dataset, results are similar to those above. We see on Figure 6.5 and on Figure 6.6 that the accuracy and the correlation with binary predictions vary in step as it is the case for the dataset *Student*. We also see that the values of correlation with the binary prediction goes up again towards 0.001. On this data set, correlation remains above 1%.

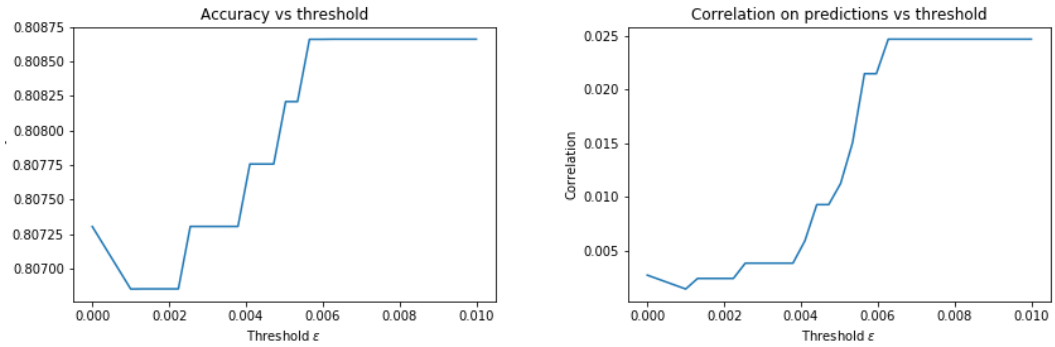


Figure 6.5: DEFAULT - Evolution of the predictions accuracy and of the correlation (in absolute value) between the predictions and the sensible variable as a function of the threshold ϵ on inequality constraints.

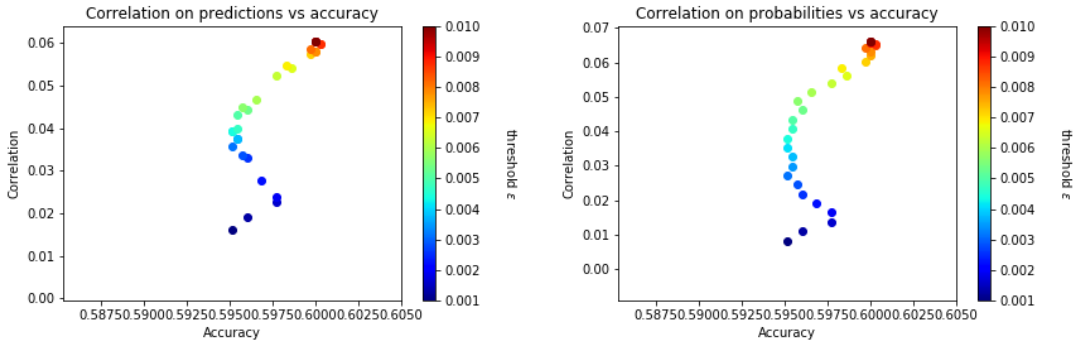


Figure 6.6: DEFAULT - Correlation between predictions and sensible variable (left) and between probabilities and sensible variable (right) versus accuracy for different value of the threshold ϵ.

Table 6.4 shows the best results that were obtained for two distinct criteria : accuracy and correlation with the binary predictions. The baseline is also noted in the table for comparison. The results are really interesting. In all cases, we managed to remove the correlation on the binary prediction almost completely and at a very low cost in terms of accuracy (about 1% at most).

		Accuracy	Correlation binary	Correlation probability
COMPAS	baseline	0.6789	0.2576	0.3131
	best acc.	0.6672	0.0185	0.0429
	best corr.	0.6645	9.37e-5	0.0234
Student	baseline	0.6789	0.0273	0.0396
	best acc.	0.6789	0.0146	0.0242
	best corr.	0.6688	0.0017	0.0204
Default	baseline	0.8087	0.0247	0.0442
	best acc.	0.8087	0.0247	0.0356
	best corr.	0.8069	0.0014	0.0020

Table 6.4: Summary of the results for the model with threshold constraints. For each data set, *best acc.* is the result for the value of the threshold that produced the highest accuracy and *best corr.* is the result for the value of the threshold that produced the lowest correlation on the binary predictions.

6.4 Results for post processing methods

Two post processing approaches were introduced in section 4.4. The first one consists in making predictions on both the training and the test set and then optimize a least square on all those predictions with fairness constraints on the covariance. The second one is similar but makes predictions on the test set only and then optimize a least square on those predictions only.

6.4.1 Results of the first approach

COMPAS On Figure 6.7 we can observe that this model also trades accuracy for low correlation. Now there is a monotonic decrease with the correlation on the binary predictions compared to the previous model where we had a value from which the correlation was going up again. On this dataset, correlation could not be completely removed.

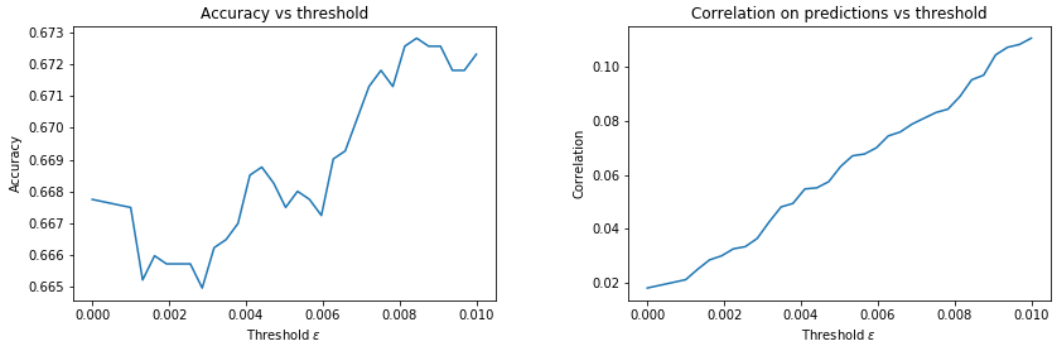


Figure 6.7: COMPAS - Evolution of the predictions accuracy and of the correlation (in absolute value) between the predictions and the sensible variable as a function of the threshold ϵ on inequality constraints.

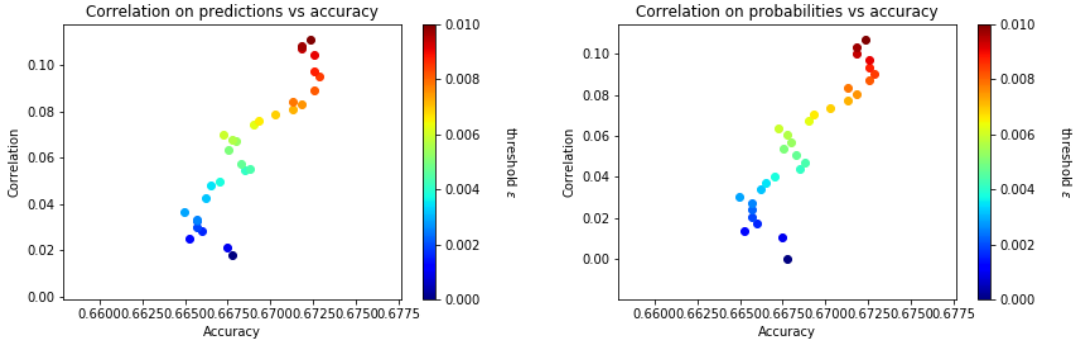


Figure 6.8: COMPAS - Correlation between predictions and sensible variable (left) and between probabilities and sensible variable (right) versus accuracy for different value of the threshold ϵ.

Student On the Student dataset, we observe that the predictions seem to change by intervals (Figure 6.10 and 6.9) which is the same observation that we made for the model that imposes threshold constraint. The correlation could not be completely removed even when the threshold is set to 0.

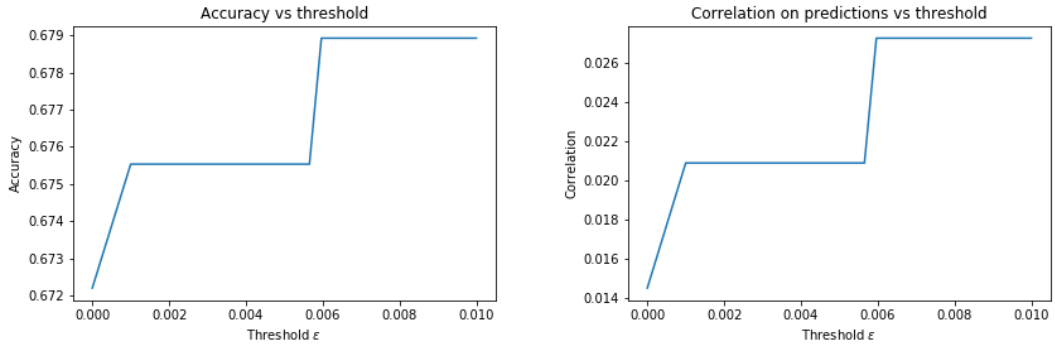


Figure 6.9: STUDENT - Evolution of the predictions accuracy and of the correlation (in absolute value) between the predictions and the sensible variable as a function of the threshold ϵ on inequality constraints.

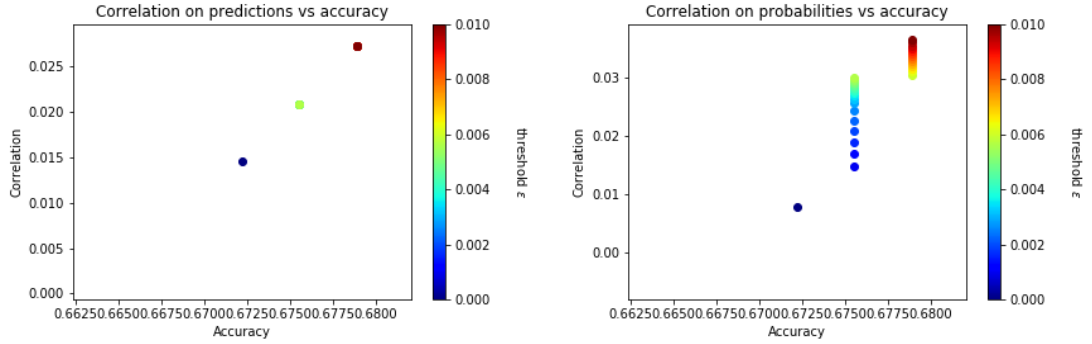


Figure 6.10: STUDENT - Correlation between predictions and sensible variable (left) and between probabilities and sensible variable (right) versus accuracy for different value of the threshold ϵ.

Default The results for this dataset are reported in the Appendix B.2 as they are similar to the previous ones.

		Accuracy	Correlation binary	Correlation probability
COMPAS	baseline	0.6789	0.2576	0.3131
	best acc.	0.6728	0.0953	0.0904
	best corr.	0.6678	0.0181	0.0002
Student	baseline	0.6789	0.0273	0.0396
	best acc.	0.6789	0.0273	0.0306
	best corr.	0.6722	0.0145	0.0078
Default	baseline	0.8087	0.0247	0.0442
	best acc.	0.8091	0.0147	0.0222
	best corr.	0.8073	0.0038	0.0027

Table 6.5: Summary of the results for the approach using **post-processing first approach**. For each data set, *best acc.* is the result for the value of the threshold that produced the highest accuracy and *best corr.* is the result for the value of the threshold that produced the lowest correlation on the binary predictions.

6.4.2 Results of the second approach

For this approach, we only present the summary table in the paper and the graphs can be found in Appendix B.2.2. The same conclusion that were drawn for the first approach can be drawn here.

Table 6.6 shows a summary of the results for each data set. It is interesting to note that with the method with the threshold constraints, we managed to remove the correlation on the binary predictions whereas here we were able to remove the correlation on the probabilities. With this approach, the covariance constraint is directly on the predictions of the test set which is why it is able to fully remove the correlation with the probabilities. For the model with the threshold constraint, constraining the binary predictions will result in a non-linear constraint but with the post-processing approach we could try to constraint the covariance with the binary predictions and it would be interesting to compare those results with the model with the threshold constraints.

		Accuracy	Correlation binary	Correlation probability
COMPAS	baseline	0.6789	0.2576	0.3131
	best acc.	0.6705	0.0948	0.1007
	best corr.	0.6680	0.0204	0.1.31e-9
Student	baseline	0.6789	0.0273	0.0396
	best acc.	0.6722	0.0139	7.71e-11
	best corr.	0.6688	0.0071	0.0018
Default	baseline	0.8087	0.0247	0.0442
	best acc.	0.8091	0.0148	0.0210
	best corr.	0.8073	0.0038	1.96e-10

Table 6.6: Summary of the results for the approach using **post-processing second approach**. For each data set, *best acc.* is the result for the value of the threshold that produced the highest accuracy and *best corr.* is the result for the value of the threshold that produced the lowest correlation on the binary predictions.

6.5 Combining pre-processing with post-processing

Pre-processing the data matrix using projection and both post-processing approaches that we presented are not exclusive, they can be done simultaneously. Recall that we proposed two different projection approach, one where we project using the training set and test set together and one where we do it separately for both sets. Here, the test were made with the projection of both sets together.

6.6 Pre-processing and post-processing applied to other classifiers

One shortcoming of our maximum entropy formulation is that it is non parametric. As a result, one can not tune any meta-parameter to improve the results as it is often the case with other classifiers. We can easily compare our MaxEnt model with a random forest classifier. Indeed, the idea behind our post-processing model is that it can be used as a blackbox filter after the prediction phase. Let us compare the MaxEnt model with sklearn's random forest, first with default meta-parameter and then with a model that has been tuned to increase accuracy's performance.

To tune the performance of the Random Forest classifier, randomized search and grid search were used. The idea is to select several specific values for each of the parameters that we want to tune. Then, randomized search will first try

some of the possible combinations of the parameters setting and output the best combination. This gives a first grasp of the range of values that will improve the model's performance and we will build a second set of parameter's range based on this result. Grid search will then test every combination of the parameters setting that it got as input. All results are cross-validated and this allows us to boost the performance of our model through a good choice of the meta-parameters.

Table 6.7 shows the result for complete decorrelation with the first post-processing approach that we presented earlier. We see that by adjusting the parameters of random forest, we were able to improve the performance of random forest by 6% in terms of accuracy and we were able to reduce the correlation as well. In this case, the tuned random forest model does also slightly better than the maximum entropy model.

From Table 6.8 we can see similar results for the pre-processing. Again, we see that random forest with default parameters was outperformed by the maximum entropy model but the adjustment of those parameters made the tuned random forest classifier more accurate than MaxEnt.

This highlights the interest of parametric models. A counter-benefit is that adjusting the meta-parameter is quite costly from a computational point of view, even more when one resorts to grid search. Moreover, it is not an exact science, there is no procedure that will always lead to a very good set of meta-parameters which makes this step of the model selection quite difficult.

	Accuracy	Correlation on predictions	Correlation on probabilities
MaxEnt	0.6675	**0.0212**	0.0106
RF	0.6233	0.0627	0.0417
Tuned RF	**0.6685**	0.0860	**0.0085**

Table 6.7: COMPAS - Comparison between MaxEnt and Random Forest when requiring complete decorrelation with **post processing** first approach.

	Accuracy	Correlation on predictions	Correlation on probabilities
MaxEnt	0.6619	**0.0272**	**0.0098**
RF	0.6362	0.1765	0.1994
Tuned RF	**0.6695**	0.2476	0.3170

Table 6.8: COMPAS - Comparison between MaxEnt and Random Forest trained with **pre-processed** data matrix.

6.7 Equivalence with logistic regression

In this section we answer one of the research question of this thesis : *Can we empirically show equivalence between logistic regression and a method based on a maximum entropy argument ?*

Earlier, we have demonstrated that the model using maximum entropy is equivalent to a simple logistic regression model. In this section, we will empirically show that this is verified.

We compare our MaxEnt implementation with sklearn[1]'s logistic regression. Logistic regression from sklearn uses a slightly different objective function than the logistic regression that we introduced in Section 3.1.

$$\min_{w,c} \frac{1}{2} w^T w + C \sum_{i=1}^{n} \log(\exp(-y_i(X_i^T w + c)) + 1) \tag{6.1}$$

This is in fact the exact same form with an extra regularization term. To find equivalence between our model, we must set the C parameter to a very large value so that the regularization term become unimportant to the optimization.

Figure 6.11 gives a convincing visual argument that both methods give the same results. Our maximum entropy model and sklearn's logistic regression classifier were trained on the same training set and then predicted the probabilities of being in the + class for the same test set. We can see in Figure 6.11 that plotting the predictions of logistic regression versus those of maximum entropy yield a line $y = x$ which shows that the predictions are the same with both method. In this case, the correlation between the two vectors of predictions is obviously 1.

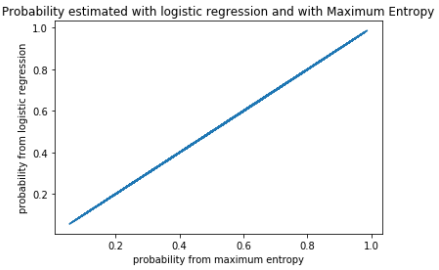


Figure 6.11: COMPAS - Probabilities estimated by logistic regression (from sklearn) versus probabilities estimated by maximum entropy.

[1] `https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html`

6.8 Overall comparison between the models

Let us now compare all the models together. Figures 6.13, 6.14 and 6.15 show the accuracy-correlation trade-off for every dataset. We can see that in every cases, we managed to find models that completely removed the correlation between the sensible variable and the predictions (binary and probability). Most often, this decrease in correlation comes with a decrease in accuracy. However, this decrease is relatively low on our datasets, for instance on *COMPAS*, we were able to remove a correlation of .25 for just 1% of accuracy.

The pre-processing approach using projections does not seem very effective. In some cases, it manages to decrease the correlation but the trade-off seems worse than for the other models. Moreover, in some cases the correlation increases. The fact that it only makes a difference on the linear relationship between the attributes and the sensible variable sure is a drawback of this approach. On the upside, projection is fairly easy to implement and it can be used before any classifier that already exists.

The model that uses threshold constraints (labeled fair in the graphs) shows potential. In every dataset, it offers the lowest correlation on the predictions. The drawback of this method is that it can not be applied as a black box procedure with the already existing classifier as it the case for pre- and post-processing approaches.

Finally, we should ask ourselves if our models behave the same way for every protected groups. For instance, an algorithm could have an 80% overall accuracy while being extremely inaccurate for some sub-populations of the dataset. In our case, all methods have shown similar accuracy across all protected groups. Figure 6.12 shows this for the COMPAS data set and the model with post-processing. The graphs for the other data set and models can be found in the Appendix.

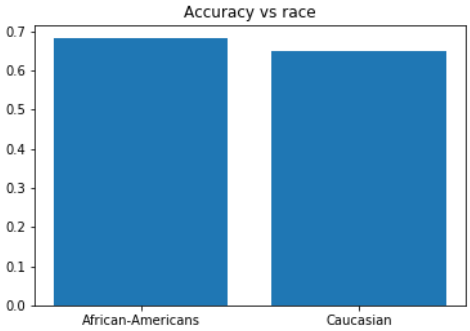


Figure 6.12: COMPAS - Accuracy across all protected groups for the model with post-processing.

We can answer several research questions of this thesis in the light of these results

- *What method offer the best trade-off between fairness and classification performances ?*

It seems that the method with the threshold constraints and the post-processing methods are the most promising one as they manage to decrease correlation at a very low accuracy cost. The results between the two are comparable. The best trade-off is clearly application dependent. In some cases, fairness will be our most important criteria and we would not mind losing a bit of accuracy when in other application it will be the opposite. Both methods have the ability to incorporate more than one definition of fairness which is really interesting since we know that there exists plenty of different measures of fairness and that deciding which is the most appropriate is difficult.

- *Can post-processing achieve a good trade-off between classification performance and fairness? How does supervised method compare with semi-supervised approaches ?*

Semi-supervised approaches and supervised approaches seem to yield very similar results. The advantage of the supervised method is that it constraints our predictions directly so it is able to reduce the correlation no matter the costs which is not the case with the other models.

- *Does a combination of pre-processing and post-processing methods improve the results?*

In some cases, it seems that is has been the case but there is no real tendency as it is definitely not always the case.

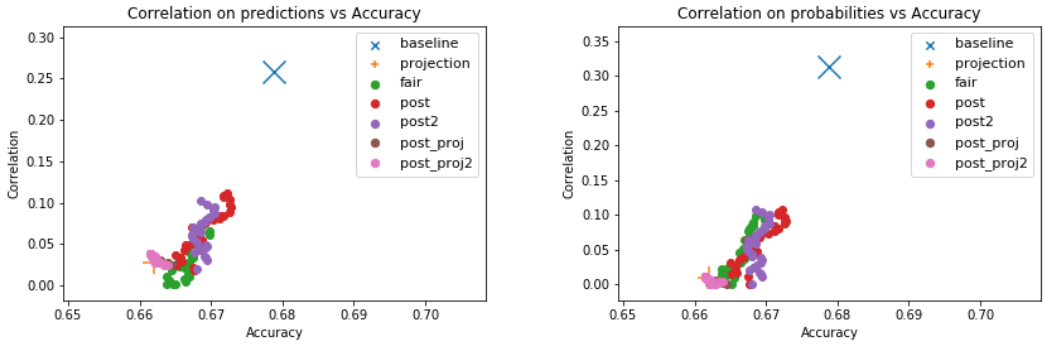


Figure 6.13: COMPAS - Correlation between predictions and sensible variable (left) and between probabilities and sensible variable (right) versus accuracy for different value of the threshold ϵ.

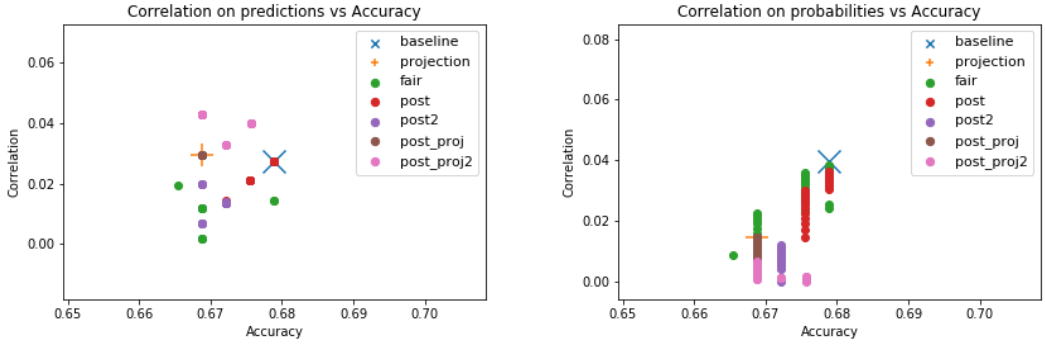


Figure 6.14: STUDENT - Correlation between predictions and sensible variable (left) and between probabilities and sensible variable (right) versus accuracy for different value of the threshold ϵ.

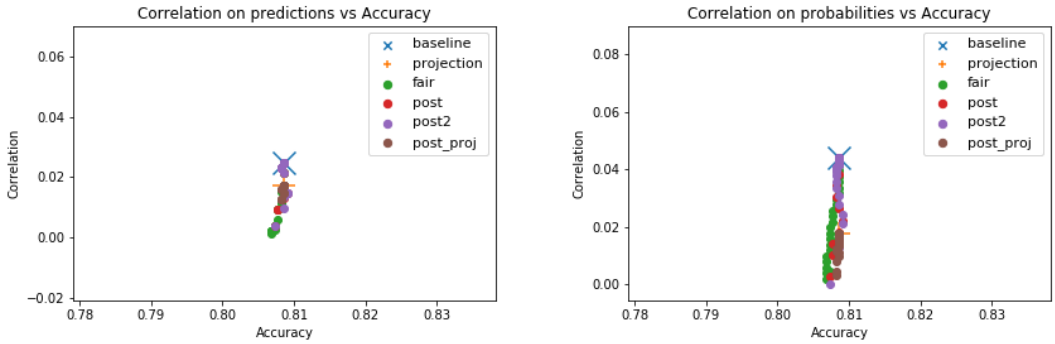


Figure 6.15: DEFAULT - Correlation between predictions and sensible variable (left) and between probabilities and sensible variable (right) versus accuracy for different value of the threshold ϵ.

Chapter 7

Further work and conclusion

Through this paper we have found ways to limit bias in supervised classification algorithms. The sources of bias are numerous and the role that machine learning will play in the medium and long term in decision processes has made this research topic an imminent research branch in only a few years.

After a broad state of the art, we defined the statistical measure we used for our experiments. We used the correlation between the sensitive variables and the predictions to quantify the degree of fairness of our models. We have also shown that this measure is in fact quite close to the disparate impact also called discrimination score which is a quite popular measure in the literature covering the subject thanks to its interpretation and its simple use. We also saw that the solutions already proposed to improve the fairness of classifiers could be grouped in three categories: pre-processing methods, methods that operate during the learning phase and finally post-processing approaches.

We then introduced the maximum entropy model for which we established the equivalence with logistic regression, a very popular model for classification problems. This model is efficient and allows to easily introduce linear constraints such as covariance constraints. From there, we defined different promising models to reduce the correlation between the predictions and the sensitive variable. There are three categories of methods to improve fairness and we proposed methods for each category.

We used projection methods to remove the linear correlation that exists between the sensitive variable and the other variables in the dataset. On the upside, this method can act as a black box before any classification algorithm already implemented. The drawback is that it only accounts for linear relationship between the sensible

attribute and the other features in the data set. When a non-linear relationship exists, this method often fails to mitigate the bias.

For the methods that operate during learning, we proposed to use the maximum entropy model and to incorporate linear constraints on the covariance. This method has shown promising result on our tests. It has been able to reduce the correlation with the binary predictions to zero, trading little accuracy in the process. The problem is that there exists no guarantee that the correlation will fall to zero for a certain threshold.

Finally, for the post-processing methods, we proposed to optimize a least square to constrain the covariances of our predictions with the sensible attribute. Two different approaches were suggested, both showed similar and promising results. There are two main advantages for those methods. First, they can operate as black box which makes them usable with pre-existing implementations without having to change anything. Secondly, the second approach offers guarantee that the correlation with the probabilities will fall to zero. The downside is that those methods are known to be sub-optimal as they trade to much accuracy compared to cutting-edge fairness methods.

Finally, the methods we proposed all performed well in some way and all managed to mitigate the bias initially present in the predictions. These methods are also relatively simple to implement. The post-processing methods and the method with threshold constraints also have the advantage of being able to incorporate different fairness measures and to be able to use several at the same time. We know that it is very difficult to choose the most appropriate measure, so this feature is very interesting.

Further work. In this paper, we limited ourselves to the binary classification task with one sensitive attribute. A first improvement in the continuity of this paper would be to extend and evaluate how the methods we have presented behave in the context of multiple classification and with several sensitive attributes.

Another possible improvement would be to add other fairness constraints to the MaxEnt model. The covariance constraints that we used was very close to the disparate impact measure but there exists other measures that might be suitable for the MaxEnt formulation. It would be interesting to see if the results are also promising for those measures and to see if we could use multiple measures at the same time.

Finally, the MaxEnt model that we used is non parametric (aside from the thresh-

old). It would be interesting to see if we could add meta-parameters to the model such as regularization parameters to try to enhance the performances.

Bibliography

[1] Richard Berk, Hoda Heidari, Shahin Habbari, Michael Kearns and Aaron Roth. Fairness in criminal justice risk assessments: The state of the art. *Sociological Methods & Research*, 2018.

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

[3] Matthew Joseph, Michael Kearns, Jamie H Morgenstern and Aaron Roth. Fairness in learning: Classic and contextual bandits. *Advances in Neural Information Processing Systems*, pages 325-333, 2016.

[4] E.T. Jaynes. Information Theory and Statistical Mechanics. *Physical Review : Series II*, pages 620-630. 1957.

[5] Latanya Sweeney. Discrimination in online ad delivery. *Queue*. 2013.

[6] Joy Buolamwini and Timnit Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. 2018.

[7] Adel Abusitta, Esma Aïmeur, Omar Abdel Wahab. Generative Adversarial Networks for Mitigating Biases in Machine Learning Systems. 2020.

[8] Toon Calders, Faisal Kamiran, Mykola Pechenizkiy. Building Classifiers with Independency Constraints. *2009 IEEE International Conference on Data Mining Workshops*. 2009.

[9] Guilherme D. Pelegrina, Renan D. B. Brotto, João M. T. Romano, Romis Attux. A multi-objective-based approach for Fair Principal Component Analysis. 2020.

[10] Valerio Perrone, Michele Donini, Krishnaram Kenthapadi, Cédric Archambeau. Fair Bayesian Optimization. 2020.

[11] Shenghuo Zhu, Xiang Ji, Wei Xu, Yihong Gong. Multi-labelled Classification Using Maximum Entropy Method

[12] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, W. Philip Kegelmeyer. SMOTE: Synthetic Minority Over-sampling Technique. 2002.

[13] Moritz Hardt, Eric Price, Nathan Srebro. Equality of Opportunity in Supervised Learning. 2016.

[14] C.E. Shannon. A Mathematical Theory of communication. *The Bell System Technical Journal*. 1948.

[15] Tomasz Maszczyk. Wlodzislaw Duch. Comparison of Shannon, Renyi and Tsallis Entropy used in Decision Trees. 2008.

[16] D. Klein, C.D. Manning. Maxent Models, Conditional Estimation, and Optimization. 2003.

[17] McKinney, S.M., Sieniek, M., Godbole, V. et al. International evaluation of an AI system for breast cancer screening. *Nature* 577, 89–94 (2020).

[18] Aylin Caliskan, Joanna J Bryson, and Arvind Narayanan. Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334):183–186, 2017.

[19] "Fairness." *Merriam-Webster.com Dictionary, Merriam-Webster*. `https://www.merriam-webster.com/dictionary/fairness`. Accessed 20 May. 2021.

[20] Stephen Boyd, Lieven Vandenberghe. Convex Optimization. U.K., Cambridge:Cambridge Univ. Press, 2004.

[21] A. Asuncion, D. Newman. UCI machine learning repository. 2007.

[22] Irene Y. Chen, Fredrik D. Johansson, David Sontag. Why is my classifier discriminatory. 2018.

[23] Alexandra Chouldechova, Aaron Roth. The Frontiers of Fairness is Machine Learning. 2018.

[24] C. D. Manning, P. Raghavan, and H. Schütze. Introduction to Information Retrieval, pp. 155. Cambridge University Press, 2008. `https://nlp.stanford.edu/IR-book/pdf/irbookonlinereading.pdf`

[25] Ludovic Lebart, Alain Morineau, Marie Piron. *Statistique exploratoire multidimensionnelle*, pp.319-326. Dunod, 1995.

[26] Andrew Hayes. *Introduction to mediation, moderation and conditional process analysis.*, pp.69-75. The Guilford Press, 2013.

[27] F. Fouss, M. Saerens. Yet another method for combining experts opinions: A maximum entropy model. *Proceedings of the 5th International Workshop on Multiple Classifier Systems (MCS 2004)*. Lecture Notes in Computer Science, VOL. LNAI3077, Springer-Verlag, pp. 82-91.

[28] Alexander Gerniers. Maximum entropy method for multi-label classification. *Ecole polytechnique de Louvain, Université Catholique de Louvain*, 2018.

[29] Faisal Kamiran, Indre Zliobaite, Toon Calders. Quantifying explainable discrimination and removing illgeal discrimination in automated decision making. *Knowl. Inf. Syst.* 35,3 (2013), 613-644.

[30] Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh and Jun Sakuma. Fairness-Aware Classifier with Prejudice Remover Regularizer *EMCL/PKDD (2) (Lecture Notes in Computer Science)*, Peter A.Flash, Tijl de Bie and Nello Christianini (Eds.), Vol. 7524. Springer, 35-50. 2012.

[31] Redlining Map for Birmingham, Alabama, ca. 1935. *Residential Security Maps 1933-1939.* NARA II RG 195.

[32] EEOC., T. U. Uniform guidelines on employee selection procedures. March 1979.

[33] Calders, T., Verwer, S. Three naive Bayes approaches for discrimination-free classification. *Data Min Knowl Disc* 21, 277–292 (2010). `https://doi.org/10.1007/s10618-010-0190-x`

[34] Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani. An Introduction to Statistical Learning : with Applications in R, pp.130-149. New York :Springer, 2013. `https://static1.squarespace.com/static/5ff2adbe3fe4fe33db902812/t/6009dd9fa7bc363aa822d2c7/1611259312432/ISLR+Seventh+Printing.pdf`

Appendices

Appendix A

Source code

A notebook containing all implementations and graphs presented in this report is available from :

`https://github.com/constdesch/Fairness`

Running the implementation will require:

- Numpy : `https://numpy.org`
- sklearn : `https://scikit-learn.org/`
- pandas : `https://pandas.pydata.org`
- matplotlib : `https://matplotlib.org`
- cvxpy : `https://www.cvxpy.org`
- Mosek licence : `https://www.mosek.com/license/request/?i=acp`

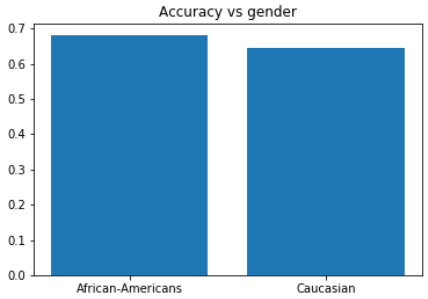
The notebooks are documented, alternatively if you do not have a Mosek licence, you can change the solver for each optimization program of the code.

Appendix B

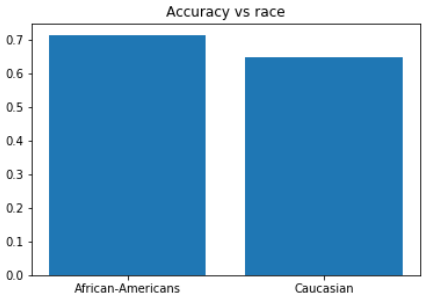
Complementary results

B.1 Model with threshold constraint

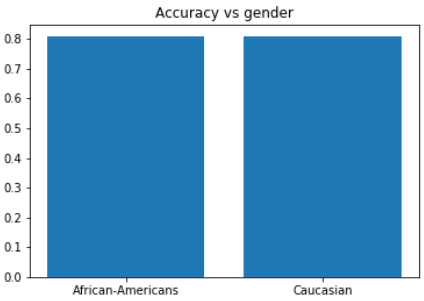
This is the accuracy for each protected groups.



(a) COMPAS



(b) Student



(c) Default

Figure B.1: Accuracy for each protected groups in the Student and Default data set

B.2 Post-processing results

Here are the results that were not presented in the paper for the post-processing approaches.

B.2.1 First approach : result for the Default dataset

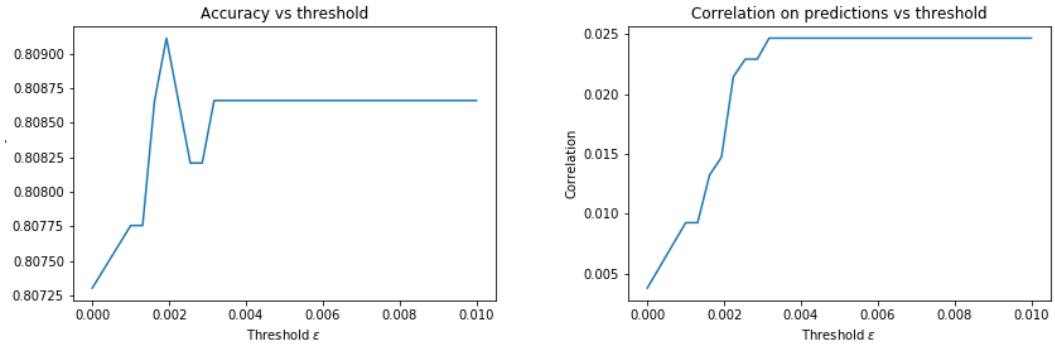


Figure B.2: DEFAULT - Evolution of the predictions accuracy and of the correlation (in absolute value) between the predictions and the sensible variable as a function of the threshold ϵ on inequality constraints.

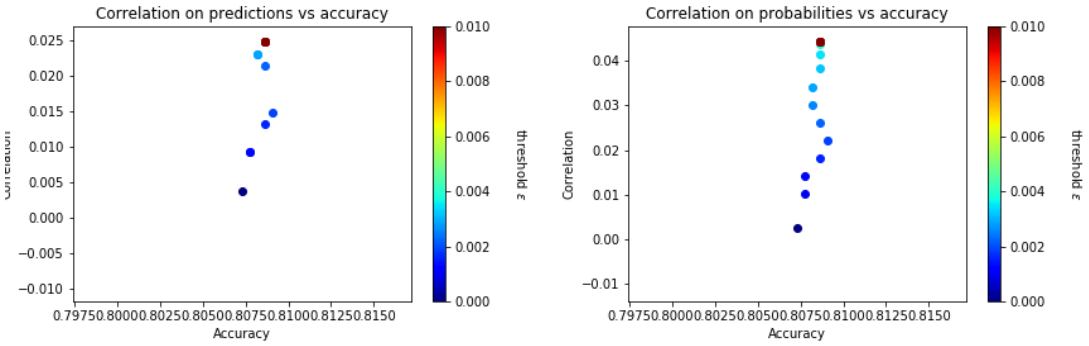


Figure B.3: DEFAULT - Correlation between predictions and sensible variable (left) and between probabilities and sensible variable (right) versus accuracy for different value of the threshold ϵ.

B.2.2 Second approach : results for each dataset

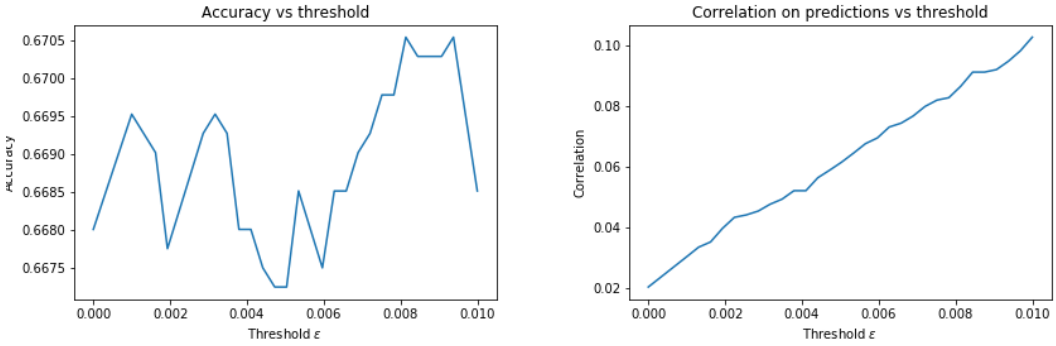


Figure B.4: COMPAS - Evolution of the predictions accuracy and of the correlation (in absolute value) between the predictions and the sensible variable as a function of the threshold ϵ on inequality constraints.

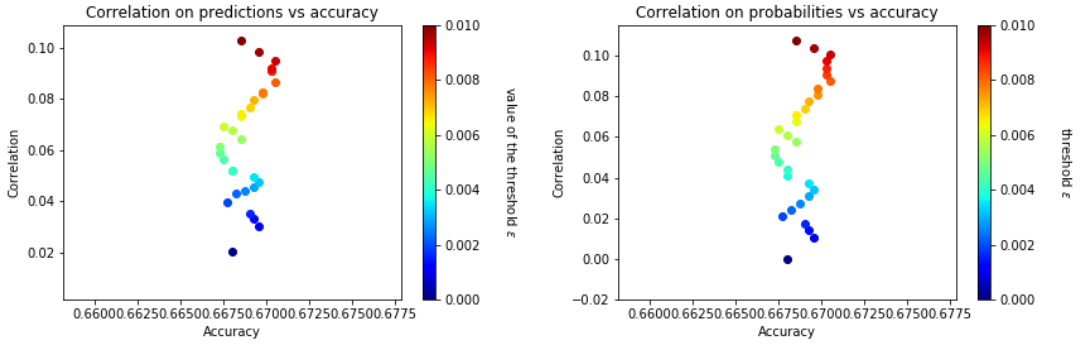


Figure B.5: COMPAS - Correlation between predictions and sensible variable (left) and between probabilities and sensible variable (right) versus accuracy for different value of the threshold ϵ.

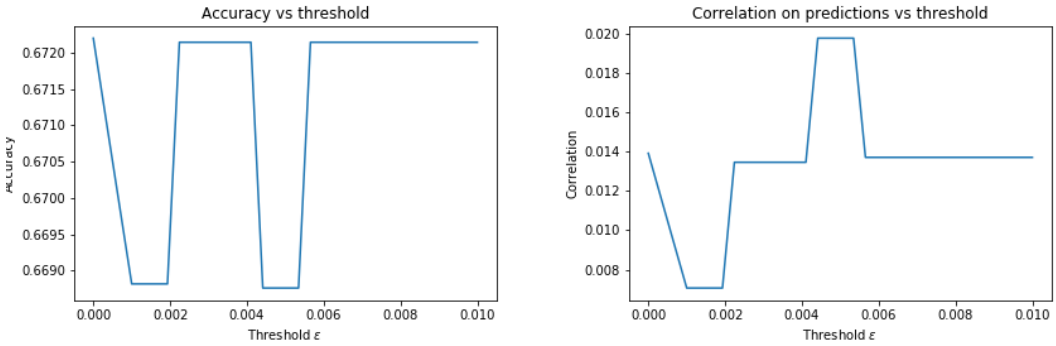


Figure B.6: STUDENT - Evolution of the predictions accuracy and of the correlation (in absolute value) between the predictions and the sensible variable as a function of the threshold ϵ on inequality constraints.

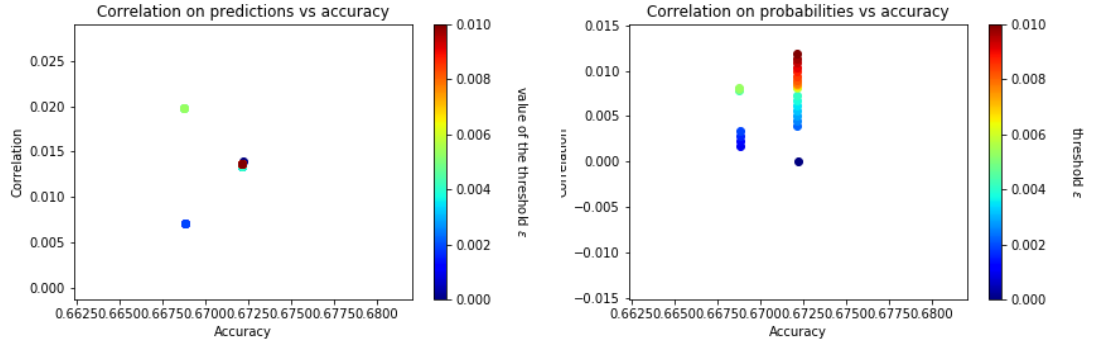


Figure B.7: STUDENT - Correlation between predictions and sensible variable (left) and between probabilities and sensible variable (right) versus accuracy for different value of the threshold ϵ.

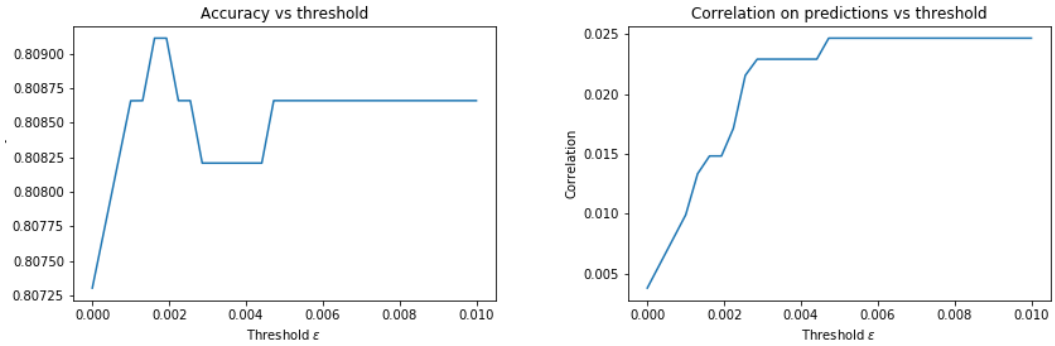


Figure B.8: DEFAULT - Evolution of the predictions accuracy and of the correlation (in absolute value) between the predictions and the sensible variable as a function of the threshold ϵ on inequality constraints.

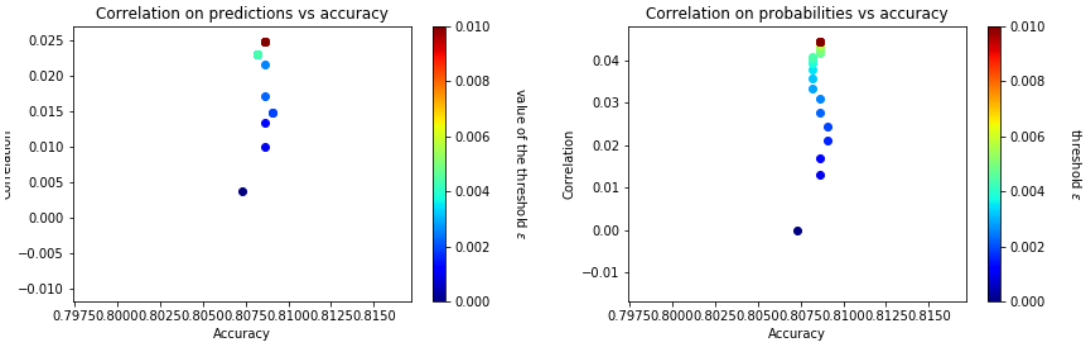
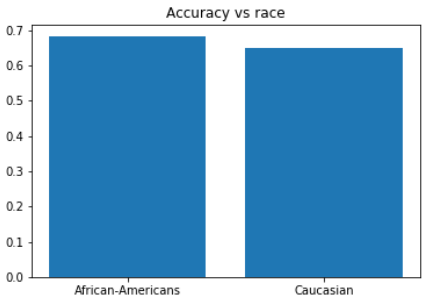
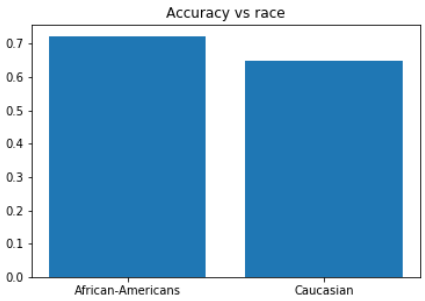


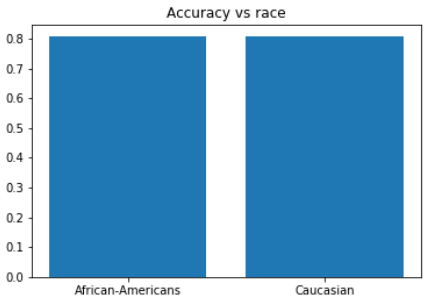
Figure B.9: DEFAULT - Correlation between predictions and sensible variable (left) and between probabilities and sensible variable (right) versus accuracy for different value of the threshold ϵ.



(a) Student

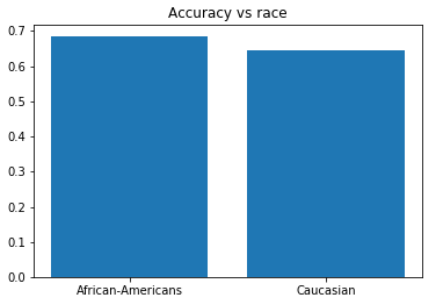


(b) Student

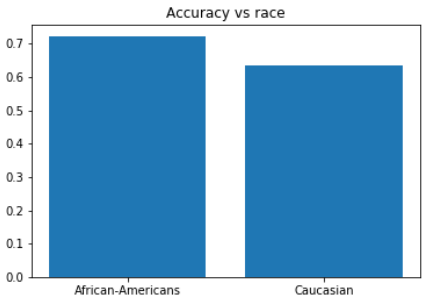


(c) Default

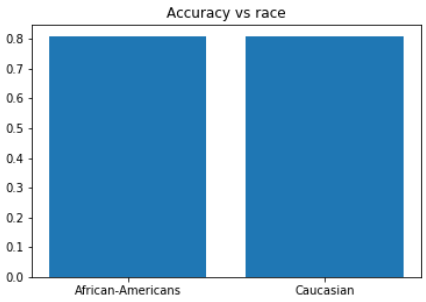
Figure B.10: First approach - Accuracy for each protected groups in each data set



(a) COMPAS



(b) Student



(c) Default

Figure B.11: Second approach - Accuracy for each protected groups in each data set

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
École polytechnique de Louvain
Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve, Belgique | **www.uclouvain.be/epl**