UCLouvain

École polytechnique de Louvain

Fairness in supervised classification: a comparison between a naïve, a GAN-based, and some other methods

Author: **Thomas Bounoider**
Supervisor: **Marco Saerens**
Readers: **Siegfried Nijssen, Sylvain Courtain**
Academic year 2022–2023
Master [120] in Computer Science and Engineering

Abstract

Concerns about biased results in machine learning models, especially in supervised classification, have led to extensive research on fairness strategies. This thesis presents a comprehensive comparison of some different fairness methods, including Pre-processing, In-processing and Post-processing approaches, as well as a naïve strategy and a last one based on a Generative Adversarial Network (GAN). The effectiveness of these methods in reducing discrimination without significantly compromising accuracy is evaluated on five datasets. The experiments show that the Pre-processing, In-processing and Post-processing strategies show promising results in reducing discrimination while maintaining reasonable accuracy. The naïve strategies show interesting but unreliable results in the reduction of the discrimination. The performance of the GAN-based strategy, although inconsistent, offers opportunities for improvement.

Keywords: Fairness, Supervised Classification, Machine Learning, Bias, Pre-processing, In-processing, Post-processing, Naïve, GAN

Acknowledgements

First of all, I would like to thank my thesis supervisor Professor M. Saerens for his continuous help throughout the realisation of this thesis. This work would not have been possible without him.

I would also like to thank the different persons who were involved in this project: S. Courtain for his follow-up and as a reader of this thesis; Professor S. Nijssen also as a reader; C. de Schaetzen, A. Kneip and E. Beghein for their valuable code and master's thesis.

Computational resources have been provided by the supercomputing facilities of the Université catholique de Louvain (CISM/UCL) and the Consortium des Équipements de Calcul Intensif en Fédération Wallonie Bruxelles (CÉCI) funded by the Fond de la Recherche Scientifique de Belgique (F.R.S.-FNRS) under convention 2.5020.11 and by the Walloon Region.

Contents

Chapter 1

General introduction

In recent years, there has been growing concern about the potential for biased results in machine learning models, particularly in the area of supervised classification. This is because these models are often used to make important decisions that can have a significant impact on individuals, such as determining creditworthiness or predicting recidivism. If the training data used to build these models is biased, then the resulting predictions may also be biased, leading to unfair outcomes for certain groups.

For example, a model trained to predict loan defaults may be more likely to flag a loan application as high risk if the applicant is of a certain race or gender, even if they are otherwise qualified for the loan. This could lead to discrimination against these groups, with financial and social consequences.

However, human bias directly encoded in the training data is not the only source of unfairness. It can also be linked to the algorithm itself, which can be biased towards maximising a particular parameter at the expense of fairness.

Given the importance of fairness in machine learning, much research and methods have been developed to improve the fairness of supervised classification models. In this thesis, we will compare and analyse different techniques that have been proposed to address fairness in machine learning, with a focus on supervised classification and the concept of demographic parity.

Some of the methods we will explore include pre-processing techniques, such as data cleaning and balancing, and post-processing methods, such as re-weighting and recalibration. We will also examine in-processing methods, which involve modifying the learning algorithm itself to incorporate fairness considerations. These three techniques are the standard methods proposed by C. de Schaetzen [12], A. Kneip and E. Beghein [25], three former students. They were also supervised by M. Saerens [39, 40].

In addition, we selected a GAN-based method directly from the literature, which may provide interesting results to analyse. We also implemented a simpler and more naïve strategy that can lead to fairness with reduced computational cost.

In this thesis we will compare these methods and evaluate their effectiveness in improving fairness, both in terms of reducing bias and without reducing the overall accuracy of the model too much. We will do this by applying these methods to a variety of datasets and comparing the resulting models using a range of fairness metrics.

Overall, our goal is to provide a comprehensive overview of some of the current state of the art in fairness in supervised classification and to identify areas for future research and development. In doing so, we hope to contribute to the ongoing effort to create machine learning models that are both accurate and fair.

Contributions. This work contributes to (1) the analysis of Pre-, In- and Post-processing fairness strategies, (2) the investigation of a GAN-based fairness strategy, (3) the investigation of a naïve fairness strategy, and (4) the experimental comparison of all previously mentioned fairness strategies on five datasets.

Content. The chapters are organised as follows. First, the chapter 2 summarises some theory about fairness in predictive modelling. Then, the chapter 3 presents the fairness strategies studied. Finally, the chapter 4 describes our methodology and the chapter 5 presents the results of our experiments.

Chapter 2

Fairness notions and definitions in predictive modelling

As the title suggests, this chapter summarises what we need to know about fairness in predictive modelling, from the basic concept to the mathematical definition. Fairness is a broad subject and is not even limited to predictive modelling.

2.1 Problem introduction

As previously mentioned, we only consider supervised classification problems. Moreover, in this master thesis we focused ourselves on binary supervised classification problems. Let us introduce the elements of the classification problem [5].

- $\mathbf{X}$ is a $n \times m$ data matrix. It represents the data without the labels column $\mathbf{y}$, but with the sensitive column $\mathbf{z}$. It consists of n instances of m features.
- $\mathbf{y}$ is a column vector of size n. It represents the original data labels (or targets). This is what the models need to predict. Furthermore, this column vector $\mathbf{y}$ is made up of n observations of the random variable Y. In general, Y consists of q mutually exclusive classes. In this work, we will restrict it to two classes, since we are focusing on binary classification problems. This means that Y can only take two values, either 0 or 1. The target is then binary and $Y = 1$ is the class of interest.
- $\mathbf{z}$ is also a binary column vector of size n. In general, it is a $n \times p$ data matrix made of n instances of p protected attributes, but in this work, we will only work with one protected variable. The corresponding binary random variable is Z.
- $\hat{Y}$ is another random variable. It corresponds to the output score produced by the classification model. It estimates the predicted a posteriori probabilities of belonging to the positive class $Y = 1$.
- $\hat{Y}_d$ is also a random variable. It represents the decisions. This is the binary output of the classification model.

2.2 Source of unfairness

Sources of unfairness can be of many types and located on different parts of the model. We will present three types of sources [29] that are linked to the dataset. However, sources of unfairness can also be found during model training or when the model makes decisions.

- **Human bias directly encoded inside a dataset**
  Historical or societal bias are sometimes directly encoded inside a dataset. If a model is fitted on this dataset, it will be likely to replicate the bias when making decisions. This is often the case when the decision label is based on an arbitrary or subjective human judgement.

- **Unbalanced dataset**
  A dataset is unbalanced when it does not correctly reflect the full representation of the population. This often happens when a sample of the data is selected. If one type of instance is present in majority in the data, the model decisions can become really unfair to other minority types of instances.

- **Model variables highly correlated with the protected one**
  In fair Machine Learning, protected variables are sometimes separated because they are likely to be sources of unfairness. But « normal » variables can also cause unfairness. They can act as a proxy for a protected variable if they are correlated with it even if that protected variable has been deleted.

2.3 Different operational notions of fairness

Giving a general definition of fairness is a difficult task that has not yet been accomplished. In the recent years many different notions of fairness have been proposed, each with reference to a specific application [7, 8, 16, 22, 24, 28, 32, 43, 50].
This section focuses only on statistical definitions of fairness. These definitions induce fairness between the dataset and the protected attribute based on statistical measurements. Many other types of definitions exist, but investigating them is left for future works. Here are some common definitions [39, 43] (note that the terminology is not universal and might change from one source to another).

- **Demographic, or statistical, parity (DP)**
  This notion implies that a classifier should assign the same probability of being in the positive output class to both sensitive and non-sensitive groups. This definition is based on predicted output only. For example, anyone applying for a job should have the same probability of getting the job, regardless of their gender.

- **Conditional demographic parity (CDP)**
  This notion is similar to the previous one as it also implies that a classifier should assign the same probability of being in the positive output class to both sensitive and non-sensitive groups, but it is controlled by a set of attributes. For example, if the controlling attributes are age and minimum degree, then anyone who meets these two attributes and applies for a job offer should have the same probability of getting the job, regardless of their gender.

- **Equal opportunity (EO)**
  This concept implies that a classifier should assign the same true positive rate (TPR)

to both sensitive and non-sensitive groups. This definition is based on the predicted performance and the actual label. For example, someone who actually meets the required attributes to get a particular job should have the same probability of being correctly accepted for the job, regardless of their gender.

- **Predictive equality (PE)**
  This concept is similar to the previous one and is also based on the predicted output and the actual label. A classifier should assign the same false positive rate (FPR) to both sensitive and non-sensitive groups. For example, someone who actually meets the requirements for a particular job should have the same chance of being falsely accepted for the job, regardless of their gender.

- **Equalized odds (EOdds)**
  This notion combines the definitions of equal opportunity and predictive equality. A classifier should assign the same false positive rate (FPR) and the same true positive rate (TPR) to both sensitive and non-sensitive groups.

We have decided to focus on two notions of fairness, demographic parity and equal opportunity. These two definitions will now be developed [9], but let us first introduce some basic concepts, such as the confusion matrix. It will be used in the development. With Y the actual label and $\hat{Y}_d$ the predicted class, it is defined as:

	Predicted class: $\hat{Y}_d = 1$	Predicted class: $\hat{Y}_d = 0$
Actual class: Y = 1	True Positives (TP)	False Negatives (FN)
Actual class: Y = 0	False Positives (FP)	True Negatives (TN)

Figure 2.1: Definitions related to the confusion matrix.

From this table, we can define two useful quantities:

- FPR - False Positive Rate : the rate of incorrectly predicted negative instances.

$$FPR = \frac{FP}{FP + TN} \tag{2.1}$$

- TPR - True Positive Rate : the rate of correctly predicted positive instances.

$$TPR = \frac{TP}{TP + FN} \tag{2.2}$$

2.3.1 Demographic Parity

If the output of a model satisfies the demographic parity constraint [9] it more formally means that it minimises the quantity DP.

$$DP = |P(\hat{Y}_d = 1|Z = 1) - P(\hat{Y}_d = 1|Z = 0)| \tag{2.3}$$

And a perfect DP means that:

$$P(\hat{Y}_d = 1|Z = 1) = P(\hat{Y}_d = 1|Z = 0) \tag{2.4}$$

where $\hat{Y}_d$ is the predicted class and Z is the protected attribute. This quantity can be computed as follows according to what has been presented in section 2.3.

$$\begin{gathered} TPR(Z = 1) = TPR(Z = 0) \\ \frac{TP(Z = 1) + FP(Z = 1)}{N(Z = 1)} = \frac{TP(Z = 0) + FP(Z = 0)}{N(Z = 0)} \end{gathered} \tag{2.5}$$

Where $TP(Z = 1)$ means that we compute the quantity only on the sub-sample $Z = 1$ and $N(Z = 1)$ is the count of instances with $Z = 1$. This quantity can also be expressed in terms of predicted probabilities [39] as:

$$\mathbb{E}[\hat{Y}|Z = 1] = \mathbb{E}[\hat{Y}|Z = 0] \tag{2.6}$$

which can be interpreted as the expected a posteriori probability of being assigned to the positive class as a male should be the same as a female. This will be useful when we talk about Post-processing in chapter 3.

2.3.2 Equal opportunity

If a model satisfies the equal opportunity constraint [9] it means, more formally, that it minimises the quantity EO.

$$\begin{gathered} P(\hat{Y}_d = 1|Y = 1, Z = 1) = P(\hat{Y}_d = 1|Y = 1, Z = 0) \\ EO = |P(\hat{Y}_d = 1|Y = 1, Z = 1) - P(\hat{Y}_d = 1|Y = 1, Z = 0)| \end{gathered} \tag{2.7}$$

This can be computed as follows according to what has been presented in section 2.3.

$$\frac{TP(Z = 1)}{TP(Z = 1) + FN(Z = 1)} = \frac{TP(Z = 0)}{TP(Z = 0) + FN(Z = 0)} \tag{2.8}$$

2.3.3 Alternative way to compute the equal opportunity

The equal opportunity quantity uses the actual labels from the dataset in its calculation. This is a problem because we are not supposed to have this information as it is exactly what we want to predict. A solution to address this problem could be to use an approximation of the labels instead of the actual labels. This approximation can simply be the predictions made by a classifier without any fairness, such as our Baseline method. Unfortunately, none of the fairness strategies studied aim to improve the equal opportunity, so it will not make sense to analyse this measure in this work. The study of equal opportunity and this alternative way of computing it are thus left for future works.

2.3.4 Covariance as a fairness metric

This section explains part of the work of previous students [12, 25], the equations below are taken directly from what they have produced based on the submitted papers (Saerens, 2022 [40] and Vancompernolle, 2023 [42]). They used the covariance computation as a metric to evaluate the level of fairness of a classifier, as in Zafar, 2017 [48].

Let us take two random variables X and Y. Then $\mathbf{x}$ and $\mathbf{y}$ are two vectors of n realisations respectively from X and Y. The covariance between the two random variables on a sample of n measurements can therefore be expressed as [41]:

$$\text{cov}(\mathbf{x}, \mathbf{y}) = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y}) \tag{2.9}$$

For a later chapter, this formula will be rewritten by introducing $\mathbf{H}$, a centering operator defined as [38]:

$$\mathbf{H} = \left(\mathbf{I} - \frac{1}{n} \mathbf{e}\mathbf{e}^T \right) \tag{2.10}$$

where $\mathbf{I}$ is a $n \times n$ identity matrix and $\mathbf{e}$ is a vector column full of n 1's [38]. $\mathbf{H}$ has two interesting properties, it is idempotent ($\mathbf{H}^2 = \mathbf{H}$) and symmetric ($\mathbf{H}^T = \mathbf{H}$) [38]. We can now rewrite the covariance formula as:

$$\begin{aligned} \text{cov}(\mathbf{x}, \mathbf{y}) &= \frac{1}{n-1} \sum_{i=1}^{n} [\mathbf{H}\mathbf{x}]_i [\mathbf{H}\mathbf{y}]_i = \frac{1}{n-1} (\mathbf{H}\mathbf{x}) \cdot (\mathbf{H}\mathbf{y}) \\ &= \frac{1}{n-1} (\mathbf{H}\mathbf{x})^T (\mathbf{H}\mathbf{y}) = \frac{1}{n-1} \mathbf{x}^T \mathbf{H}\mathbf{y} \end{aligned} \tag{2.11}$$

2.3.5 Covariance between a categorical and a numerical variable

We now consider X as a random categorical variable and Y as a random numerical variable. More precisely, X corresponds to a binary sensitive variable. This implies that all observed values from the random variable X are divided into two classes: $\mathcal{C}_0$ when $X = 0$ and $\mathcal{C}_1$ when $X = 1$.

We can now define n as the total number of observations, n_0 as the number of observations belonging to $\mathcal{C}_0$ and n_1 as the number of observations belonging to $\mathcal{C}_1$. Then, the a priori probabilities for the two classes can be computed as $\pi_0 = P(X = 0)$ and $\pi_1 = P(X = 1)$. Finally, the sample mean of X is defined as $\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i = \frac{1}{n} n_1 = \pi_1$, where x_i is the i^{th} observation of X.

This was used by C. de Schaetzen [12] to demonstrate the following covariance expression based on the submitted papers (Saerens, 2022 [40] and Vancompernolle, 2023 [42]).

$$\begin{aligned} \text{cov}(\mathbf{x}, \mathbf{y}) &= \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y}) \\ &= \frac{n}{n-1} \pi_0 \pi_1 \left(\frac{1}{n_1} \sum_{i \in \mathcal{C}_1} y_i - \frac{1}{n_0} \sum_{i \in \mathcal{C}_0} y_i \right) \end{aligned} \tag{2.12}$$

Let us then denote $\bar{y}_0 = \frac{1}{n_0} \sum_{i \in \mathcal{C}_0} y_i$ and $\bar{y}_1 = \frac{1}{n_1} \sum_{i \in \mathcal{C}_1} y_i$ and rewrite the equation as follows [12, 40]:

$$\text{cov}(\mathbf{x}, \mathbf{y}) = \frac{n}{n-1} \pi_0 \pi_1 (\bar{y}_1 - \bar{y}_0) \tag{2.13}$$

If we take the example from section 1, a model trained to predict loan defaults. The protected binary variable Z can be the gender and Y would label a loan applicant as high or low risk. If we want men and women to be treated similarly, we need to impose $\bar{y}_1 = \bar{y}_0$, which implies $\text{cov}(\mathbf{x}, \mathbf{y}) = 0$. This is directly related to the Demographic Parity definition, which imposes that the expected a posteriori probability of being assigned to the positive class as a male to be the same as a female. The covariance equation can be rewritten as proportional to [12, 40, 42]:

$$\text{cov}(\mathbf{x}, \hat{\mathbf{y}}) \propto \hat{\mathbb{E}}[\hat{Y}|Z = 1] - \hat{\mathbb{E}}[\hat{Y}|Z = 0] \tag{2.14}$$

where $\hat{\mathbb{E}}$ is the empirical expectation, i.e. the mean. This quantity will be used later in order to improve the demographic parity of classification models.

Chapter 3

Investigated methods

In this chapter, we will investigate comparisons between different mechanisms that improve the fairness notion of demographic parity in supervised learning. All the fairness strategies have been validated through 5-fold cross-validation. The only exception to this is the GAN-based strategy, which makes it impractical due to its architecture, as we will explain below.
As you will see, an optimisation problem has to be solved for the In-processing and the Post-processing methods. We chose CVXPY [1, 15] with the MOSEK solver [3], which is the more powerful solver we found with a free student licence.

3.1 Baseline method

This will be the base reference method. No changes are made to the classifier. We just add a column full of ones to the dataset to represent and integrate the bias term. The model is then fitted on the whole training set with the protected variable. This gives us the reference accuracy and correlation scores on the test set within a 5-fold cross-validation.

3.2 Naïve method

The initial idea for this strategy was to reduce the effect of the protected column in a simple way and see what could be achieved. Simply removing the protected attribute from the training set to achieve protected column blindness during training is not enough. So, the first version of this method was a simple model trained as a baseline model without any fairness. The difference comes during testing where the protected variable of the test set is replaced by 0.5 instead of 1 or 0 (as it is binary). This strategy will be referred to as « Naive05 ».

Two other similar fairness strategies were also tested to gather more results. Firstly, the « NaiveAvg » strategy is the exact same strategy, but we replaced the protected column within the test set with the average over the whole dataset of the binary protected variable. And secondly, the « NaivePrior » strategy, which differs from the other as it assigns two values to the protected column instead of one. In fact, it replaces both privileged and unprivileged classes on the test set by their a priori probabilities on the whole dataset.

This is inspired by the paper of Devin G. Pope and Justin R. Sydnor [30]. The authors remind us that in most cases removing the protected column does not completely remove its effect on prediction. In fact, other columns also have an impact on predictions and may be highly related to the protected column. Thus, focusing only on neglecting

the effect of the protected columns would sometimes lead to no results.

As we did in NaiveAvg, they proposed the idea to substitute the protected value by the means of its column and to apply it not only to the selected protected variable, but also to variables associated with this protected column. For simplicity, let us assume that all the variables are divided into two groups. We define $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$, two random vectors. $\boldsymbol{X}_1$ corresponds to the set of variables that are not correlated with the protected variable, and $\boldsymbol{X}_2$ regroups the other variables, the protected variable and those that are correlated with it. By assuming a simple linear regression model, the random variable Y that we want to predict with the Baseline method, can then be described as the following linear model [30]:

$$Y^{Baseline} = \alpha_0 + \alpha_1 * \boldsymbol{X}_1 + \alpha_2 * \boldsymbol{X}_2 \tag{3.1}$$

The proposed method outcome would then be described as [30]:

$$Y^{Naïve} = \alpha_0 + \alpha_1 * \boldsymbol{X}_1 + \alpha_2 * \overline{\boldsymbol{X}_2} \tag{3.2}$$

with $\overline{\boldsymbol{X}_2}$ another random vector corresponding to the average vector of $\boldsymbol{X}_2$. This proposed strategy would indeed achieve $\boldsymbol{X}_2$ variables blindness, since if we take two samples that differ only on these variables, they will produce the same output.

The problem now is to determine how to define $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ in practice. We decided to define them in two ways by selecting the 2 or 5 variables with the highest absolute covariance with the protected one in the original dataset. This corresponds to the « Naive2 » and « Naive5 » methods respectively.

This is an easy way to apply the proposed strategy and start gathering results, but we can already anticipate some problems that are likely to occur. Indeed, this strategy seems to be very context dependent. If we take two different datasets, dataset D_1 has 10 variables that are highly correlated to the protected one and dataset D_2 has only 3. The Naive2 and Naive5 methods are expected to give a much more visible increase in fairness in the outcomes for D_2. The ideal method should be able to adapt itself to its dataset. This is left for future works.

3.3 Pre-processing method

Pre-processing methods are methods whose purpose is to introduce fairness into the dataset before it is fitted by the classifier, so that the effect of the protected columns on the decisions is reduced. There are several Pre-processing techniques [49], but we decide to use the one proposed by the previous students [12, 25] based on the submitted paper [40] (Saerens, 2022). The purpose of this method is to completely decorrelate the protected column from all other columns. Once this is done, a classifier is trained on it

to make predictions.
This is done by projecting the data matrix $\mathbf{X}$ of the non-protected columns onto the subspace defined orthogonally to the matrix $\mathbf{Z}$ of the protected columns. We can start by defining the orthogonal projection operator on the column space of $\mathbf{Z}$ as [40]:

$$\Pi_{\mathbf{Z}} = \mathbf{Z}(\mathbf{Z}^T\mathbf{Z})^{-1}\mathbf{Z}^T \tag{3.3}$$

We also know that for any column vector v we can write $\Pi_{\mathbf{Z}}v + \Pi_{\perp\mathbf{Z}}v = v$ [40]. This allows us to express the projection on the column subspace orthogonal to $\mathbf{Z}$ as [40]:

$$\Pi_{\perp\mathbf{Z}} = \mathbf{I} - \Pi_{\mathbf{Z}} \tag{3.4}$$

By applying this projection to each column of the matrix $\mathbf{X}$, we obtain:

$$\Pi_{\perp\mathbf{Z}}\mathbf{X} = [\Pi_{\perp\mathbf{Z}}\mathbf{x}^1, \Pi_{\perp\mathbf{Z}}\mathbf{x}^2, ..., \Pi_{\perp\mathbf{HZ}}\mathbf{x}^m] \tag{3.5}$$

In our case, $\mathbf{Z}$ is a centered matrix and consists of only one protected column (the version with multiple columns is left for future works). Formally, if the original dataset is a $n \times m$ matrix, the projection is written as $\Pi_{\perp\mathbf{HZ}}\mathbf{X}$, where $\mathbf{H}$ is the centering operator [38] introduced in section 2.3.4. The projected data are now uncorrelated with $\mathbf{Z}$, $\text{cov}(\mathbf{Z}, \mathbf{x}^i_{\perp\mathbf{HZ}}) = 0$.
This is taken directly from the work of the previous students (see [12, 25, 40] for details), only the bias term has been removed, as it has no meaning here.

3.3.1 Pre-processing variants

Three different Pre-processing projection strategies have been proposed by the previous students [12, 25, 40] in their work:

- **Joined projection**
  One projection is made within the k-fold cross validation for each fold. It is the training set joined to the test set with respect to $\mathbf{z}$ from the training set joined to $\mathbf{z}$ from the test set.

- **Separated projection**
  Two projections are made inside the k-fold cross validation for each fold. The first projection is the training set with respect to $\mathbf{z}$ from the training set. The second projection is the test set with respect to $\mathbf{z}$ from the test set.

- **Whole projection**
  One projection is made outside the k-fold cross validation. This is the entire matrix with respect to the protected variable. The result is then fed into the k-fold procedure.

The second method will be used because it has been shown to be the most effective by the previous students [25, 40].

3.4 In-processing method

This section explains part of the work of previous students [12, 25] based on the submitted papers (Saerens, 2022 [40] and Vancompernolle, 2023 [42]). In-processing methods are methods whose purpose is to introduce fairness constraints into the decisions while fitting the model so that the protected column effect on the decisions is reduced. The method we will use introduces fairness according to the demographic parity definition with a Maximum Entropy [12, 25, 40, 42] based logistic regression classifier, « MaxEnt ».

3.4.1 Maximum Entropy logistic regression

The Maximum Entropy principle [12, 23, 25, 40, 42] defines a constraint for each observed variable and returns a probability distribution that maximises the entropy. For a random variable Y with n classes, it assigns to each of its values the following probability:

$$\begin{aligned} &P(y_k) \geq 0 \quad \forall k \in \{0, ..., n\} \\ &\sum_{k=1}^{n} P(y_k) = 1 \end{aligned} \tag{3.6}$$

The distribution can be further constrained the more information we have about Y. For example, if the expected value and the variance are known, we can add two more linear equations that must be satisfied. These equations are expressed as follows:

$$\sum_{k=1}^{n} a_k P(y_k) = b \tag{3.7}$$

So the more information we have, the more the imprecision is reduced. However, several possible distributions verify these constraints and we have to choose only one of these. Fortunately, the Maximum Entropy principle can be applied, and it says that the best distribution to describe a random variable is the one that uses only what we know about it and makes no further assumptions. It must therefore be the least informative and therefore the most uncertain one.

C.E Shannon (1948) defined the entropy as a measure of the uncertainty of a random variable. For the same random variable Y, the Shannon Entropy is given by [10]:

$$H(Y) = -\sum_{k=1}^{n} P(y_k) \log(P(y_k)) \tag{3.8}$$

Two useful properties of this quantity are that it is strictly concave and that the logarithm makes the probabilities only greater than or equal to zero [10]. Combined with the equation of the Maximum Entropy principle, we can define a maximisation problem to

solve [12, 25, 40, 42].

$$\begin{aligned} \max_{P(y_k)} \quad & -\sum_{k=1}^{n} P(y_k)\log(P(y_k)) \\ \text{s.t.} \quad & \sum_{k=1}^{n} a_{kc}P(y_{kc}) = b_c \quad \forall c \in \mathcal{C} \\ & \sum_{k=1}^{n} P(y_k) = 1 \end{aligned} \tag{3.9}$$

Where $\mathcal{C}$ represents the set of linear constraints.

3.4.2 Maximum Entropy logistic regression classifier

The definition of Maximum Entropy can now be translated to be used in a classifier. This classifier will be referred to as « MaxEnt ». Note that in this work we have restricted all fairness problems to binary classifications. We have the following data to define our classification problem:

- The matrix $\mathbf{X}$ represents the dataset without labels. It has n instances with m attributes. The first attribute is added. It is the bias column, a column full of 1's introduced to integrate the bias term.
- $P(Y = y_i) = \hat{y}_{ik}$ is the predicted probability that individual i belongs to class k.
- y_{ik} is the observed class (0 or 1).

The general Maximum Entropy logistic regression classification problem can thus be formulated as [12]:

$$\begin{aligned} \max_{P(y_k)} \quad & -\sum_{i=1}^{n}\sum_{k=0}^{1} \hat{y}_{ik} log(\hat{y}_{ik}) \\ \text{s.t.} \quad & \sum_{i=1}^{n} \hat{y}_{ik}\hat{x}_{ij} = \sum_{i=1}^{n} y_{ik} \qquad \forall j \in \{0, ..., m\}, \forall k \in \{0, 1\} \\ & \sum_{k=1}^{q} \hat{y}_{ik} = 1 \qquad \forall i \in \{1, ..., n\} \end{aligned} \tag{3.10}$$

where q is the number of classes. This formulation corresponds to a logistic regression and it is solved during the training part of the classifier. Once solved, it can make predictions on the test set by computing the Lagrangian parameters of the maximisation problem [12, 25, 40, 42].

3.4.3 Including fairness in MaxEnt

We want this In-processing strategy to include fairness according to the demographic parity definition. This means that modified a posteriori probabilities of belonging to class 1, $\hat{Y}$ must satisfy:

$$\begin{aligned} |\mathbb{E}(\hat{Y}|Z=0) - \mathbb{E}(\hat{Y}|Z=1)| \leq \epsilon \\ \mathbb{E}(\hat{Y}|Z=0) - \mathbb{E}(\hat{Y}|Z=1) \leq \epsilon \end{aligned} \tag{3.11}$$

without absolute value and provided that $Y = 1$ is the favourable class and $Z = 1$ is the protected class. As explained in section 2.3.1, this demographic parity quantity can be indicated by the covariance as:

$$\text{cov}(\mathbf{z}, \hat{\mathbf{y}}) = \frac{1}{n-1}\mathbf{z}^T\mathbf{H}\hat{\mathbf{y}} = \frac{1}{n-1}\dot{\mathbf{z}}^T\hat{\mathbf{y}} \tag{3.12}$$

where $\dot{\mathbf{z}}$ is the standardised $\vec{\mathbf{z}}$. The absolute value is split to keep it linear. Then, once added to the previous maximisation problem, it gives [40]:

$$\begin{aligned} \max_{P(y_k)} \quad & -\sum_{i=1}^{n}\sum_{k=0}^{1} \hat{y}_{ik} log(\hat{y}_{ik}) \\ \text{s.t.} \quad & \sum_{i=1}^{n} \hat{y}_{ik} x_{ij} = \sum_{i=1}^{n} y_{ik} \qquad \forall j \in \{0, ..., m\}, \forall k \in \{0, 1\} \\ & \frac{1}{n-1}\sum_{i=1}^{n} \dot{z}_i \hat{y}_{ik} \leq \epsilon \qquad \forall k \in \{0, 1\} \\ & \frac{1}{n-1}\sum_{i=1}^{n} \dot{z}_i \hat{y}_{ik} \geq -\epsilon \qquad \forall k \in \{0, 1\} \\ & \sum_{k=1}^{1} \hat{y}_{ik} = 1 \qquad \forall i \in \{1, ..., n\} \end{aligned} \tag{3.13}$$

If the problem is feasible, the classifier can make predictions by using the Lagrange parameters (see [12, 25, 40, 42] for details).

3.4.4 Maximum Entropy with a covariance minimisation constraint

The In-processing technique increases fairness in multiple steps by varying the ϵ within the fairness constraint. This allows us to collect more data and better analyse the accuracy and correlation trade-off. The range of ϵ selected is then essential to obtain reliable data. For each dataset, the initial ϵ value is set to the original covariance between Z the protected attribute and Y the data labels. The last ϵ was initially set to a small value such as $10e^{-4}$, but it did not provide good results for all datasets because the optimisation problem can become infeasible. In fact, the ϵ value could be set lower.

So we used Mosek [3] to minimise the covariance between Z and Y for each dataset, which provides the lower bound. This was done with a Maximum Entropy constraint [12, 25, 40, 42] and a minimisation objective function. We then set 8 intermediate steps for a total of 10, each of which progressively increases fairness.

3.5 Post-processing method

Post-processing methods are methods whose purpose is to introduce fairness into the predictions after they have been fitted by the classifier, so that the effect of the protected column on the decisions is reduced. There are several Post-processing techniques, but we decided to adapt the one proposed by M. Hardt et al [22], which was then modified by the previous students [12, 25, 40]. This strategy fits a classifier and generates predictions. A least squares optimisation problem is then solved with constraints on the covariance between the protected variable and the generated probabilistic predictions. Solving this problem yields a modified version of the predictions that includes fairness while remaining closest to the original version. This will result in a drop in accuracy, which we want to keep as small as possible.

3.5.1 The investigated method

The constraints applied aim at satisfying the demographic parity definition to achieve decorrelation between the protected attribute $\mathbf{z}$ and the probabilistic predictions $\hat{Y}$. The strict equality is relaxed so that the quantity must be lower than a given threshold ϵ. We have $\hat{Y}$ the a posteriori probabilities of belonging to class 1 provided by the classification model and we define $\tilde{Y}$ (tilde) the modified a posteriori probabilities of belonging to class 1. This can be achieved exactly as we did in section 3.4. $\tilde{Y}$ must satisfy:

$$|\mathbb{E}(\tilde{Y}|Z=0) - \mathbb{E}(\tilde{Y}|Z=1)| \leq \epsilon \tag{3.14}$$

Then, the demographic parity quantity is replaced by the covariance as (see equation 2.11):

$$\text{cov}(\mathbf{z}, \tilde{\mathbf{y}}) = \frac{1}{n-1}\mathbf{z}^T\mathbf{H}\tilde{\mathbf{y}} = \frac{1}{n-1}\dot{\mathbf{z}}^T\tilde{\mathbf{y}} \tag{3.15}$$

with $\tilde{\mathbf{y}}$ a one-dimensional vector of size n of the modified probabilities and $\dot{\mathbf{z}} = \mathbf{H}\mathbf{z}$. Furthermore, the Post-processing constraints aim to minimise the difference between the modified probability $\tilde{Y}$ and the original probability $\hat{Y}$ under this demographic parity

definition. This gives us the following optimisation problem to solve:

$$\begin{aligned} \min_{\tilde{y}_{i1}} \quad & \sum_{i=1}^{n} (\tilde{y}_{i1} - \hat{y}_{i1})^2 \\ \text{s.t.} \quad & \frac{1}{n-1} \sum_{i=1}^{n} \dot{z}_i \tilde{y}_{i1} \leq \epsilon \\ & \frac{1}{n-1} \sum_{i=1}^{n} \dot{z}_i \tilde{y}_{i1} \geq -\epsilon \\ & \tilde{y}_{i1} \leq 1 \qquad \forall i \in \{1, ..., n\} \\ & \tilde{y}_{i1} \geq 0 \qquad \forall i \in \{1, ..., n\} \end{aligned} \tag{3.16}$$

where $\hat{y}_{i1}$ is the i^{th} original probability to be predicted in class 1 and $\tilde{y}_{i1}$ is its equivalent for modified probabilities taking fairness into account.

By modifying the objective function we can define three variants of the optimisation problem [51]. The one we have just presented is the L2-norm.

$$L2 - norm \equiv ||\tilde{\mathbf{y}} - \hat{\mathbf{y}}||_2^2 = \sum_{i=1}^{n} (\tilde{y}_{i1} - \hat{y}_{i1})^2 \tag{3.17}$$

A variation of this is the L1-norm [51]:

$$L1 - norm \equiv ||\tilde{\mathbf{y}} - \hat{\mathbf{y}}||_1 = \sum_{i=1}^{n} |\tilde{y}_{i1} - \hat{y}_{i1}| \tag{3.18}$$

The last objective function is a combination of the other two and is called the L1L2-norm [51].

$$\begin{aligned} L1L2 - norm &\equiv \gamma ||\tilde{\mathbf{y}} - \hat{\mathbf{y}}||_1 + (1-\gamma)||\tilde{\mathbf{y}} - \hat{\mathbf{y}}||_2^2 \\ &= \gamma \sum_{i=1}^{n} |\tilde{y}_{i1} - \hat{y}_{i1}| + (1-\gamma) \sum_{i=1}^{n} (\tilde{y}_{i1} - \hat{y}_{i1})^2 \end{aligned} \tag{3.19}$$

Previous work [25, 40] has shown that the L1-norm has unstable behaviours. For this reason, the L1-norm will not be tested in this work. L1L2-norm will only be tested for $\gamma \in]0, 0.5]$, also to avoid unstable behaviours. Testing other values for γ can sometimes cause Mosek to crash on large datasets.

3.5.2 Post-processing variants

Two different versions of this demographic parity Post-processing method have been proposed by the previous students [12, 25, 40] during their work:

- **First approach**
  A classifier is fitted on both the training and test sets to make predictions, which are then fed into the minimisation problem.

- **Second approach**
  A classifier is only fitted on the training set. The test set is then used to make predictions, which are then fed into the minimisation problem.

It has been shown by the previous students [25, 40] that the two Post-processing approaches produce similar results in terms of accuracy and correlations. We will then only use the second approach as it only decorrelates on the test set and we will evaluate our models on the test set.

An alternative Post-processing method would be to apply the same strategy, but to satisfy equal opportunity or predictive equality. This seems more difficult as we need to provide the true class of instances, which is not available on the test set. This is left for future works.

3.5.3 Progressive decorrelation

For the In-processing method, the decorrelation is applied in several steps by varying the ϵ within the fairness constraint. The initial ϵ value is set to the original covariance between the protected variable and the label one in the base dataset. The final ϵ value is computed by minimising the same covariance with Mosek [3] as explained in section 3.4.4. We then set 8 intermediate steps, each of which progressively increases the decorrelation.

3.6 Generative Adversarial Network based method

The idea here was to investigate a strategy based on Generative Adversarial Network and compare it with all the other methods cited so far. We found a paper by Amirarsalan Rajabi and Ozlem Ozmen Garibay [33] describing their method called *TabFairGan*. It uses a Generative Adversarial Network to achieve fairness exactly as we need. Before we look at how *TabFairGan* works and how we can use it to achieve fairness, let us briefly explain what a Generative Adversarial Network is [11, 20, 45].

3.6.1 Some theory

A Generative Adversarial Network or GAN consists of two neural networks: a Generator and a Discriminator. The two networks play a minimax game. We feed a large amount of data to the GAN. The Generator network has to generate fake samples of the received data to bait the Discriminator. The role of the Discriminator network is to determine how well the fakes are produced by the Generator compared to the real data. To be able to tell if the fake data is well produced, the Discriminator needs to learn the differences between a real and a fake data. It is therefore necessary to train the GAN in two stages. In the first stage, the Discriminator is given a large set of real and fake data to learn the differences between them. In the second stage, the Generator sends fake data to the Discriminator, which sends back information on how to improve it to be more realistic, until the Discriminator can no longer distinguish between fake and real. Both networks are trained, until

they reach an equilibrium where the Generator produces data from the exact target distribution and the Discriminator gives a probability of 0.5 to both real and generated samples.

The GAN formulas will not be investigated further, as there are some twists within the *TabFairGan* architecture. For your personal documentation you can refer to the original paper [20] by Ian Goodfellow et al.

3.6.2 The TabFairGan strategy

One problem that can occur with the above GAN implementation is the non-convergence of the Discriminator. To solve this problem, *TabFairGan* replaces the Discriminator module with a Critic module. This is equivalent to a Wasserstein Generative Adversarial Network or WGAN [4], which is much more stable during learning. The Critic module is equivalent in design to the Discriminator module, but it minimises the Earth-mover's optimal transport distance [37] instead of minimising the Jensen-Shannon divergence [26] to reach the equilibrium state. The network used here is a WGAN with gradient penalty [21]. Here is a closer look at the architectures of *TabFairGAN*.

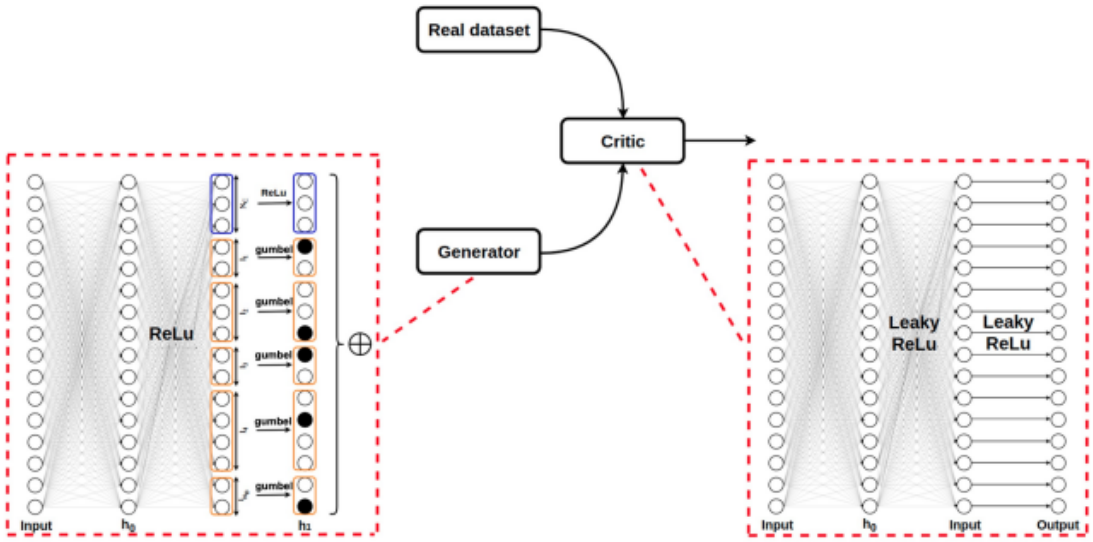


Figure 3.1: *TabFairGan* architecture. Figure taken from [33].

As you can see, the Generator module consists of a first fully-connected layer with ReLu activation functions. Its second layer is composed of Relu activation functions and Gumbel Softmax activation functions. The Relu part is used to generate the numerical features and the Softmax part produces one-hot representations of the categorical features. The final layer concatenates all the features to produce the final data [33]. The Critic module consists of fully connected hidden layers with Leaky ReLu activation functions [33].

TabFairGan training is divided into two main steps. The first one is the accuracy step and the second one is the fairness step. During the accuracy step, the Generator is trained with the Critic network to produce data with a joint probability distribution as close as possible to that of the original dataset. Then, in the fairness step, the same Generator and Critic training continues, but with a fairness constraint added. The purpose of this constraint is to increase demographic parity.

Once the model has been trained, it generates a new dataset. This dataset follows the same distribution as the real dataset, but with fairness introduced. This dataset is then fed to a classifier, which is fitted to evaluate the model. As the classifier training is done on the generated dataset and the model evaluation is done on the real test dataset (10%), it does not make sense to use k-fold cross-validation as for the other models. The whole model training is done five times to get the same amount of data as the other models.

Chapter 4

Methodology and datasets

4.1 Datasets

One of the main problems encountered when working on this thesis was the datasets. Indeed, when talking about fairness in supervised classification, there are few datasets available. It was quite a challenge to find five datasets. Notice that we did not use exactly those chosen by the previous students [25, 40]. In fact, when we looked at what could be found in the literature, the datasets used by the students were very different and looked like already modified versions.

So the selected dataset in this thesis were either chosen from papers in the literature or from known databases. This way we avoid using modified versions of a dataset and we can continue on a more solid basis for future works.
Note that for each dataset, we make the assumption that the correlation between the protected variable and the positive class is negative, otherwise there is no discrimination.

4.1.1 Adult

This dataset was taken from the UC Irvine Machine Learning Repository [34]. Its goal is to predict the income revenues of 48842 people to whether it is above 50.000 USD per year or not ($\geq$ 50.000 USD, $Y = 1$). We modified the « country » column to « continent » and dropped the « fnlwgt » and « education » columns, which were redundant as proposed in [32]. Rows with missing values were also dropped. We ended up with a dataset of 45222 instances for 13 columns and used the gender as the protected attribute (being a female, $Z = 1$) because decisions are biased against women.

Here is some important information about Adult:

- Label proportion:

  $Y = 0$: 83.40%
  $Y = 1$: 16.60%

- Sensitive label proportion:

  $Z = 0$: 67.50%
  $Z = 1$: 32.50%

- Y Z correlation: -0.1685
- Y Z covariance: -0.0294
- Dataset DP: -0.1338

4.1.2 Bank

This dataset was also taken from the UC Irvine Machine Learning Repository [35]. Its purpose is to predict whether or not people will subscribe to a term deposit in a Portuguese bank (will subscribe, $Y = 1$). It is composed of 45211 instances and 17 columns. The protected variable is the age, the clients have been separated into two age categories, the ones between 25 and 60 and the others ($25 \leq age \leq 60$, $Z = 1$).

About Bank:

- Label proportion:

  $Y = 0$: 88.30%
  $Y = 1$: 11.70%

- Sensitive label proportion:

  $Z = 0$: 4.42%
  $Z = 1$: 95.58%

- Y Z correlation: -0.1592
- Y Z covariance: -0.0105
- Dataset DP: -0.2490

4.1.3 Compas

This is probably the most commonly used dataset when talking about fairness and there are several versions of it. We used the one proposed by ProPublica [31]. It is used to predict whether a US criminal will re-offend within two year (not recidivist, $Y = 1$). We chose the ethnicity as the protected attribute because the decisions were biased against the African-Americans (being African-American, $Z = 1$). It originally had 7214 instances, but to make the protected column binary we keep only « African-American » and « Caucasian » ethnicities as proposed in this paper [13]. Then, we kept the same attributes as [32] except for the « age » attribute because of redundancy. The final dataset contains 6150 instances for 11 columns.

About Compas:

- Label proportion:

  $Y = 0$: 46.62%
  $Y = 1$: 53.38%

- Sensitive label proportion:

  $Z = 0$: 39.90%
  $Z = 1$: 60.10%

- Y Z correlation: -0.1185
- Y Z covariance: -0.0289
- Dataset DP: -0.1207

4.1.4 German

The German dataset was used by P. Delobelle et al [13] and was also found on the UC Irvine Machine Learning Repository [36]. Its goal is to predict whether or not a customer is a low or a high credit risk customer (low-risk, $Y = 1$). It is the smallest dataset as it has 1000 instances and 22 columns. The « personal_status_and_sex » column has been split into a « marital_status » column and a « sex » column. A second change was to make the age column binary by setting a cut-off value of 25 to use it as the binary sensitive column (≤ 25, $Z = 1$).

About German:

- Label proportion:

  $Y = 0$: 30.00%
  $Y = 1$: 70.00%

- Sensitive label proportion:

  $Z = 0$: 81.00%
  $Z = 1$: 19.00%

- Y Z correlation: -0.1279
- Y Z covariance: -0.0230
- Dataset DP: -0.1494

4.1.5 Law

This last dataset is from the Law School Admission Council (LSAC) in the US [46]. It comes from a survey made by the LSAC to predict whether or not 27478 law students

will pass the bar exam on the first try (pass, $Y = 1$). The ethnicity was chosen as the protected column (being « non-white », $Z = 1$). Some columns have been removed because they are redundant, as have rows with missing values. This gave us a final dataset with 20798 instances and 12 columns.

About Law:

- Label proportion:

  $Y = 0$: 5.00%
  $Y = 1$: 95.00%

- Sensitive label proportion:

  $Z = 0$: 84.10%
  $Z = 1$: 15.90%

- Y Z correlation: -0.1898
- Y Z covariance: -0.0151
- Dataset DP: -0.1131

4.1.6 Preprocessing of datasets

After the aforementioned modifications, each dataset passes through a preprocessing function. This simply consists of separating X, Y, Z and converting the categorical attributes of X into one-hot vectors. For each categorical column, the first corresponding dummy column is dropped to avoid co-linearity problems. For the numerical attributes, we could have normalised and transformed them using a min-max normalisation as it is often done, but this is not useful in this work. This preprocessing is identical for all datasets, methods and classifiers except for the GAN-based method. In fact, for this GAN-based method, it is essential that the data are correctly represented before being fed as input. Therefore, as in the *TabFairGAN* paper [33], we used one-hot encoding to represent the categorical features and a quantile transformation for the numerical features.

The protected column and the labels column of each dataset are made binary for simplicity. Achieving fairness with multiple protected columns and/or with more than two possible outputs is left for future works.

4.2 Compared classifiers

Results were obtained with five different classifiers. We chose them according to the work of the previous students [12, 25, 40]. The classifiers used are the followings and are available in the SciPy library [44].

- A Random Forest Classifier (RFC).
- A Decision Tree Classifier (DTC).
- A Logistic Regression (LR) with no added penalty.
- A Support Vector Machine (SVM) that must be linear. We used a « LinearSVC » that is implemented in terms of LIBLINEAR [18]. To allow probability prediction we applied it a probability calibration with a logistic regression.
- A Multi Layers Perceptron (MLP) with two hidden layers of 6 neurons each.

No specific classifier tuning was done, but it may be a way to get better fairness results from our models. This is left for future works.

Note that the In-Processing procedure was only run with a Maximum Entropy logistic regression classifier (MaxEnt) and no other procedure was run with this classifier. This classifier has been described in detail in section 3.4.

4.3 Assessment

This section describes the different results the methods produced and how they were validated.

4.3.1 Model validation

As mentioned above, after the preprocessing, the full datasets are then fed into all of our models, whose results are then validated through a 5-fold cross-validation. The only exception to this is the GAN-based model, whose setting makes cross-validation impractical. For all models, the accuracy, correlations and fairness measurements are stored at each fold to allow us to compute the macro-averages of these data. These measures are analysed in chapter 5.

4.3.2 In- and Post-processing validation

As explained in subsection 3.4.4 and subsection 3.5.3, for both In- and Post-processing techniques, the decorrelation is done progressively by varying the thresholds ϵ on covariance within the fairness constraint. For example, here are the epsilons used on ADULT (see Appendix C for the other datasets).

ϵ_0	ϵ_1	ϵ_2	ϵ_3	ϵ_4
0.0147	0.0164	0.0180	0.0196	0.0212
ϵ_5	ϵ_6	ϵ_7	ϵ_8	ϵ_9
0.0229	0.0245	0.0261	0.0277	0.0294

Table 4.1: Thresholds on ADULT for the In- and the Post-processing.

The threshold ϵ_9 corresponds to the original covariance between the protected attribute and the labels. The threshold ϵ_0 is the smallest one and is computed through a covariance minimisation problem (minimum achievable covariance). Please refer to the paper by Vancompernolle [42] for more details.

This will be useful to better analyse the effects of these methods. When displaying the results in chapter 5, the smallest threshold ϵ_0 is shown. This will also be the case when we use statistical tests (see section 4.3.4) in our analysis, the smallest threshold ϵ_0 will be used to compute them. On the other hand, the graphs showing the experimental results throughout the analysis will include all ϵ values.

4.3.3 Performances comparison and quality measures

Here is a description of all the quality measures computed and discussed in chapter 5.

- Accuracy is the prediction accuracy of the classification model.
- Correlation on decisions (Corr. Deci.) is the correlation between the protected variable and the binary prediction (decision) provided by the classifier.
- Correlation on the predicted probabilities (Corr. Proba.) is the correlation between the protected variable and the continuous probability of belonging to the positive class. It is the predicted values provided by the classification model.
- Demographic Parity (DP or Dem. Par.) or statistical parity/fairness implies that a classifier assigns the same probability of being in the positive class to both sensitive and non-sensitive groups. This quantity can be calculated based on binary predictions or probabilistic predictions (see section 2.3.1).

These measures are all computed on the different execution folds and aggregated via a macro-average. The best model should have high accuracy, with correlations and demographic parity as close to zero as possible. These measures will help us throughout our analysis, but they are not sufficient to determine which method or classifier produced the best results in each case. Indeed, one model may beat another in terms of accuracy but be beaten in terms of demographic parity. We therefore need to focus on the trade-off [6] between these two measures, as they are the ones we are the most interested in here. To do this, we will use a modified version of the F1-measure that uses accuracy (Acc) and demographic parity (DP) as follows.

$$F1 = 2 * \frac{Acc * (1 - |DP|)}{Acc + (1 - |DP|)} \tag{4.1}$$

We will also report this modified F1-measure (Modif. F1) alongside with the other quality measures.

4.3.4 Statistical tests

In addition to the above measurement analysis, we carried out some non-parametric tests that make no assumptions about the underlying distribution. Nevertheless, the results of these tests do not always allow us to draw statistical conclusions and are mainly for information purposes, as some of their assumptions are not met. Indeed, they assume that all the data samples are independent, which is not the case. In fact, we concatenate the results of the same method, but for different classifiers and different datasets. We used the same three methods as the previous students [25, 40].

These tests were used to compare methods and classifiers between them in terms of accuracy, demographic parity and the modified F1-measure. As the demographic parity results are mostly negative, we used the absolute value and we want it to be as close to

zero as possible. For accuracy and the modified F1-measure, we want them to be as high as possible.

4.3.4.1 Friedman test

The Friedman test [19] is a non-parametric statistical test and is an alternative to an analysis of variance (ANOVA). It will be used to find variations in the performance of the classifiers across all our simulations, according to the accuracy or correlation obtained when the null hypothesis is rejected.
Let us define R_i the average rank of classifier i out of n classifiers and k strategies. The hypothesis of the Friedman test is then [25, 40, 14]:

$$\begin{aligned} &H_0 : R_i = R_j \forall i, j \in [0, n] \\ &H_1 : \exists i, j \in [0, n] : i \neq j, R_i \neq R_j \end{aligned} \tag{4.2}$$

The test statistic Q is given by [14]:

$$Q = \frac{12n}{k(k+1)} \left[\sum_{j=1}^{k} R_j^2 - \frac{k(k+1)^2}{4} \right] \tag{4.3}$$

Note that if $k \cdot n$ is large enough, Q can be approximated by a chi-squared distribution.

4.3.4.2 Nemenyi test

The Nemenyi test [27] is a post-hoc test used to find divergences between groups of data after a statistical test. It is used after the Friedman test when the null hypothesis is rejected. Two strategies can be categorised as different if their average rank difference is greater than the critical distance CD [25, 40, 14]:

$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6n}} \tag{4.4}$$

where q_α is the Nemenyi critical value for a significance level α, n is the number of classifiers and k is the number of strategies. To calculate q_α the Studentised range statistic for infinite degrees of freedom divided by $\sqrt{2}$ is used [14].

4.3.4.3 Wilcoxon test

The Wilcoxon test [47] is also a non-parametric statistical test and is an alternative to the Student t-test. It allows us to find variations in the performances of the classifiers or strategies across all our simulations. It uses the same test hypothesis as the Friedman test.
For each pairs of classifiers or strategies, it assigns a rank to their performances difference. It then defines R^+ as the sum of the ranks of the positive differences and R^- as the sum

of the ranks of the negative differences. The minimum of these two sums, W_0 allows us to define the test statistic W [25, 40, 14]:

$$W = \frac{W_0 - \frac{1}{4}n(n+1)}{\sqrt{\frac{1}{24}n(n+1)(2n+1)}} \tag{4.5}$$

Note that if n is large enough, W can be approximated by a normal distribution.

4.4 Research questions summary

The next chapter will present the results of our experiments. There is a large amount of results to present, so most of these will be shown in the « Appendices ». To help us with this analysis we will follow the next research questions.

1. The main objective of this work is to analyse and compare the performances of the fairness methods and classification models, thus is there a fairness strategy and/or a classifier that outperforms the others in terms of accuracy, correlation, demographic parity and the trade-off between these values?

2. Can we set up a fairness gradation system between our fairness strategies? This means that we sort our fairness strategies in such a way that they theoretically increase fairness compared to the previous one, and then analyse whether this is verified in the results.

3. Considering the Naïve methods, is there a classifier that performs better than the others in terms of accuracy, correlation, fairness measures and the trade-off between these values? Moreover, how do the Naïve methods perform among them and compared to other methods?

4. Considering Post-processing methods, is it useful to tune the gamma parameters within the L1L2 objective function?

5. How does the In-processing method perform compared to Post-processing methods? Moreover, as the MaxEnt classifier used with the In-processing strategy is a logistic regression classifier, how does it performs compared to the Pre- and Post-processing methods both using a LR classifier ?

6. How does the GAN-based method perform compared to other fairness strategies? In particular, compared to the Pre-processing method, as their objectives are similar. Still considering the GAN-based method, is there a classifier that performs better than the others in terms of accuracy, correlation, fairness measures and the trade-off between these values?

Chapter 5

Experimental comparison

In this chapter the different fairness strategies are compared with different classifiers and different datasets, which leads to many results to show. Therefore, only results on the ADULT dataset are shown in this chapter, but everything else is available in the « Appendices ».

We will go through each methods to answer the research questions from section 4.4. The analysis will be based on observations made on data and graphs. When possible, statistical tests will also be used in order to validate the conclusions. All the measurements are macro-averages, i.e. the results are calculated on the different folds and are then aggregated via a macro-average. Note that all the selected fairness strategies include the sensitive column during the classification model training.

A section will be devoted to each method and will be structured as follows: presentation of the results on the ADULT dataset, analysis on the ADULT dataset, overall analysis on the five datasets and, when possible, verification of the results by statistical tests. The section 5.6 is devoted to the research question number 2 and the last section of this chapter discusses, summarises and reviews the information gathered so far. Lastly, here is a table summarising and defining the abbreviations used throughout this analysis. Please refer to sections 4.2 and 4.3.3 for further explanations.

Abbreviation	**Definition**
DTC	Decision Tree Classifier
LR	Logistic Regression classifier
MLP	Multi Layers Perceptron classifier
RFC	Random Forest Classifier Classifier
SVM	Support Vector Machine classifier
MaxEnt	Maximum Entropy logistic regression classifier
Corr. Deci.	Correlation based on the decisions
Corr. Proba.	Correlation based on the predicted probabilities
Dem. Par. (or DP)	Demographic Parity
Modif. F1.	Modified F1-measure

Table 5.1: List of abbreviations and definitions.

5.1 Baseline results

First of all, the results are presented for the baseline model with no fairness constraint. This method includes the sensitive column during the classification model training. It will serve as a reference method throughout our analysis.

5.1.1 Results overview and analysis

Classifier	Accuracy	Corr. Deci.	Corr. Proba.	Dem. Par.	Modif. F1
DTC	0.8531	-0.1244	-0.3063	-0.0718	0.8891
LR	0.8442	-0.1496	-0.3474	-0.0838	0.8787
MLP	0.8400	-0.1090	-0.3282	-0.0661	0.8845
RFC	0.8463	-0.0968	-0.3733	-0.0383	0.9003
SVM	0.8389	-0.0529	-0.2673	-0.0149	**0.9061**

Table 5.2: Results of the Baseline strategy on ADULT.

Main observations on ADULT. A direct observation is that the demographic parity results have already been reduced compared to the initial demographic parity from the ADULT dataset (see section 4.1.1) which is equal to -0.1338.

All datasets. Lower demographic parity scores than the initial dataset values are also observed for the GERMAN and LAW datasets. For the BANK and COMPAS datasets, the demographic parity values are significantly higher than the initial value of the base dataset. This may be due to the classification model that could have introduce biases. When analysing the Baseline results of all datasets together (see Appendix D), the SVM and RFC classifiers seem to perform relatively well. On the other hand, the LR and DTC classifiers seem to be the worst ones according to the modified F1-measure. Looking at the results of the SVM classifier, it can be seen that it gives the best results on ADULT and BANK, but the accuracies are very close to the initial distribution of the labels in the dataset. The SVM classifier may have predicted the majority of the instances in the same class.

5.2 Pre-processing results

The Pre-processing method has already been studied by the previous students [12, 25], so we will not explore the results of the three variations (see section 3.3.1) of this method. Only the separated projection pre-processing method, which has shown the best performances during the work of the previous students, will be described.

5.2.1 Results overview and analysis

Classifier	Accuracy	Corr. Deci.	Corr. Proba.	Dem. Par.	Modif. F1
DTC	0.8543	-0.1369	-0.3002	-0.0846	0.8838
LR	0.8397	0.0176	-0.0659	0.0084	**0.9094**
MLP	0.8212	-0.1995	-0.2773	-0.1360	0.8420
RFC	0.8450	-0.0926	-0.3912	-0.0351	0.9010
SVM	0.8406	-0.0356	0.0055	-0.0104	0.9090

Table 5.3: Results of the Pre-processing strategy on ADULT.

Main observations on ADULT. On this dataset, the results of the DTC, RFC and SVM classifiers are equivalent to what has been observed with the Baseline method. About the MLP classifier, it does not seem to behave as expected. In fact, it produced significantly worse results than with the Baseline method. However, even though the RFC and SVM results are equivalent to those of the Baseline method, they still produced good results.
At first sight, the LR classifier seems to work relatively well. It correctly reduced the demographic parity without reducing accuracy too much. Given that the LR classifier showed the worse fairness results for the Baseline method, this is an interesting improvement.

All datasets. Two classifiers perform better than the others when considering the four other datasets from the Appendix E. The LR and SVM classifiers consistently produce the best modified F1-measures. The MLP also performs well, but not on every dataset. The RFC classifier sometimes provides good results, but seems to be inconsistent, and the DTC classifier is clearly the worst one here. This probably comes from the fact that they are both non-linear models and this Pre-processing method is a linear decorrelation model. It seems to give better results with classifiers that have linear components as observed above.
Going back to the results of the LR classifier, we can also observe that the demographic parity value has been reversed. It now slightly discriminates the positive attribute, but the value is really low, so this is not an issue.

5.2.2 Statistical tests

Statistical comparisons on the Pre-processing strategy will be performed in section 5.6 together with other comparisons.

5.3 Post-processing results

Confirming the results of [25], the two Post-processing approaches (see section 3.5.2) produce similar results in terms of accuracy and demographic parity. We will therefore

only analyse the second approach. Moreover it makes more sense to use this second approach because it decorrelates on the test set only, as done when evaluating our models. This strategy will sometimes be referred to as *PP_L2* or *PP_L1L2*, depending on the norm used. All results for both norms are shown in the Appendix G.

5.3.1 L2-norm

The results of the post-processing strategy with the L2-norm will be presented and visualised first.

5.3.1.1 Results overview and analysis

Method	Classifier	Accuracy	Corr. Deci.	Corr. Proba.	Dem. Par.	Modif. F1
PP_L2	DTC	0.8524	-0.0918	-0.1651	-0.0526	0.8974
PP_L2	LR	0.8415	-0.0909	-0.1811	-0.0461	0.8942
PP_L2	MLP	0.8391	-0.0705	-0.1772	-0.0363	0.8971
PP_L2	RFC	0.8457	-0.0702	-0.2198	-0.0273	0.9047
PP_L2	SVM	0.8377	-0.0450	-0.2388	-0.0119	**0.9067**

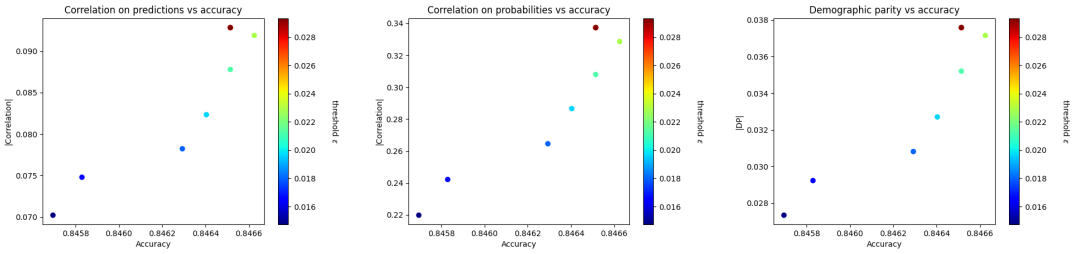
Table 5.4: Results of the Post-processing strategy with L2-norm on ADULT.

Main observations on ADULT. By comparing this method with the Baseline method, better demographic parity scores are observed for each classifier, with a non-significant decrease in accuracy. However, the modified F1-measure scores are very close to those of the Baseline method. Note that better fairness improvements can be observed on the other datasets, the improvements are just smaller here on the ADULT dataset.

All datasets. Looking at all the datasets, it is difficult to say which classifier performs the best. In fact, they all perform well on a different dataset.

5.3.1.2 Visualising the variation of the ϵ threshold

As explained in section 4.3.2, both In- and Post-processing strategies use a progressive decorrelation. This is done by varying the ϵ threshold within the fairness constraint. The three graphs below illustrate the effects of this variation.



(a) (b) (c)

Figure 5.1: Epsilon variation - Post-processing strategy with L2-norm with a Random Forest Classifier on ADULT dataset: comparison between accuracy (in abscissa) and (a) correlation on the decisions (b) correlation on the predicted probabilities (c) demographic parity (in ordinate). The absolute value is applied to the results, so the lower the correlation and demographic parity and the higher the accuracy, the better.

As can be seen, this strategy achieves better correlations and demographic parity results as the threshold ϵ is lowered, without affecting the accuracy too much. This is also the case for the other datasets and classifiers (see Appendix G.1.2), where the drop in accuracy remains small.

5.3.2 L1L2-norm

The results for the post-processing strategy with the L1L2-norm will now be presented and visualised. This will allow us to visualise the effects of varying the gamma parameter. As explained in section 3.5, the L1L2-norm is equivalent to $\gamma(L1-norm)+(1-\gamma)(L2-norm)$. Also, because the L1-norm is unstable when $\gamma > 0.5$, we will only variate the γ parameter between $[0, 0.5]$. Note that when we display the results of the L1L2-norm, we use a γ parameter set to 0.5 exactly.

5.3.2.1 Results overview and analysis

Method	Classifier	Accuracy	Corr. Deci.	Corr. Proba.	Dem. Par.	Modif. F1
PP_L1L2	DTC	0.8524	-0.0998	-0.1630	-0.0590	0.8945
PP_L1L2	LR	0.8413	-0.1049	-0.1805	-0.0572	0.8892
PP_L1L2	MLP	0.8406	-0.0842	-0.1761	-0.0466	0.8935
PP_L1L2	RFC	0.8472	-0.0751	-0.2192	-0.0312	0.9039
PP_L1L2	SVM	0.8376	-0.0449	-0.2386	-0.0120	**0.9066**

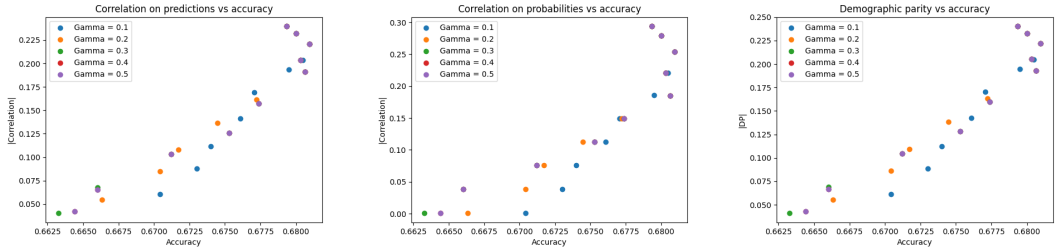
Table 5.5: Results of the Post-processing strategy on ADULT. L1L2-norm is used with the gamma parameter set to 0.5.

All datasets. At first sight, there is no significant difference with the L2-norm, for any dataset. At the end of this section we will use statistical tests to see if we can find any significant difference between the fairness strategies.

5.3.2.2 Visualising the variation of the γ parameter

For the L1L2-norm, we want to visualise the effect of varying of the ϵ threshold but this time for different γ parameter values. The graphs for each dataset are available in the Appendix G.2.2.

For the ADULT dataset, all γ values give identical results (see Appendix G.2.2). Varying the γ parameter within the L1L2 formula does not produce visually interesting results. This is also observed on the other datasets except for COMPAS as shown below.



(a) (b) (c)

Figure 5.2: Gamma variation - Post-processing strategy with L1L2 with a Random Forest Classifier on COMPAS dataset: comparison between accuracy (in abscissa) and (a) correlation on the decisions (b) correlation on the predicted probabilities (c) demographic parity (in ordinate). The absolute value is applied to the results, so the lower the correlation and demographic parity and the higher the accuracy, the better.

Looking at the results of the the five classifiers on COMPAS (see Appendix G.2.2), it is difficult to draw conclusions about which γ value gives the best fairness results. None of them stand out.

5.3.3 Statistical tests

Given the observed similarities between the L2- and L1L2-norms of the Post-processing strategy, some statistical tests were performed to compare the two norms. We conducted a Friedman test to compare the L2-norm with the L1L2-norm using five values for the gamma parameter ranging from 0.1 to 0.5. Each method is represented by 25 values (5 datasets times 5 classifiers). At a significance level of $5.00e^{-2}$, the Friedman test concludes that none of the methods is significantly different from the others in terms of accuracy (p-value = 0.6082), demographic parity (p-value = 0.4159) and the modified F1-measure (p-value = 0.9470).

Conclusion. This allows us to answer to the research question number 4. For each of the measurements, no significant difference was observed between the norms for any of the values of the gamma parameter. Following what has been said, and in order to

avoid redundancy of results, the L1L2-norm will be arbitrarily chosen to represent the Post-processing method in the rest of this work.

5.4 In-processing results

The results of the In-processing strategy will now be analysed to see how it compares with the others.

5.4.1 Results overview and analysis

As explained in section 4.3.2, the In-processing model achieves decorrelation step by step by varying the ϵ parameter. The results for the smallest ϵ are shown below. The In-processing strategy is only executed with a MaxEnt classifier, so the results on the five datasets are shown.

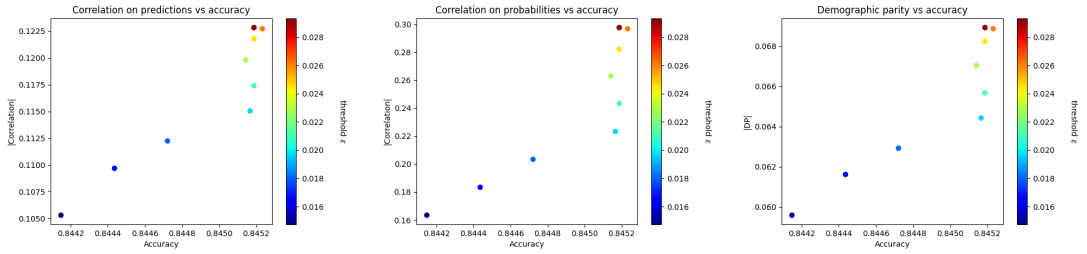
Dataset	Accuracy	Corr. Deci.	Corr. Proba.	Dem. Par.	Modif. F1
ADULT	0.8441	-0.1053	-0.1635	-0.0596	**0.8897**
BANK	0.8984	-0.0192	-0.0233	-0.0234	**0.9359**
COMPAS	0.6636	-0.0298	-0.0001	-0.0300	**0.7880**
GERMAN	0.7470	0.0077	0.0069	0.0085	**0.8521**
LAW	0.9465	-0.0203	-0.0172	-0.0071	**0.9692**

Table 5.6: Results of the In-processing strategy on all datasets. The In-processing strategy uses a Binary Maximum Entropy classifier.

All datasets. This strategy gives very low demographic parity scores. It seems to correctly improve fairness and decorrelation from the protected variable. It even sometimes achieves slightly better accuracy than some classifiers used with the Baseline method. These results are similar to those observed with Post-processing methods. It would therefore be interesting to compare them. Both fairness strategies will be analysed together by statistical tests at the end of this section.

5.4.2 Visualising the variation of the ϵ threshold

Here is a visualization of how the decorrelation increases fairness step by step on the ADULT dataset.



(a) (b) (c)

Figure 5.3: Epsilon variation - In-processing strategy on ADULT dataset: comparison between accuracy (in abscissa) and (a) correlation on the decisions (b) correlation on the predicted probabilities (c) demographic parity (in ordinate). The absolute value is applied to the results, so the lower the correlation and demographic parity and the higher the accuracy, the better.

The graphs allow us to visualise how the In-processing method achieves better correlations and demographic parity results as the threshold ϵ is lowered with a very small drop in accuracy. This is again similar to what has been observed with the Post-processing methods.

5.4.3 Statistical tests

Comparisons between the In- and Post-processing strategies. Considering the observed similarities between the In- and Post-processing strategies, several statistical tests were carried out to compare the two methods. The L1L2-norm of Post-processing will be used for this comparison. Since the In-processing was executed with only one classifier, we have separated the Post-processing method by classifier. This leads to only 5 values (5 classifiers) per method, which is not enough to draw conclusions from our statistical tests. However, all results were obtained by a 5-fold validation as explained in section 4.3.1. The results on each fold will be considered as a measurement, so that each comparison is performed on 25 paired values (5 datasets times 5 folds) [2].
At a significance level of $5.00e^{-2}$, the Friedman test concludes that none of the methods is significantly different from the others in terms of accuracy (p-value = 0.3115), demographic parity (p-value = 0.0843) and the modified F1-measure (p-value = 0.4021).

Conclusion. We can then answer the research question number 5, the In- and Post-processing strategies are not statistically different in terms of accuracy, demographic parity and the modified F1-measure.

Comparisons between logistic regressions. As the MaxEnt classifier used with the In-processing method is a logistic regression classifier, it would be interesting to know how it performs compared to the Pre- and Post-processing (L1L2-norm) methods both using

a LR classifier. Three Wilcoxon statistical tests were then performed to determine if any difference could be observed for the accuracy, the demographic parity and the modified F1-measure. As for the previous test, the results of each fold are used to represent the methods within the statistical tests [2]. So each method is also represented by 25 values (5 datasets times 5 folds).
At a significance level of $5.00e^{-2}$, the Wilcoxon tests (see Appendix A.4) conclude that none of the fairness strategies is significantly different from the others in terms of accuracy, demographic parity and the modified F1-measure.

Conclusion. We can then complete the answer to the research question number 5, the three logistic regressions are not statistically different in terms of accuracy, demographic parity and the modified F1-measure.

5.5 Naïve Methods

As a reminder, the Naïve method consists of a baseline training phase that includes the protected attribute followed by a modified testing phase. During testing the protected test column is replaced by 0.5 (Naive05), the average (NaiveAvg) or the a priori distribution (NaivePrior) of the protected column.

5.5.1 Results overview and analysis

Method	Classifier	Accuracy	Corr. Deci.	Corr. Proba.	Dem. Par.	Modif. F1
NaiveAvg	DTC	0.8531	-0.1240	-0.3008	-0.0717	0.8891
NaiveAvg	LR	0.8405	-0.0397	-0.2203	-0.0182	0.9056
NaiveAvg	MLP	0.8366	0.0053	-0.1832	0.0023	**0.9101**
NaiveAvg	RFC	0.8467	-0.0927	-0.3265	-0.0372	0.9010
NaiveAvg	SVM	0.8405	-0.0467	-0.1944	-0.0149	0.9071

Table 5.7: Results of the NaiveAvg strategy on ADULT.

Method	Classifier	Accuracy	Corr. Deci.	Corr. Proba.	Dem. Par.	Modif. F1
Naive05	DTC	0.8531	-0.1240	-0.3008	-0.0717	0.8891
Naive05	LR	0.8398	-0.0315	-0.2156	-0.0128	0.9076
Naive05	MLP	0.8373	0.0172	-0.1756	0.0066	**0.9087**
Naive05	RFC	0.8467	-0.0927	-0.3265	-0.0372	0.9010
Naive05	SVM	0.8402	-0.0460	-0.1923	-0.0142	0.9072

Table 5.8: Results of the Naive05 strategy on ADULT.

Method	Classifier	Accuracy	Corr. Deci.	Corr. Proba.	Dem. Par.	Modif. F1
NaivePrior	DTC	0.8531	-0.1240	-0.2981	-0.0717	0.8891
NaivePrior	LR	0.8394	0.0094	-0.1513	0.0036	**0.9112**
NaivePrior	MLP	0.8362	0.0552	-0.1204	0.0202	0.9023
NaivePrior	RFC	0.8430	-0.0632	-0.2750	-0.0215	0.9057
NaivePrior	SVM	0.8398	-0.0386	-0.1578	-0.0117	0.9080

Table 5.9: Results of the NaivePrior strategy on ADULT.

Main observations on ADULT. The first direct observation is that the three proposed techniques produced almost identical results. However, although the results are very similar, some minor differences can be observed. We will analyse them further with a statistical test at the end of this section. Here on ADULT the LR and NaivePrior combination and the MLP and NaiveAvg combination achieved the higher observed modified F1-measures so far.

All datasets. The overall fairness improvement is highly dependent on the dataset and the classifier used. On BANK, the SVM classifier performed well and on LAW weit is the RFC classifier that produced the best results.
For GERMAN, we observed a significant drop in accuracy when using an MLP classifier. In fact, an accuracy of 0.456 is observed for all the Naïve strategies, while the other classifiers reach a score of at least 0.7. Moreover, on GERMAN, the results are even more similar from one strategy to another which is probably due to the size of the dataset.
On COMPAS no classifier managed to improve the demographic parity. This may be due to some « non-sensitive » columns that are correlated with the protected variable. They can then act as a proxy for the protected column. In the next subsection, two updated Naïve methods were tested to overcome this problem and improve fairness.

Conclusion. This allows us to answer to the first part of the research question number 3. There does not seem to be one classifier that is globally better than the others for the Naïve methods.

5.5.2 Extending the Naïve methods to more columns

To improve the Naïve methods, we want to apply the average Naïve method to the two or five columns with the higher covariance with the protected column. Note that the transformation is also applied to the protected attribute in addition to the 2 and 5 columns. The methods are referred to as Naive2 and Naive5 respectively.

Method	Classifier	Accuracy	Corr. Deci.	Corr. Proba.	Dem. Par.	Modif. F1
Naive2	DTC	0.8401	-0.1026	-0.3206	-0.0437	0.8945
Naive2	LR	0.8366	-0.0170	-0.2194	-0.0071	0.9081
Naive2	MLP	0.8340	NaN	0.0008	NaN	NaN
Naive2	RFC	0.8355	-0.0300	-0.3373	-0.0044	**0.9085**
Naive2	SVM	0.8341	0.0401	-0.1737	0.0027	0.9084

Table 5.10: Results of the Naive2 strategy on ADULT.

Method	Classifier	Accuracy	Corr. Deci.	Corr. Proba.	Dem. Par.	Modif. F1
Naive5	DTC	0.8337	0.0025	0.0426	0.0001	**0.9093**
Naive5	LR	0.8340	0.0168	-0.1335	0.0038	0.9079
Naive5	MLP	0.8340	NaN	0.0008	NaN	NaN
Naive5	RFC	0.8340	NaN	-0.1899	NaN	NaN
Naive5	SVM	0.8339	0.0179	0.0227	0.0005	0.9092

Table 5.11: Results of the Naive5 strategy on ADULT.

Main observations on ADULT. As can be seen, some values are missing. This means that the corresponding classifier and strategy combinations predicted all instances in the same class. However, when a result is produced, the modified F1-measure scores are similar to what was observed above, and sometimes even higher. It also appears that the demographic parity scores are lower, especially for the Naive5 method. We will use statistical tests at the end of this section to see if a significant difference can be defined between the five methods.

All datasets. When a result is produced, the Naive2 and Naive5 methods seem to produce worse accuracy but better demographic parity scores. An improvement in terms of demographic parity is observed on the COMPAS dataset, although nothing happened with the three initial Naïve strategies. However, on the LAW dataset, the Naive5 method produced no results for any classifier.

5.5.3 Performances visualisation of the five Naïve strategies

The graphs below show the performances of the five Naïve methods on the ADULT dataset with our five classifiers. This allows us to observe the trade-off between accuracy and demographic parity. The same graphs for the other datasets are available in the Appendix H.2.

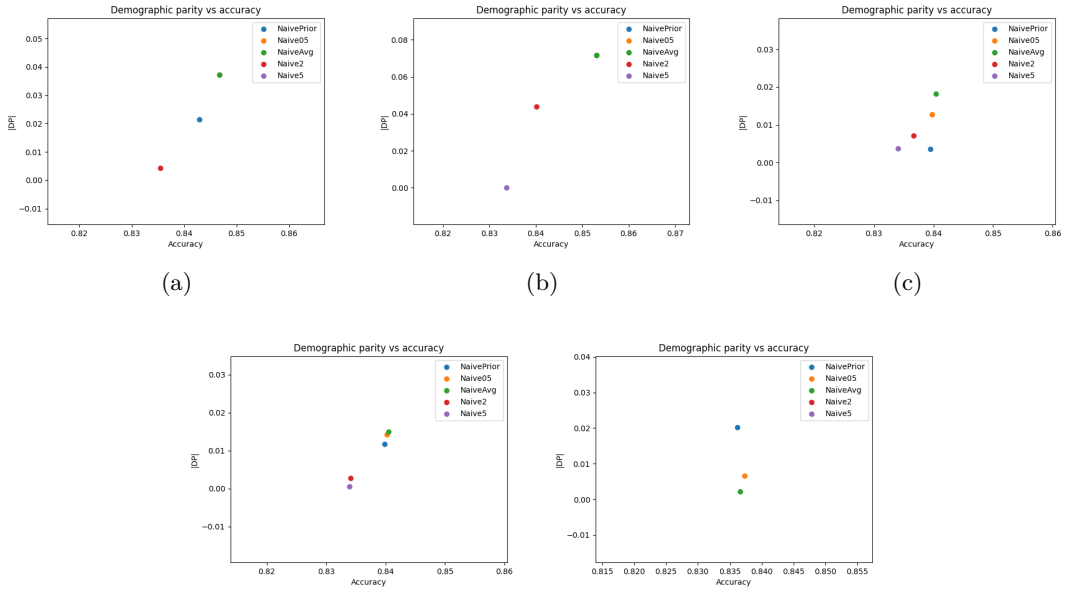


Figure 5.4: Comparison between accuracy (in abscissa) and the absolute value of demographic parity (in ordinate) on the ADULT dataset with a (a) Random Forest Classifier (b) Decision Tree Classifier (c) Logistic Regression (d) Support Machine Vector (e) Multi Layer Perceptron. The lower the demographic parity and the higher the accuracy, the better.

This time we can see more clearly that the Naive2 and Naive5 methods seem to produce worse accuracy but better demographic parity values than the NaiveAvg, Naive05 and NaivePrior strategies. This is also globally true for the other datasets.

5.5.4 Statistical tests: Naïve methods comparison

Friedman statistical tests. Let us use some statistical tests to see if, despite all that has been explained, one of the five methods is overall better than the others. To begin with, a Friedman test was conducted to compare the Naïve strategies. The Baseline method has been added to the comparison as a reference. Each method is represented by 25 values (5 datasets times 5 classifiers). At a significance level of $5.00e^{-2}$, the Friedman test concludes that at least one strategy is significantly different from the others in terms of accuracy (p-value = $1.306e^{-7}$). However, it also concludes that none of the methods is significantly different from the others in terms of demographic parity (p-value = 0.4472) and the modified F1-measure (p-value = 0.3806).

Nemenyi statistical tests. We can therefore conduct three post-hoc Nemenyi statistical tests, but only the one for the accuracy is meaningful. Below are the three

corresponding graphs for accuracy, demographic parity and the modified F1-measure. Since the Naive2 and Naive5 methods sometimes fail to produce results, the following observations are therefore only valid when these two fairness strategies produce results.

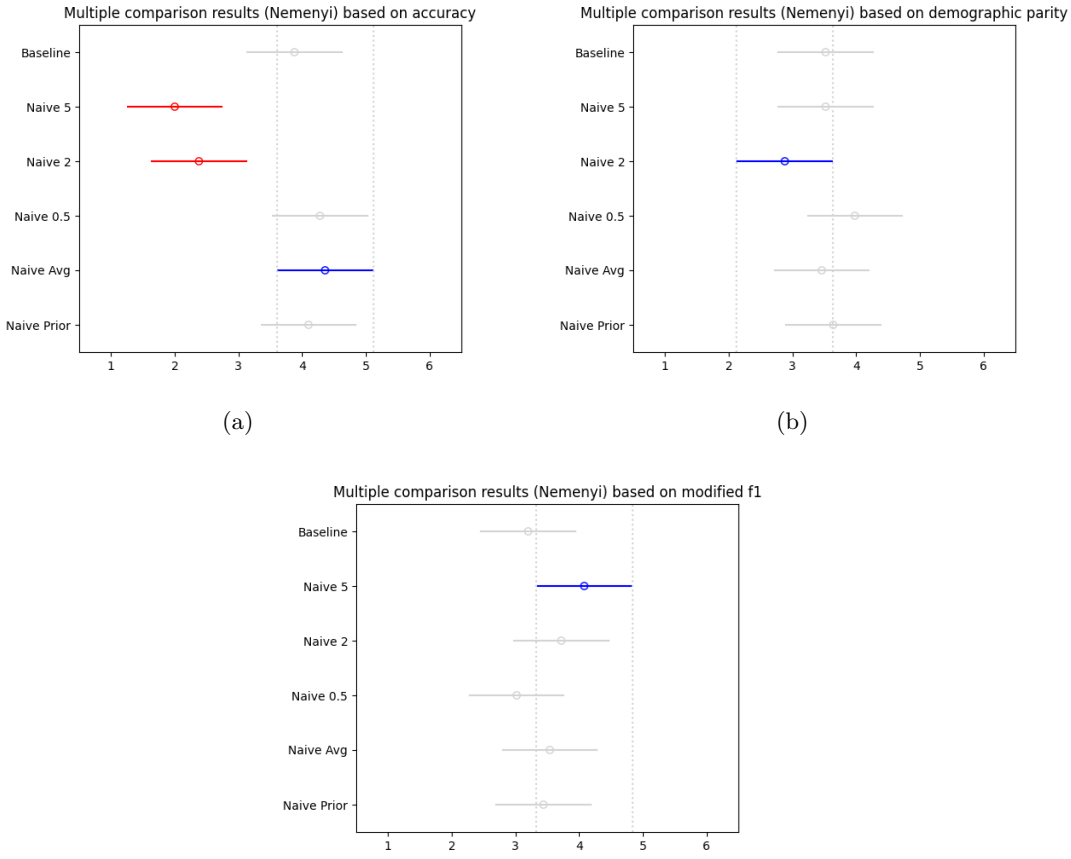


Figure 5.5: Nemenyi statistical test to compare the Naive strategies between themselves according to (a) accuracy (b) the absolute value of demographic parity (c) the modified F1-measure. The blue mark is the best performing strategy i.e. the lower rank for demographic parity or the higher rank for accuracy and the modified F1-measure. The other fairness strategies are represented as a red mark if their average rank difference with the blue mark is greater than the critical distance.

Despite the fact that the Naive2 and Naive5 methods are significantly worse in terms of accuracy, there is no significant difference in terms of demographic parity and the modified F1-measures.

Wilcoxon statistical tests. A Wilcoxon statistical test (see Appendix A.2) was run to analyse this further, as Wilcoxon tests are more powerful than Nemenyi tests to detect differences between pairs of methods. Here is what we can deduce. About the first three Naïve methods, they are similar in terms of accuracy and demographic parity, but the NaiveAvg method is significantly better than the Naive05 method. For the two extended Naïve methods, the Wilcoxon test confirms the previous observations, the Naive2 and Naive5 methods are significantly worse than the other strategies in terms of accuracy and both are significantly better in terms of demographic parity. However, the Naive5 strategy is significantly better than all the other Naïve and Baseline methods. Finally, about the Baseline method, it is equivalent to the first three Naïve strategies. Both extended Naïve strategies are better than the Baseline method in terms of demographic parity, but only the Naive5 strategy is significantly better in terms of the modified F1-measure.

Conclusion. This analysis allows us to partially answer to the research question number 3. It appears that the NaiveAvg, Naive05 and NaivePrior strategies provide equivalent results, but NaiveAvg has a slightly higher mean rank. If we take into account the five Naive strategies, it is the Naive5 strategy that stands out. However, this method does not give a solution for every dataset with every classifier, we will then use the NaiveAvg method as our reference Naïve strategy for the rest of this work.

5.6 Improving fairness step by step

In this section, we will try to gradually reduce the effect of the protected column on the decisions in order to increase fairness step by step. To do this, we will go through five types of methods that are expected to gradually improve fairness. The methods are a Baseline strategy, an « Omit Z » strategy, a Naïve strategy, a Pre-processing strategy and a Post-processing strategy. The results of the five methods are shown below.

Stage 1. The Baseline strategy.
This method will be used as a basis for the analysis of the other four stages.

Classifier	Accuracy	Corr. Deci.	Corr. Proba.	Dem. Par.	Modif. F1
DTC	0.8531	-0.1244	-0.3063	-0.0718	0.8891
LR	0.8442	-0.1496	-0.3474	-0.0838	0.8787
MLP	0.8400	-0.1090	-0.3282	-0.0661	0.8845
RFC	0.8463	-0.0968	-0.3733	-0.0383	0.9003
SVM	0.8389	-0.0529	-0.2673	-0.0149	**0.9061**

Table 5.12: Results of the Baseline strategy on ADULT. The protected column is included during the training phase.

Stage 2. The Omit Z strategy.
This method is simply a baseline strategy where the sensitive column is not included

during the training. The results of this method have not been analysed in the previous sections of the analysis, but they are available for all datasets in the Appendix D.

Classifier	Accuracy	Corr. Deci.	Corr. Proba.	Dem. Par.	Modif. F1
DTC	0.8531	-0.1244	-0.3027	-0.0718	0.8891
LR	0.8448	-0.1328	-0.2955	-0.0733	0.8839
MLP	0.8469	-0.1253	-0.2814	-0.0755	0.8840
RFC	0.8465	-0.0924	-0.3354	-0.0371	0.9010
SVM	0.8376	-0.0371	-0.2260	-0.0087	**0.9080**

Table 5.13: Results of the Baseline strategy on ADULT. The protected column is not included during the training phase.

Stage 3. The Naïve average strategy.
This method was selected from the first three Naïve strategies according to the conclusion of the subsection 5.5.4.

Classifier	Accuracy	Corr. Deci.	Corr. Proba.	Dem. Par.	Modif. F1
DTC	0.8531	-0.1240	-0.3008	-0.0717	0.8891
LR	0.8405	-0.0397	-0.2203	-0.0182	0.9056
MLP	0.8366	0.0053	-0.1832	0.0023	**0.9101**
RFC	0.8467	-0.0927	-0.3265	-0.0372	0.9010
SVM	0.8405	-0.0467	-0.1944	-0.0149	0.9071

Table 5.14: Results of the NaiveAvg strategy on ADULT.

Stage 4. The Pre-processing strategy.

Classifier	Accuracy	Corr. Deci.	Corr. Proba.	Dem. Par.	Modif. F1
DTC	0.8543	-0.1369	-0.3002	-0.0846	0.8838
LR	0.8397	0.0176	-0.0659	0.0084	**0.9094**
MLP	0.8212	-0.1995	-0.2773	-0.1360	0.8420
RFC	0.8450	-0.0926	-0.3912	-0.0351	0.9010
SVM	0.8406	-0.0356	0.0055	-0.0104	0.9090

Table 5.15: Results of the Pre-processing strategy on ADULT.

Stage 5. The Post-processing strategy.
The L1L2-norm was selected to represent the Post-processing strategy according to the conclusion of the section 5.3.3.

Classifier	Accuracy	Corr. Deci.	Corr. Proba.	Dem. Par.	Modif. F1
DTC	0.8524	-0.0998	-0.1630	-0.0590	0.8945
LR	0.8413	-0.1049	-0.1805	-0.0572	0.8892
MLP	0.8406	-0.0842	-0.1761	-0.0466	0.8935
RFC	0.8472	-0.0751	-0.2192	-0.0312	0.9039
SVM	0.8376	-0.0449	-0.2386	-0.0120	**0.9066**

Table 5.16: Results of the Post-processing strategy with L1L2-norm ($\gamma = 0.5$) on ADULT.

All datasets. As observed, the results are very close from one stage to the next, even when looking at all the datasets. It is difficult to tell whether the next stage has improved fairness just by looking at the tables. Therefore, in the next section we will use statistical tests to provide a better analysis.

5.6.1 Statistical tests

Friedman statistical tests. Let us start our analysis with a Friedman test. Each stage is represented by 25 values (5 datasets times 5 classifiers). At a significance level of $5.00e^{-2}$, the Friedman test concludes that at least one stage is significantly different from the others in terms of demographic parity (p-value = 0.0029). However, it also concludes that none of the methods is significantly different from the others in terms of accuracy (p-value = 0.3762) and the modified F1-measure (p-value = 0.1123).

Nemenyi statistical tests. We can therefore conduct three post-hoc Nemenyi statistical tests, but again, only the one for demographic parity is meaningful. Here are the three corresponding graphs for accuracy, demographic parity and the modified F1-measure.

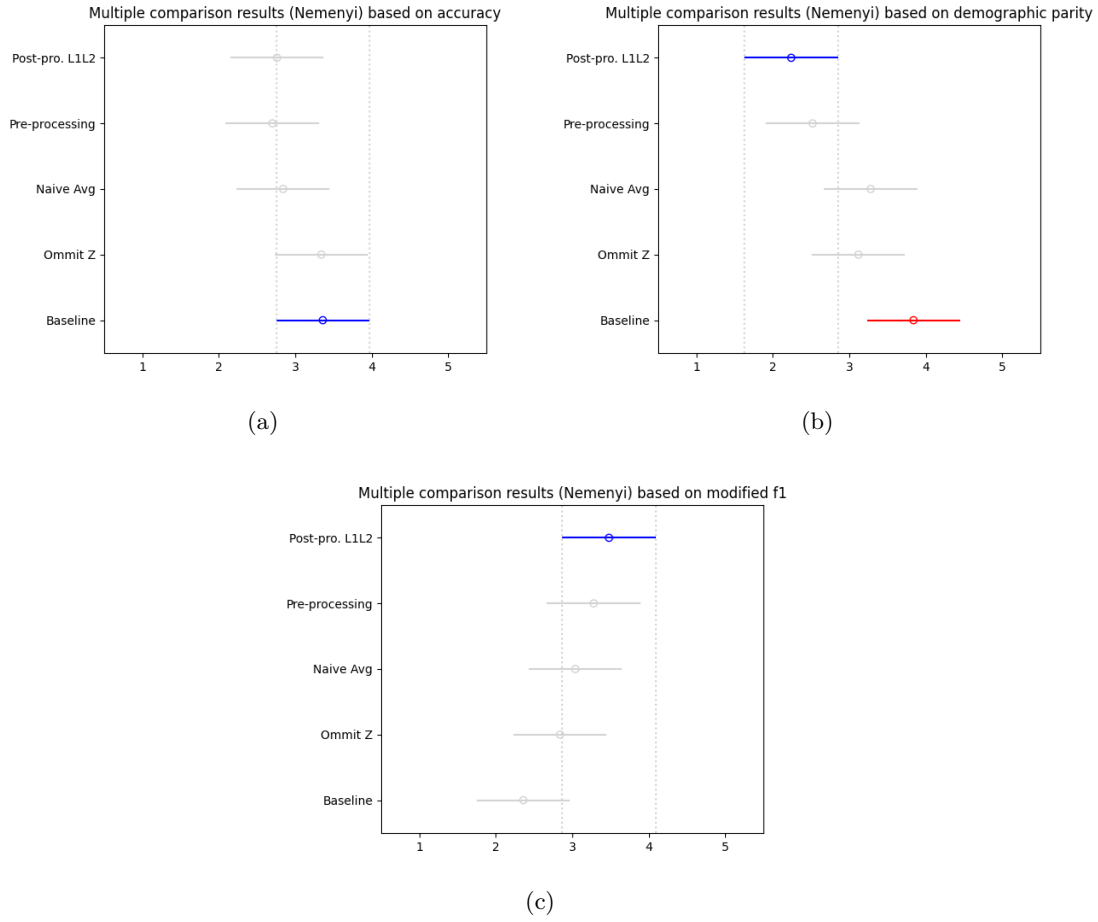


Figure 5.6: Nemenyi statistical test to compare the five stages of gradation according to (a) accuracy (b) the absolute value of demographic parity (c) the modified F1-measure. The blue mark is the best performing stage i.e. the lower rank for demographic parity or the higher rank for accuracy and the modified F1-measure. The other stages are represented as a red mark if their average rank difference with the blue mark is greater than the critical distance.

The order of the theoretical decorrelation of the sensitive attribute is observed in practice in terms of the modified F1-measure, even if there is no significant difference between any of the stages. This is also observed approximately for the demographic parity, where the only order difference is between stages 2 and 3, which swap places. The Baseline method has the best accuracy but the worst demographic parity, as we could expect. It produced a significantly worse demographic parity value than the Pre- and Post-processing strategies. The second and third stages are not significantly different from the Baseline case on any criterion. The fourth and fifth stages are also very similar to each other.

Wilcoxon statistical tests. Multiple Wilcoxon statistical tests (see Appendix A.5) were performed to compare stages two by two. This allows us to make more precise observations as a Wilcoxon test is more powerful in detecting differences between strategies. First, we compared stage 1 to stage 2. This shows that the Omit Z strategy is significantly better than the Baseline method in terms of demographic parity and the modified F1-measure, but there was no difference in accuracy. Then, no difference was found for any of these three measures when comparing stage 2 with stage 3, stage 3 with stage 4 and stage 4 with stage 5.
However, some significant differences were observed when comparing non-consecutive stages. The Pre-processing strategy is significantly better than the Baseline method for all measurements. It is also significantly better than the Omit Z strategy in terms of demographic parity. The Post-processing method (stage 5) produces significantly better demographic parity results than the first three stages.

Conclusion. This allows us to answer to the research question number 2. We observe a fairness grading system where one strategy increases the fairness score compared to the previous one. The two fairness strategies with the higher ranks are the Pre- and Post-processing strategies.

5.7 GAN-based method: TabFairGAN

The GAN-based strategy has a preprocessing and a training different from our other fairness strategies (see sections 3.6.2 and 4.1.6). A comparison with our Baseline method does not make much sense. We will therefore use a new baseline method with the same preprocessing and training as the GAN strategy. It will be referred to as « BaselineGAN ». Our GAN-based strategy training is divided into two phases. The first one optimises the accuracy and the second one optimises the demographic parity. The BaselineGAN method will only perform the first phase. This will ensure an equivalent level of accuracy between the two strategies and will allow us to better analyse the effect of the second phase of training.

The number of steps in each training phase depends on the size of the dataset. A small dataset such as GERMAN will require more training steps to produce results, but the overall execution time will be faster than for a larger dataset such as ADULT. This number of steps can be increased to get different and probably better results, but this will also increase the execution time. The number of steps for each training phase is given in the caption of each table. These numbers have been defined to produce an accuracy score as close as possible to the accuracy score produce by our Baseline model. However, it is not possible to produce the same accuracy score for some datasets without taking too much computation time.

For both GAN-based strategies, we have run the models five times and shown the average of the five runs. This is done because the strategy is not yet stabilised and

optimised. The fairness results can vary greatly from run to run, sometimes giving excellent and sometimes mediocre results. Moreover, on the smallest datasets, some of the five executions are not able to produce a result. This leads to some missing values and can affect the results. In this section we will see if this GAN-based method has the potential to produce good fairness results on average, but in the future it will be important to stabilise and optimise it to get the best performances out of it.
Also, for a given model and dataset, the five runs may sometimes produce positive and negative fairness results. Taking only the average of these five runs will affect the demographic parity value and the calculation of the modified F1-measure. The average of the absolute value of the demographic parity is therefore be shown for our GAN-based strategies.

5.7.1 Results overview and analysis: BaselineGAN

Here are the average and standard deviation of the results of five executions for the BaselineGAN strategy on the ADULT dataset.

Classifier	Accuracy	Corr. Deci.	Corr. Proba.	|DP|	Modif. F1
DTC	0.8190 ± 0.0148	-0.2058 ± 0.0141	-0.3178 ± 0.0358	0.1716 ± 0.0236	0.8236 ± 0.0190
LR	0.8293 ± 0.0085	-0.2080 ± 0.0280	-0.3252 ± 0.025	0.1735 ± 0.0306	0.8279 ± 0.0190
MLP	0.8209 ± 0.0080	-0.2069 ± 0.0172	-0.3192 ± 0.0114	0.1782 ± 0.0200	0.8214 ± 0.0137
RFC	0.8384 ± 0.0077	-0.2012 ± 0.0273	-0.3874 ± 0.0153	0.1517 ± 0.0291	**0.8433** ± 0.0177
SVM	0.8307 ± 0.0085	-0.2044 ± 0.0286	-0.3142 ± 0.0240	0.1697 ± 0.0310	0.8305 ± 0.0191

Table 5.17: Average and standard deviation for the results of five runs of the GAN-based strategy on ADULT. This strategy does not include fairness. Accuracy training steps: 350.

Main observations on ADULT. The accuracy obtained is close, but lower than with the original Baseline method. We probably could have achieved a higher accuracy, but it would have taken too much computing time. This drop in accuracy is less significant for smaller datasets.

All datasets. For the GERMAN and LAW datasets (see Appendix I), some executions predicted all instances in the same class and produced NaN results. This was even observed for the five executions on LAW with an RFC classifier, which produced no results. We tried to increase the number of training steps, but with no success. This is then probably due to the optimisation and tuning of this GAN-based strategy.
Nevertheless, relatively good fairness results were observed on the LAW dataset. This should not be the case as this BaselineGAN method does not include fairness in its training. This seems to be the same behaviour as already observed in this chapter with the SVM classifier (see section 5.1).

5.7.2 Results overview and analysis: GAN

We will now present the average and standard deviation of the results of the GAN strategy, which includes both training phases on ADULT.

Classifier	Accuracy	Corr. Deci.	Corr. Proba.	|DP|	Modif. F1
DTC	0.8285 ± 0.0119	-0.1822 ± 0.0209	-0.2906 ± 0.0065	0.1395 ± 0.0289	0.8442 ± 0.0200
LR	0.8358 ± 0.0049	-0.1562 ± 0.0103	-0.2329 ± 0.0159	0.1228 ± 0.0076	0.8560 ± 0.0040
MLP	0.8275 ± 0.0050	-0.1698 ± 0.0133	-0.2393 ± 0.0194	0.1398 ± 0.0108	0.8435 ± 0.0034
RFC	0.8464 ± 0.0032	-0.1524 ± 0.0116	-0.3312 ± 0.0075	0.1017 ± 0.0101	**0.8716** ± 0.0039
SVM	0.8378 ± 0.0054	-0.1482 ± 0.0096	-0.2254 ± 0.0154	0.1155 ± 0.0065	0.8605 ± 0.0039

Table 5.18: Average and standard deviation for the results of five runs of the GAN-based strategy on ADULT. This strategy include fairness. Accuracy training steps: 350. Fairness training steps: 200.

All datasets. On all the datasets in Appendix I, the second phase of training managed to keep the accuracy almost as high as it was at the end of the first phase. The accuracy is not affected while the demographic parity is improved. However, this improvement is disappointing when compared to other fairness strategies, and even more when considering the training time.
Looking at the classifiers, the RFC classifier is the one that performs the best on all datasets except on COMPAS. About the COMPAS dataset, we observe that no classifier was able to improve the demographic parity. The same behaviour was observed with the Naïve strategies (see section 5.5).
The standard deviations give interesting information about the potential behind this fairness strategy. They are sometimes very high, especially for demographic parity and the modified F1-measure. This strategy could give good fairness results, particularly with an RFC classifier, but it will need to be stabilised to give consistently good fairness results.

Conclusion. We can give a first answer to the research question 6, it seems that the RFC classifier is relatively well performing with the GAN-based strategy.

5.7.3 Statistical tests

Comparisons between GAN and BaselineGAN strategies. We perform several statistical tests to compare this GAN-based strategy with other methods. First, three Wilcoxon tests were used to compare the GAN strategy with the BaselineGAN strategy (see Appendix A.6). There was no difference in accuracy between the two methods, but the tests showed that the GAN strategy was significantly better for both the demographic parity and the modified F1-measure, as we had previously observed.

Comparisons between GAN and Pre-processing strategies. Three other Wilcoxon tests were then performed to compare the GAN-based method with the Pre-processing

method, as they both aim to remove the linear correlation between the label variable and the protected variable. Indeed, this will show how the GAN-based strategy, which is more computationally intensive, can come close to the results of the Pre-processing method. As the preprocessing of the two strategies is different, we adapted the one used for the Pre-processing strategy to make this comparison. We split the whole dataset and used 90% for training and 10% for testing as for the GAN-based method.
The tests showed that the Pre-processing strategy was significantly better in terms of accuracy, demographic parity and the modified F1-measure. This result was expected given the poor performances of the GAN-based method. Of course, those tests will have to be confirmed once the GAN-based method has been stabilised and optimised to produce better results.

Conclusion. This allows us to partially answer to the research question 6. If we take the GAN-based strategy as it is, using it instead of a Pre-processing strategy will never be worthwhile. However, the GAN-based strategy has the potential to produce good fairness results in some cases, but it will need to be stabilised and optimised to produce consistently good results. This is left for future works.

5.8 Overall discussion

This section will serve as a summary of the previous sections. We will compare all the fairness strategies analysed so far. It will also allow us to answer the remaining research questions from the section 4.4.

5.8.1 Statistical tests on all fairness strategies

We will first analyse the performances of all previously studied fairness strategies. The comparisons can be conducted using the Friedman, Nemenyi and Wilcoxon statistical tests.

Friedman statistical tests. For the Friedman test, each strategy is represented by 25 values (5 datasets times 5 classifiers). As the In-processing method is run with only one classifier, it will not be shown here, but another comparison including it will be described later in this section. For the Post-processing and the Naïve methods, the best one has been selected according to the conclusions in sections 5.3.3 and 5.5.4.
At a significance level of $5.00e^{-2}$, the Friedman test concludes that at least one strategy is significantly different from the others in terms of demographic parity (p-value = 0.0010). However, it also concludes that none of the methods is significantly different from the others in terms of accuracy (p-value = 0.1321) and the modified F1-measure (p-value = 0.1610).

Nemenyi statistical tests. We can therefore conduct a post-hoc Nemenyi statistical tests to investigate which models are statistically different in terms of demographic parity.

The Nemenyi tests for the two other measures are also shown, but for information purpose. Here are the three corresponding graphs.

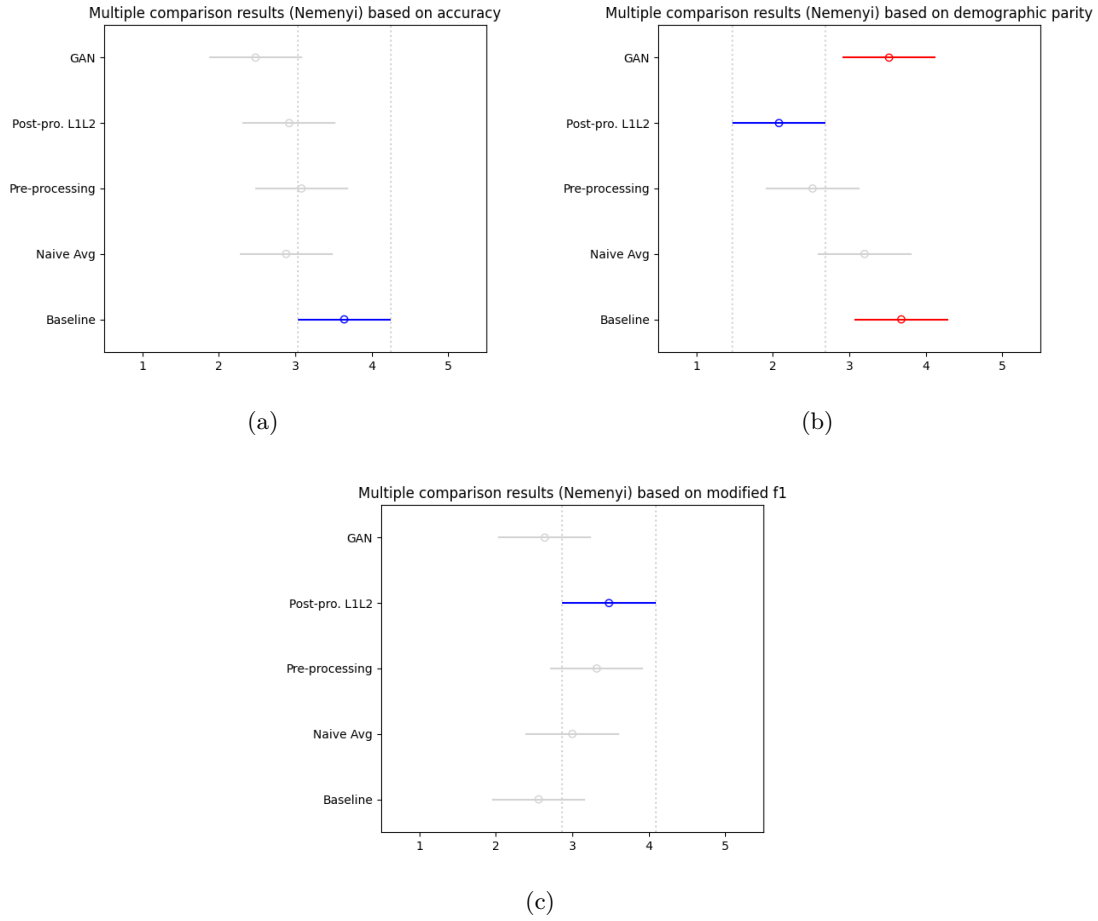


Figure 5.7: Nemenyi statistical test to compare all fairness strategies with each other according to (a) accuracy (b) the absolute value of demographic parity (c) the modified F1-measure. The blue mark is the best performing strategy i.e. the lower rank for demographic parity or the higher rank for accuracy and the modified F1-measure. The other fairness strategies are represented as a red mark if their average rank difference with the blue mark is greater than the critical distance.

As concluded by the Friedman tests, the Nemenyi tests do not detect any significant differences between the methods on any of the measures, except for the Baseline and GAN methods which produce significantly lower demographic parity scores. This is as expected given what has been discussed in the section 5.7. Overall, the Post-processing strategy with the L1L2-norm has the higher rank for both demographic parity and the modified F1-measure.

Wilcoxon statistical tests. The Wilcoxon tests from the Appendix A.1 will help us to analyse this further. In terms of accuracy, the Pre-processing strategy is found to be significantly better than both the Baseline and Naïve methods. For demographic parity, a new difference is that the Post-processing strategy is significantly better than the Naïve method. And for the modified F1-measure, only the Pre-processing and Naïve methods were found to be significantly better than the Baseline method.

5.8.2 Best results for each dataset

This section shows the 5 best method and classifier combinations for each dataset.

Method	Classifier	Accuracy	Corr. Deci.	Corr. Proba.	Dem. Par.	Modif. F1
NaivePrior	LR	0.8394	0.0094	-0.1513	0.0036	**0.9112**
NaiveAvg	MLP	0.8366	0.0053	-0.1832	0.0023	0.9101
Pre-pro.	LR	0.8397	0.0176	-0.0659	0.0084	0.9094
Naive5	DTC	0.8337	0.0025	0.0426	0.0001	0.9093
Naive5	SVM	0.8339	0.0179	0.0227	0.0005	0.9092

Table 5.19: Best results according to modified F1-measures on ADULT.

Method	Classifier	Accuracy	Corr. Deci.	Corr. Proba.	Dem. Par.	Modif. F1
Pre-pro.	SVM	0.8834	0.0033	-0.0284	0.0006	**0.9378**
Naive2	SVM	0.8824	-0.0069	-0.1499	-0.0011	0.9371
Omit Z	SVM	0.8830	-0.0197	-0.1636	-0.0021	0.9369
Baseline	SVM	0.8834	-0.0114	-0.1848	-0.0029	0.9368
Pre-pro.	LR	0.8995	-0.0209	-0.0246	-0.0238	0.9363

Table 5.20: Best results according to modified F1-measures on BANK.

Method	Classifier	Accuracy	Corr. Deci.	Corr. Proba.	Dem. Par.	Modif. F1
In-pro.	MaxEnt	0.6636	-0.0298	-0.0001	-0.0300	**0.7880**
PP L1L2	MLP	0.6689	-0.0502	-0.0004	-0.0512	0.7847
PP L1L2	LR	0.6680	-0.0484	-0.0005	-0.0494	0.7846
PP L1L2	RFC	0.6644	-0.0421	-0.0006	-0.0429	0.7843
PP L1L2	SVM	0.6675	-0.0544	-0.0005	-0.0555	0.7822

Table 5.21: Best results according to modified F1-measures on COMPAS.

Method	Classifier	Accuracy	Corr. Deci.	Corr. Proba.	Dem. Par.	Modif. F1
In-pro.	MaxEnt	0.7470	0.0077	0.0069	0.0085	**0.8521**
PP L1L2	LR	0.7510	-0.0185	-0.0001	-0.0196	0.8505
PP L2	LR	0.7450	-0.0141	-0.0001	-0.0153	0.8483
PP L1L2	DTC	0.7280	-0.0149	-0.0008	-0.0159	0.8369
PP L2	DTC	0.7280	-0.0149	-0.0014	-0.0159	0.8369

Table 5.22: Best results according to modified F1-measures on GERMAN.

Method	Classifier	Accuracy	Corr. Deci.	Corr. Proba.	Dem. Par.	Modif. F1
PP L1L2	RFC	0.9502	-0.0140	-0.0298	-0.0011	**0.9739**
Naive2	MLP	0.9501	-0.0272	-0.3497	-0.0012	**0.9739**
GAN	RFC	0.9517	-0.0505	-0.4549	0.0030	0.9738
PP L2	MLP	0.9500	0.0143	-0.0264	0.0016	0.9736
PP L1L2	MLP	0.9498	0.0154	-0.0287	0.0015	0.9735

Table 5.23: Best results according to modified F1-measures on LAW.

It can be observed that there is no clear leading method performing better than the others on all datasets. Indeed the best strategy is data dependent, but the observed best performing methods often obtain similar results. However, some interesting observations can be made:

- The Naïve methods perform relatively well on the ADULT dataset. The Naive2 strategy is also observed to perform well on the BANK and LAW datasets. These promising results encourage us to improve the Naïve strategies.
- Even though the actual GAN method should be improved to give better results, it still seems to perform better with an RFC classifier on the LAW dataset.
- The particular behaviour of the SVM classifier (see section 5.1) is observed for the BANK dataset. Two Baseline methods are in the top five, which should not happen as they don't have a fairness constraint.
- The Post-processing strategy with L1L2-norm, the In- and Pre-processing strategies seem to perform globally well on all datasets.

5.8.3 Trade-off visualisation

The same results are now presented in a more visual way. The two graphs below show the trade-off between accuracy and demographic parity for the ADULT and BANK datasets. The graphs for the other three datasets were not visually interesting, so they are not shown. The demographic parity is plotted as $1 - |DP|$.
These graphs will be used alongside our previous results, in the next sections to draw our conclusions and answer to the remaining research questions.

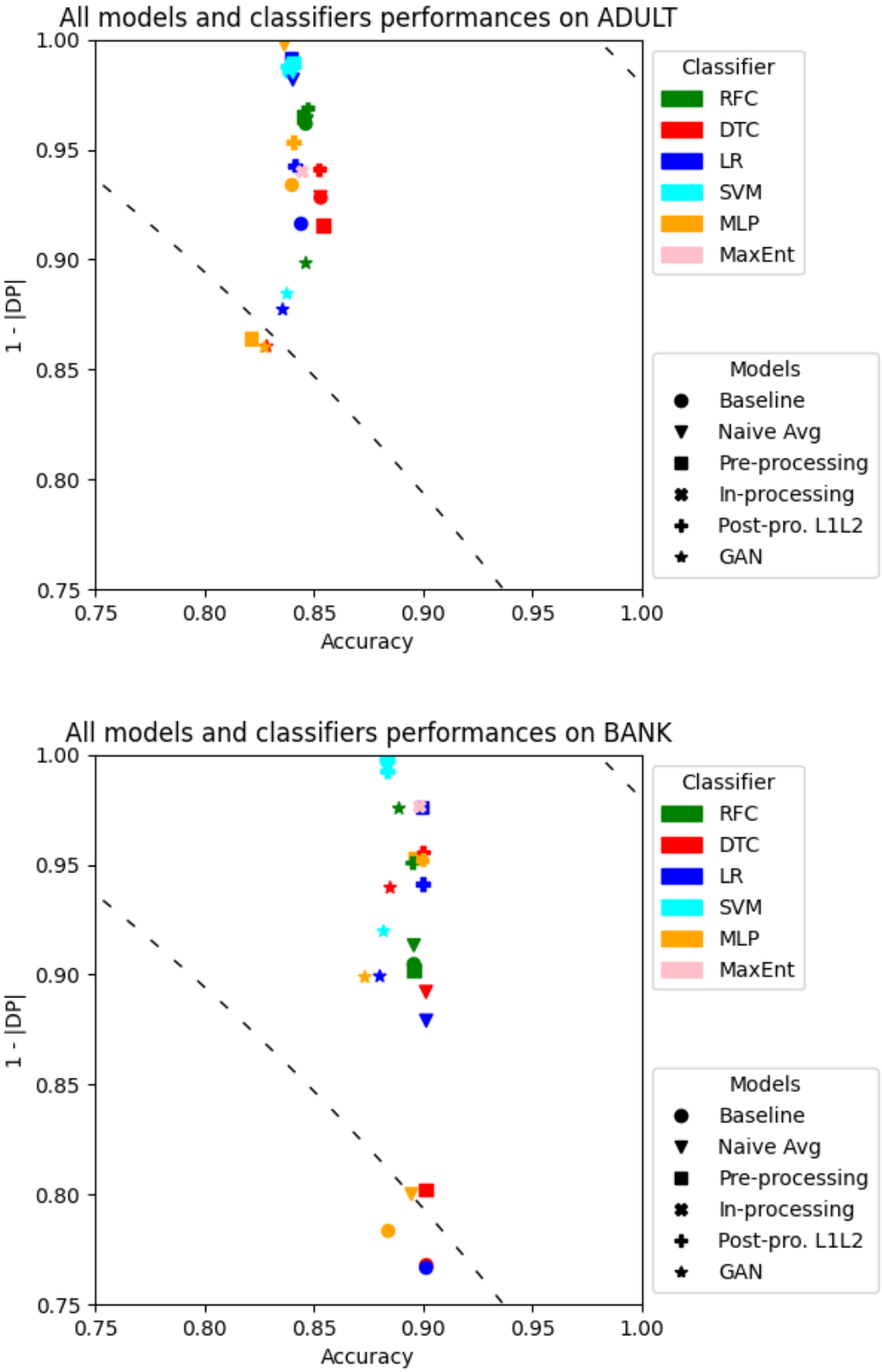


Figure 5.8: Accuracy and demographic parity trade-off visualisation for all strategy and classifier combinations on ADULT and BANK datasets. The higher the demographic parity (in ordinate) and the accuracy (in abscissa), the better.

5.8.4 Borda Rankings of the strategy and classifier combinations

Three simple Borda Rankings [17] will now be computed to rank all combinations across all datasets and to answer the remaining research questions. To obtain these Borda Rankings, we sorted all 61 strategy and classifier combinations by a selected measure for each dataset. The selected measures are the accuracy, the demographic parity and the modified F1-measure (obtained by cross-validation). If we choose the accuracy, we therefore have one ranking for each dataset. For each ranking we assign a score of 61 to the first (best) ranked combination. The second combination is given 60 points and so on until the last ranked combination is given only 1 point. We then sum up the scores of each strategy and classifier combination and divide it by 5, the number of datasets. The maximum score a combination can then get is 61 if it is ranked first five times.

These rankings depend on many parameters such as the datasets and the tuning of the models, but they should provide an idea of the overall relative results obtained on the five datasets.

Borda Ranking on the modified F1-measure. The first 24 pairs of this ranking are shown below. The complete ranking is available in the Appendix B.

Pair	**Score**	**Pair**	**Score**
Pre-processing LR	55.8	Post-processing L2 RFC	43.2
In-processing MaxEnt	50.4	Pre-processing MLP	40.0
Pre-processing SVM	50.0	GAN RFC	38.8
Post-processing L1L2 LR	48.4	Naive Prior SVM	38.4
Post-processing L1L2 SVM	47.6	Baseline RFC	38.2
Post-processing L2 MLP	45.8	Omit Z RFC	37.8
Post-processing L1L2 DTC	45.6	Omit Z SVM	37.6
Post-processing L1L2 MLP	45.6	Naive 0.5 RFC	37.2
Post-processing L1L2 RFC	45.4	Naive Avg SVM	36.8
Post-processing L2 LR	44.8	Pre-processing RFC	36.8
Post-processing L2 SVM	43.4	Naive 0.5 SVM	36.6
Post-processing L2 DTC	43.2	Naive Prior RFC	36.0

Table 5.24: Borda Ranking results on the modified F1-measure for all strategy and classifier combinations on the five datasets. The first 24 pairs are shown. The higher the score of a combination, the higher its rank. The maximum score is 61.

The Pre-processing method combined with an LR classifier is the best pair and the top ranks are dominated by the Pre-, Post- and In-processing strategies. They perform consistently well. Interesting things happen in the second half of this table where we can observe some GAN, Baseline and Naïve methods, but only with the RFC and SVM classifiers. For the SVM classifier, this is due to the annoying behaviour of classifying the majority of the instances in the same class (see section 5.1). The same cannot be

said for the RFC classifier, which seems to perform well on average.

Borda Ranking on the accuracy and the demographic parity. Two other Borda rankings have also been calculated separately for the accuracy and the demographic parity. They are available in the Appendix B. About the accuracy, the best ranked combinations are mostly Baseline and Naïve methods with LR and RFC classifiers as shown below. In terms of demographic parity, the Pre-, In- and Post-processing strategy are well performing. The Naive5 method and the SVM classifier also perform relatively well (combined and separately), as shown below.

Pair	**Score**	**Pair**	**Score**
Baseline LR	51.8	Naive Prior RFC	44.4
Omit Z LR	50.2	Post-processing L2 LR	44.2
Naive Avg LR	50.2	Naive Prior LR	44.2
Naive 0.5 LR	47.2	Naive 0.5 RFC	43.6
Baseline RFC	45.8	Omit Z RFC	43.4
Naive Avg RFC	45.0	Post-processing L1L2 LR	41.2

Table 5.25: Borda Ranking results on the accuracy for all strategy and classifier combinations on the five datasets. The first 12 pairs are shown. The higher the score of a combination, the higher its rank. The maximum score is 61.

Pair	**Score**	**Pair**	**Score**
Naive 5 SVM	55.8	In-processing MaxEnt	44.8
Pre-processing SVM	52.0	Pre-processing LR	44.4
Naive 5 RFC	50.0	Post-processing L1L2 MLP	44.0
Post-processing L1L2 SVM	48.4	Naive 2 SVM	43.6
Post-processing L2 SVM	45.6	Naive 5 MLP	43.4
Naive 5 DTC	45.4	Post-processing L2 MLP	43.2

Table 5.26: Borda Ranking results on the demographic parity for all strategy and classifier combinations on the five datasets. The first 12 pairs are shown. The higher the score of a combination, the higher its rank. The maximum score is 61.

5.8.5 General discussion

Using the above rankings and what has been shown in the previous sections, we can draw some conclusions and answer the research question number 1. Firstly, the results of the Naïve and GAN-based strategies are not convincing. The Naïve methods show some relatively good performances, but only on the ADULT dataset. The GAN-based method sometimes performs well but is too unstable and is always outperformed by other fairness strategies. Secondly, there is no classifier that strictly outperforms the others when considering all fairness strategies. However, we can say that the Logistic

Regression classifiers perform well with the Pre- and In- and Post-processing strategies. The RFC classifier also performs well when combined with the Naïve, GAN-based and Baseline methods. Finally, although the Pre-processing strategy is ranked first with the LR classifier, the best and most consistent strategy seems to be the Post-processing methods with the L1L2-norm. Indeed, the five corresponding combinations are all in the top 10. This was also observed in the section 5.8.2.

Overall the In-, Pre- and Post-processing strategies provide relatively good fairness results and the best combination of classifier and method is the Pre-processing strategy with an LR classifier. The following Borda Ranking describes how their different classifier combinations are ranked.

Pair	**Score**	**Pair**	**Score**
Pre-processing LR	12.8	Post-processing L2 DTC	8.4
In-processing MaxEnt	11.6	Post-processing L2 SVM	7.8
Pre-processing SVM	10.8	Post-processing L1L2 DTC	7.8
Post-processing L1L2 SVM	10.8	Post-processing L2 RFC	7.4
Post-processing L1L2 MLP	10.2	Post-processing L2 LR	6.8
Post-processing L1L2 RFC	9.8	Pre-processing MLP	6.0
Post-processing L2 MLP	9.8	Pre-processing RFC	4.8
Post-processing L1L2 LR	9.8	Pre-processing DTC	1.4

Table 5.27: Borda Ranking results on the modified F1-measure for the Pre-, In- and Post-processing strategies and classifiers combinations on the five datasets. The higher the score of a combination, the higher its rank. The maximum score is 16.

Chapter 6

Conclusion and future works

In this work, we have conducted a complete comparison of several fairness strategies. In this chapter, are summarised our key findings on the investigated standard methods, on the Naïve strategies and on the GAN-based strategies. Then, some ideas for future works will be stated according to what has been shown during this work.

Section 5.8 allows us to draw some conclusions. First, the best ranked method and classifier combination in terms of the modified F1-measure is the Pre-processing strategy combined with an LR classifier. Then, the Logistic Regression classifiers (LR and MaxEnt) provide overall good performances when compared to the other classification models. Finally, the Pre-, In- and Post-processing strategies are all performing relatively well. Concerning the three Pre-, In- and Post-processing methods, this work has indeed shown that they perform relatively well in the reduction of the discrimination, without reducing the accuracy too much, at least on the datasets used and compared to the others analysed strategies. It would therefore be interesting to tackle more complex problems in order to differentiate the methods. For example, working on other metrics that are more subtle than simple demographic parity, dealing with multi-class problems that go beyond binary classification, or dealing with problems that contain several protected variables, both categorical and numerical.

About the Naïve strategies, if we consider the first three ones, they delivered similar performances. This work has demonstrated that they produce an equivalent level of accuracy to the Pre-, In- and Post-processing methods, but they produced significantly lower fairness scores. However, interesting results were observed on the ADULT dataset, where they produce good performances in the reduction of the discrimination. About the Naive2 and Naive5 strategies, they provide globally better results than the other Naïve methods, but are more unstable as they do not provide results for every execution. The performances in terms of fairness of the Naïve strategies seems to be context dependent. They have shown some promising results, but in general the observed effects can be quite different from one dataset or classifier to another. It would therefore be interesting to improve these strategies by adapting them to the dataset and classifier they are applied to. For example, by taking into account the number of columns that are highly correlated with the protected attribute and those that are too important for the accuracy to be transformed.

Finally, the performances of the GAN-based strategy in its current state are not convincing. In fact, the GAN-based method sometimes performs well, but is too unstable and is always surpassed by other fairness strategies. However, considering the standard deviation of the results in section 5.7, we can foresee some interesting performances

that encourage us to improve this model. The most important thing is therefore to stabilise the method, for example by using a validation set. Furthermore, the results of the GAN-based method were obtained with a default configuration. It would thus be interesting to tune the various parameters in order to further optimise the results for each dataset and classifier combination.

Future works. As a continuation of this work, there are several promising directions for future research. To better distinguish and classify the methods, it would be interesting to work on other fairness metrics such as the Equal Opportunity, to deal with multiple classes or multiple protected variables problems and to tune the classification models. We could also remove the protected attribute from the training of some fairness models. The Naïve and GAN-based strategies need to be the subject of more in-depth investigations especially the GAN, Naive2 and Naive5 strategies. It might also be interesting to find another fairness strategy based on a Generative Adversarial Network and adapt the training and preprocessing to be similar to the other fairness strategies in order to provide more reliable comparisons. A final but important point is to find some new reliable datasets to use in future studies.

References

[1] Akshay Agrawal, Robin Verschueren, Steven Diamond, and Stephen Boyd. A rewriting system for convex optimization problems. *Journal of Control and Decision*, 5(1):42–60, 2018.

[2] E. Alpaydin. Introduction to machine learning, fourth edition, 2020.

[3] MOSEK ApS. *MOSEK Optimizer API for C*.

[4] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 214–223. JMLR.org, 2017.

[5] Solon Barocas, Moritz Hardt, and Arvind Narayanan. Fairness and machine learning: Limitations and opportunities. fairmlbook.org, 2019. `http://www.fairmlbook.org`.

[6] Toon Calders, Faisal Kamiran, and Mykola Pechenizkiy. Building classifiers with independency constraints. In *2009 IEEE International Conference on Data Mining Workshops*, pages 13–18, 2009.

[7] Toon Calders and Sicco Verwer. Three naive bayes approaches for discrimination-free classification. *Data Mining and Knowledge Discovery*, 21:277–292, 09 2010.

[8] Alessandro Castelnovo, Riccardo Crupi, Greta Greco, Daniele Regoli, Ilaria Giuseppina Penco, and Andrea Claudio Cosentini. The zoo of fairness metrics in machine learning. 2021.

[9] Alessandro Castelnovo, Riccardo Crupi, Greta Greco, Daniele Regoli, Ilaria Giuseppina Penco, and Andrea Claudio Cosentini. A clarification of the nuances in the fairness metrics landscape. *Scientific reports*, 12(1):4209, March 2022.

[10] Thomas M. Cover and Joy A. Thomas. Maximum entropy. In *Elements of Information Theory*, pages 409–425. John Wiley and Sons, Ltd, 2005.

[11] Antonia Creswell, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta, and Anil A. Bharath. Generative adversarial networks: An overview. *IEEE Signal Processing Magazine*, 35(1):53–65, 2018.

[12] Constantin de Schaetzen. Increasing fairness in supervised classification: a study of simple decorrelation methods applied to the logistic regression model. Master Thesis, UCLouvain, École polytechnique de Louvain, 2020-2021.

[13] Pieter Delobelle, Paul Temple, Gilles Perrouin, Benoit Frénay, Patrick Heymans, and Bettina Berendt. Ethical adversaries: Towards mitigating unfairness with adversarial machine learning. *SIGKDD Explor. Newsl.*, 23(1):32–41, may 2021.

[14] Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, dec 2006.

[15] Steven Diamond and Stephen P. Boyd. Cvxpy: A python-embedded modeling language for convex optimization. *Journal of Machine Learning Research : JMLR*, 17, 2016.

[16] Cambridge Academic Content Dictionary. Fairness. Cambridge University Press, 2008.

[17] Peter Emerson. The original borda count and partial voting. *Social Choice and Welfare*, 40(2):353–358, 2013.

[18] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. Liblinear: A library for large linear classification. *Journal of Machine Learning Research*, 9(Aug):1871–1874, 2008.

[19] Milton Friedman. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association*, 32(200):675–701, 1937.

[20] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

[21] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[22] Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

[23] E. T. Jaynes. Information theory and statistical mechanics. *Physical Review*, 106:620–630, May 1957.

[24] Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, and Jun Sakuma. Fairness-aware classifier with prejudice remover regularizer. In Peter A. Flach, Tijl De Bie, and Nello Cristianini, editors, *Machine Learning and Knowledge Discovery in Databases*, pages 35–50, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[25] Alexia Kneip and Eve Beghein. Fairness in supervised classification: investigation of three different techniques. Master Thesis, UCLouvain, Louvain School of Management, 2021-2022.

[26] M.L. Menéndez, J.A. Pardo, L. Pardo, and M.C. Pardo. The Jensen-Shannon divergence. *Journal of the Franklin Institute*, 334(2):307–318, 1997.

[27] P. Nemenyi. Distribution-free multiple comparisons. Princeton University, 1963.

[28] Luca Oneto and Silvia Chiappa. Fairness in machine learning. *Computing Research Repository*, abs/2012.15816, 2020.

[29] Dana Pessach and Erez Shmueli. Algorithmic fairness. *Computing Research Repository*, abs/2001.09784, 2020.

[30] Devin G. Pope and Justin R. Sydnor. Implementing anti-discrimination policies in statistical profiling models. *American Economic Journal: Economic Policy*, 3(3):206–31, August 2011.

[31] Propublica. Compas recidivism risk score data and analysis. `https://www.propublica.org/datastore/dataset/compas-recidivism-risk-score-data-and-analysis` (2023).

[32] Tai Le Quy, Arjun Roy, Vasileios Iosifidis, Wenbin Zhang, and Eirini Ntoutsi. A survey on datasets for fairness-aware machine learning. *WIREs Data Mining and Knowledge Discovery*, 12(3), mar 2022.

[33] Amirarsalan Rajabi and Ozlem Ozmen Garibay. Tabfairgan: Fair tabular data generation with generative adversarial networks. *Machine Learning and Knowledge Extraction*, 4(2):488–501, 2022.

[34] UC Irvine Machine Learning Repository. Adult dataset. `https://archive.ics.uci.edu/ml/machine-learning-databases/adult/`.

[35] UC Irvine Machine Learning Repository. Bank dataset. `https://archive.ics.uci.edu/ml/datasets/bank+marketing`.

[36] UC Irvine Machine Learning Repository. German dataset. `https://archive.ics.uci.edu/ml/datasets/statlog+%28german+credit+data)`.

[37] Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40:99–121, 2000.

[38] Bashir Sadeghi and Vishnu Naresh Boddeti. Imparting fairness to pre-trained biased representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2020.

[39] M. Saerens. A brief introduction to fairness in supervised classification. Data mining and decision making course. Unpublished manuscript. Catholic University of Louvain, 2022.

[40] M. Saerens. A note on increasing fairness in prediction by simple linear subspace projection. Unpublished manuscript. Catholic University of Louvain, 2022.

[41] P.-N. Tan, M. Steinbach, A. Karpatne, and V. Kumar. Introduction to Data Mining. Pearson Education. 2018.

[42] Flore Vancompernolle Vromman, Sylvain Courtain, Pierre Leleux, Constantin de Schaetzen, Eve Beghein, Alexia Kneip, and Marco Saerens. Maximum entropy logistic regression for demographic parity in supervised classification. Submitted for publication. LouRIM & ICTEAM, Catholic University of Louvain, 2023.

[43] Sahil Verma and Julia Rubin. Fairness definitions explained. In *Proceedings of the International Workshop on Software Fairness*, FairWare '18, page 1–7, New York, NY, USA, 2018. Association for Computing Machinery.

[44] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, et al., and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

[45] Kunfeng Wang, Chao Gou, Yanjie Duan, Yilun Lin, Xinhu Zheng, and Fei-Yue Wang. Generative adversarial networks: introduction and outlook. *IEEE/CAA Journal of Automatica Sinica*, 4(4):588–598, 2017.

[46] Linda F. Wightman. LSAC National Longitudinal Bar Passage Study. LSAC Research Report Series. 1998.

[47] Frank Wilcoxon. Individual comparisons by ranking methods. volume 1, pages 80–83. [International Biometric Society, Wiley], 1945.

[48] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rodriguez, and Krishna P. Gummadi. Fairness constraints: Mechanisms for fair classification. 2017.

[49] Zhe Zhang, Shenhang Wang, and Gong Meng. A review on pre-processing methods for fairness in machine learning. In Ning Xiong, Maozhen Li, Kenli Li, Zheng Xiao, Longlong Liao, and Lipo Wang, editors, *Advances in Natural Computation, Fuzzy Systems and Knowledge Discovery*, pages 1185–1191, Cham, 2023. Springer International Publishing.

[50] Hong Zhu and Muhammad Usman. Research on human resource recommendation algorithm based on machine learning. *Science of Computer Programming*, 2021, jan 2021.

[51] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 67(2):301–320, 2005.

Appendices

A Statistical tests

A.1 Comparisons between all fairness strategies

	Baseline	Naive Avg	Pre-processing	Post-pro. L1L2	GAN
Baseline	1				
Naive Avg	0.7510	1			
Pre-processing	**0.0275**	0.8532	1		
Post-pro. L1L2	0.8949	0.9578	1	1	
GAN	**0.0061**	0.9789	**0.0115**	0.6150	1

Table 6.1: P-values of the Wilcoxon signed-ranks tests comparing the accuracy scores of all fairness strategies (two by two). The difference between two strategies is significant if the corresponding p-value is lower than $5.00e^{-2}$.

	Baseline	Naive Avg	Pre-processing	Post-pro. L1L2	GAN
Baseline	1				
Naive Avg	0.4742	1			
Pre-processing	**0.0038**	0.4908	1		
Post-pro. L1L2	**0.0006**	**0.0042**	0.0851	1	
GAN	0.5249	0.5424	**0.0367**	**6.366E-05**	1

Table 6.2: P-values of the Wilcoxon signed-ranks tests comparing the demographic parity scores of all fairness strategies (two by two). The difference between two strategies is significant if the corresponding p-value is lower than $5.00e^{-2}$.

	Baseline	Naive Avg	Pre-processing	Post-pro. L1L2	GAN
Baseline	1				
Naive Avg	0.8119	1			
Pre-processing	**0.0081**	0.6150	1		
Post-pro. L1L2	0.0802	0.4108	0.2411	1	
GAN	0.9578	0.9789	**0.0187**	0.1014	1

Table 6.3: P-values of the Wilcoxon signed-ranks tests comparing the modified F1-measure scores of all fairness strategies (two by two). The difference between two strategies is significant if the corresponding p-value is lower than $5.00e^{-2}$.

A.2 Comparisons between Naive methods

	Naive Prior	Naive Avg	Naive 0.5	Naive 2	Naive 5	Baseline
Naive Prior	1					
Naive Avg	0.1453	1				
Naive 0.5	0.4781	0.3520	1			
Naive 2	**0.0002**	**9.067E-05**	**0.0014**	1		
Naive 5	**4.590E-05**	**2.071E-05**	**0.0002**	0.1178	1	
Baseline	0.7112	0.7510	0.6915	0.4261	0.3957	1

Table 6.4: P-values of the Wilcoxon signed-ranks tests comparing the accuracy scores of Naive strategies (two by two). The difference between two strategies is significant if the corresponding p-value is lower than $5.00e^{-2}$.

	Naive Prior	Naive Avg	Naive 0.5	Naive 2	Naive 5	Baseline
Naive Prior	1					
Naive Avg	0.4140	1				
Naive 0.5	0.6894	0.0627	1			
Naive 2	**0.0018**	**0.0039**	**0.0005**	1		
Naive 5	**7.144E-05**	**0.0002**	**3.027E-05**	**0.0001**	1	
Baseline	0.6721	0.4742	0.5249	**0.0422**	**0.0001**	1

Table 6.5: P-values of the Wilcoxon signed-ranks tests comparing the demographic parity scores of Naive strategies (two by two). The difference between two strategies is significant if the corresponding p-value is lower than $5.00e^{-2}$.

	Naive Prior	Naive Avg	Naive 0.5	Naive 2	Naive 5	Baseline
Naive Prior	1					
Naive Avg	0.4549	1				
Naive 0.5	0.7943	**0.0299**	1			
Naive 2	0.2301	**0.0487**	0.2652	1		
Naive 5	**0.0039**	**0.0003**	**0.0025**	**0.0066**	1	
Baseline	0.9578	0.8119	0.8949	0.2411	**0.0007**	1

Table 6.6: P-values of the Wilcoxon signed-ranks tests comparing the modified F1-measure scores of Naive strategies (two by two). The difference between two strategies is significant if the corresponding p-value is lower than $5.00e^{-2}$.

A.3 Comparisons between In-processing and Post-processing strategies

	In-processing	PP L1L2 RFC	PP L1L2 DTC	PP L1L2 LR	PP L1L2 SVM	PP L1L2 MLP
In-processing	1					
PP L1L2 RFC	0.8740	1				
PP L1L2 DTC	0.6528	0.5580	1			
PP L1L2 LR	0.8740	0.6476	0.2735	1		
PP L1L2 SVM	0.3123	0.2296	**0.0177**	**0.0004**	1	
PP L1L2 MLP	0.4418	0.4654	0.1909	0.1409	**0.0147**	1

Table 6.7: P-values of the Wilcoxon signed-ranks tests comparing the accuracy scores of the In-processing strategy and the L1L2-norm of the Post-processing strategy. The difference between two strategies is significant if the corresponding p-value is lower than $5.00e^{-2}$.

	In-processing	PP L1L2 RFC	PP L1L2 DTC	PP L1L2 LR	PP L1L2 SVM	PP L1L2 MLP
In-processing	1					
PP L1L2 RFC	0.4261	1				
PP L1L2 DTC	0.6338	**0.0237**	1			
PP L1L2 LR	0.8740	0.2872	0.0710	1		
PP L1L2 SVM	0.2872	0.3525	**0.0096**	**0.0177**	1	
PP L1L2 MLP	0.4418	0.9368	**0.0191**	0.1073	**0.0203**	1

Table 6.8: P-values of the Wilcoxon signed-ranks tests comparing the demographic parity scores of the In-processing strategy and the L1L2-norm of the Post-processing strategy. The difference between two strategies is significant if the corresponding p-value is lower than $5.00e^{-2}$.

	In-processing	PP L1L2 RFC	PP L1L2 DTC	PP L1L2 LR	PP L1L2 SVM	PP L1L2 MLP
In-processing	1					
PP L1L2 RFC	0.8949	1				
PP L1L2 DTC	0.5424	0.4418	1			
PP L1L2 LR	1	0.9368	0.0851	1		
PP L1L2 SVM	0.9368	0.4261	**0.0004**	0.7915	1	
PP L1L2 MLP	0.5602	0.7915	0.1409	0.6721	0.9158	1

Table 6.9: P-values of the Wilcoxon signed-ranks tests comparing the modified F1-measure scores of the In-processing strategy and the L1L2-norm of the Post-processing strategy. The difference between two strategies is significant if the corresponding p-value is lower than $5.00e^{-2}$.

A.4 Comparisons between logistic regressions

	In-processing	PP L1L2 LR	Pre-processing LR
In-processing	1		
PP L1L2 LR	0.8740	1	
Pre-processing LR	0.4908	0.8949	1

(a)

	In-processing	PP L1L2 LR	Pre-processing LR
In-processing	1		
PP L1L2 LR	0.8740	1	
Pre-processing LR	0.0802	0.2635	1

(b)

	In-processing	PP L1L2 LR	Pre-processing LR
In-processing	1		
PP L1L2 LR	1	1	
Pre-processing LR	0.1336	0.9158	1

(c)

Table 6.10: P-values of the Wilcoxon signed-ranks tests comparing (a) the accuracy scores (b) the demographic parity scores (c) the modified F1-measure scores of the In-processing strategy, the L1L2-norm of the Post-processing strategy and the Pre-processing strategy both with an LR classifier. The difference between two strategies is significant if the corresponding p-value is lower than $5.00e^{-2}$.

A.5 Comparisons between the five gradation stages to improve fairness

	Baseline	Ommit Z	Naive Avg	Pre-processing	Post-pro. L1L2
Baseline	1				
Ommit Z	0.9861	1			
Naive Avg	0.7510	0.7712	1		
Pre-processing	**0.0275**	0.0675	0.8532	1	
Post-pro. L1L2	0.8949	0.8532	0.9578	1	1

Table 6.11: P-values of the Wilcoxon signed-ranks tests comparing the accuracy scores of the 5 gradation stages to improve fairness. The difference between two stages is significant if the corresponding p-value is lower than $5.00e^{-2}$.

	Baseline	Ommit Z	Naive Avg	Pre-processing	Post-pro. L1L2
Baseline	1				
Ommit Z	**0.0051**	1			
Naive Avg	0.4742	0.9789	1		
Pre-processing	**0.0038**	**0.0451**	0.4908	1	
Post-pro. L1L2	**0.0006**	**0.0020**	**0.0042**	0.0851	1

Table 6.12: P-values of the Wilcoxon signed-ranks tests comparing the demographic parity scores of the 5 gradation stages to improve fairness. The difference between two stages is significant if the corresponding p-value is lower than $5.00e^{-2}$.

	Baseline	Ommit Z	Naive Avg	Pre-processing	Post-pro. L1L2
Baseline	1				
Ommit Z	**0.0035**	1			
Naive Avg	0.8119	0.9368	1		
Pre-processing	**0.0081**	0.0755	0.6150	1	
Post-pro. L1L2	0.0802	0.1563	0.4108	0.2411	1

Table 6.13: P-values of the Wilcoxon signed-ranks tests comparing the modified F1-measure scores of the 5 gradation stages to improve fairness. The difference between two stages is significant if the corresponding p-value is lower than $5.00e^{-2}$.

A.6 Comparisons between the GAN-based strategies with and without fairness

	Baseline GAN	GAN
Baseline GAN	1	
GAN	0.3011	1

(a)

	Baseline GAN	GAN
Baseline GAN	1	
GAN	**0.0010**	1

(b)

	Baseline GAN	GAN
Baseline GAN	1	
GAN	**0.0002**	1

(c)

Table 6.14: P-values of the Wilcoxon signed-ranks tests comparing (a) the accuracy scores (b) the demographic parity scores (c) the modified F1-measure scores of the GAN-based strategies with and without fairness. The difference between the two strategies is significant if the corresponding p-value is lower than $5.00e^{-2}$.

A.7 Comparisons between the GAN-based and the Pre-processing strategies

	Pre-processing	GAN
Pre-processing	1	
GAN	**0.0187**	1

(a)

	Pre-processing	GAN
Pre-processing	1	
GAN	**0.0038**	1

(b)

	Pre-processing	GAN
Pre-processing	1	
GAN	**0.0275**	1

(c)

Table 6.15: P-values of the Wilcoxon signed-ranks tests comparing (a) the accuracy scores (b) the demographic parity scores (c) the modified F1-measure scores of the GAN-based and the Pre-processing strategies. The difference between the two strategies is significant if the corresponding p-value is lower than $5.00e^{-2}$.

B Borda Rankings of all strategy and classifier combinations

B.1 Borda Ranking on the accuracy

Pair	Score	Pair	Score
Baseline LR	51.8	Naive Avg SVM	30.2
Omit Z LR	50.2	Baseline MLP	29.8
Naive Avg LR	50.2	Naive 0.5 DTC	29.6
Naive 0.5 LR	47.2	Naive 0.5 MLP	29.6
Baseline RFC	45.8	Baseline GAN SVM	29.6
Naive Avg RFC	45.0	Baseline GAN LR	28.6
Naive Prior RFC	44.4	Naive Avg MLP	28.4
Post-processing L2 LR	44.2	Naive Prior MLP	28.2
Naive Prior LR	44.2	Baseline GAN DTC	27.8
Naive 0.5 RFC	43.6	GAN SVM	27.6
Omit Z RFC	43.4	GAN LR	27.2
Post-processing L1L2 LR	41.2	Pre-processing MLP	27.0
Omit Z DTC	41.0	Naive 2 LR	26.8
Baseline DTC	40.6	Baseline SVM	26.2
Pre-processing RFC	40.4	Omit Z SVM	25.6
Naive Prior DTC	40.0	Naive 2 DTC	24.2
Post-processing L2 RFC	38.2	Baseline GAN MLP	24.0
Naive Avg DTC	37.8	Post-processing L2 SVM	23.0
Post-processing L1L2 DTC	35.8	Post-processing L1L2 SVM	22.2
GAN RFC	35.6	Pre-processing SVM	21.0
Post-processing L2 DTC	35.4	Naive 5 LR	21.0
Pre-processing DTC	34.8	Naive 2 RFC	18.2
Omit Z MLP	34.8	Naive 5 RFC	15.4
In-processing MaxEnt	34.6	GAN DTC	12.8
Pre-processing LR	34.2	Naive 5 DTC	12.6
Baseline GAN RFC	33.0	Naive 5 MLP	12.0
Post-processing L1L2 RFC	32.8	GAN MLP	12.0
Post-processing L2 MLP	31.2	Naive 5 SVM	10.2
Post-processing L1L2 MLP	31.0	Naive 2 MLP	9.8
Naive 0.5 SVM	31.0	Naive 2 SVM	6.6
Naive Prior SVM	30.4		

Table 6.16: Borda Ranking results on the accuracy for all strategy and classifier combinations on the five datasets. The higher the score of a combination, the higher its rank. The maximum score is 61.

B.2 Borda Ranking on the demographic parity

Pair	Score	Pair	Score
Naive 5 SVM	55.8	Baseline RFC	30.6
Pre-processing SVM	52.0	Pre-processing RFC	30.4
Naive 5 RFC	50.0	Naive 0.5 RFC	30.0
Post-processing L1L2 SVM	48.4	Naive Prior RFC	28.6
Post-processing L2 SVM	45.6	Naive Avg RFC	28.4
Naive 5 DTC	45.4	Naive 0.5 MLP	28.0
In-processing MaxEnt	44.8	GAN DTC	26.0
Pre-processing LR	44.4	Naive Prior LR	25.8
Post-processing L1L2 MLP	44.0	Omit Z MLP	25.4
Naive 2 SVM	43.6	Naive 2 DTC	25.0
Naive 5 MLP	43.4	Naive Prior MLP	24.8
Post-processing L2 MLP	43.2	GAN SVM	24.0
Post-processing L1L2 RFC	42.6	GAN LR	23.4
Naive 2 RFC	41.6	Baseline GAN RFC	23.0
Post-processing L1L2 LR	41.2	Naive Avg LR	21.2
Naive 5 LR	39.6	Baseline MLP	21.2
Naive 2 MLP	39.4	Naive Avg DTC	20.0
Post-processing L1L2 DTC	39.4	Naive 0.5 DTC	20.0
Post-processing L2 LR	39.0	Naive 0.5 LR	19.0
Pre-processing MLP	39.0	GAN MLP	18.8
Post-processing L2 RFC	37.8	Omit Z DTC	18.4
Post-processing L2 DTC	37.2	Pre-processing DTC	16.0
Naive Prior SVM	36.6	Baseline GAN MLP	16.0
GAN RFC	36.0	Baseline GAN DTC	14.4
Naive Avg SVM	35.8	Omit Z LR	12.8
Naive 2 LR	35.2	Naive Prior DTC	11.4
Naive 0.5 SVM	34.6	Baseline GAN SVM	11.4
Omit Z SVM	34.0	Baseline GAN LR	11.2
Baseline SVM	33.6	Baseline DTC	10.2
Naive Avg MLP	31.6	Baseline LR	10.2
Omit Z RFC	30.6		

Table 6.17: Borda Ranking results on the demographic parity for all strategy and classifier combinations on the five datasets. The higher the score of a combination, the higher its rank. The maximum score is 61.

B.3 Borda Ranking on the modified F1-measure

Pair	Score	Pair	Score
Pre-processing LR	55.8	GAN DTC	29.0
In-processing MaxEnt	50.4	Naive Avg LR	28.8
Pre-processing SVM	50.0	Naive 0.5 DTC	28.0
Post-processing L1L2 LR	48.4	Omit Z DTC	27.6
Post-processing L1L2 SVM	47.6	Naive 0.5 LR	26.6
Post-processing L2 MLP	45.8	Naive 2 RFC	26.2
Post-processing L1L2 DTC	45.6	Naive Prior MLP	26.0
Post-processing L1L2 MLP	45.6	Naive Avg DTC	26.0
Post-processing L1L2 RFC	45.4	Naive 2 SVM	25.6
Post-processing L2 LR	44.8	GAN SVM	24.6
Post-processing L2 SVM	43.4	GAN LR	24.0
Post-processing L2 DTC	43.2	Naive 2 DTC	23.8
Post-processing L2 RFC	42.2	Baseline MLP	23.4
Pre-processing MLP	40.0	Naive 5 LR	22.6
GAN RFC	38.8	Naive 5 DTC	21.6
Naive Prior SVM	38.4	GAN MLP	21.4
Baseline RFC	38.2	Pre-processing DTC	21.2
Omit Z RFC	37.8	Naive Prior DTC	21.0
Omit Z SVM	37.6	Baseline GAN MLP	19.4
Naive 0.5 RFC	37.2	Omit Z LR	19.0
Naive Avg SVM	36.8	Baseline DTC	17.8
Pre-processing RFC	36.8	Baseline LR	17.4
Naive 0.5 SVM	36.6	Baseline GAN SVM	16.8
Naive Prior RFC	36.0	Naive 2 MLP	16.8
Naive Avg RFC	35.8	Naive 5 RFC	16.6
Naive Prior LR	35.6	Baseline GAN LR	16.2
Baseline SVM	35.4	Naive 5 SVM	15.8
Naive 2 LR	33.0	Baseline GAN DTC	14.2
Naive Avg MLP	31.8	Baseline GAN RFC	13.6
Omit Z MLP	31.0	Naive 5 MLP	6.0
Naive 0.5 MLP	29.0		

Table 6.18: Borda Ranking results on the modified F1-measure for all strategy and classifier combinations on the five datasets. The higher the score of a combination, the higher its rank. The maximum score is 61.

C Thresholds values for the In- and Post-processing strategies

ϵ_0	ϵ_1	ϵ_2	ϵ_3	ϵ_4
0.0147	0.0164	0.0180	0.0196	0.0212
ϵ_5	ϵ_6	ϵ_7	ϵ_8	ϵ_9
0.0229	0.0245	0.0261	0.0277	0.0294

(a)

ϵ_0	ϵ_1	ϵ_2	ϵ_3	ϵ_4
0.0009	0.0020	0.0031	0.0041	0.0052
ϵ_5	ϵ_6	ϵ_7	ϵ_8	ϵ_9
0.0063	0.0073	0.0084	0.0095	0.0105

(b)

ϵ_0	ϵ_1	ϵ_2	ϵ_3	ϵ_4
0.00006	0.0033	0.0065	0.0097	0.0129
ϵ_5	ϵ_6	ϵ_7	ϵ_8	ϵ_9
0.0161	0.0193	0.0225	0.0257	0.0289

(c)

ϵ_0	ϵ_1	ϵ_2	ϵ_3	ϵ_4
0.000004	0.0026	0.0051	0.0077	0.0102
ϵ_5	ϵ_6	ϵ_7	ϵ_8	ϵ_9
0.0128	0.0153	0.0179	0.0205	0.0230

(d)

ϵ_0	ϵ_1	ϵ_2	ϵ_3	ϵ_4
0.0007	0.0023	0.0039	0.0055	0.0071
ϵ_5	ϵ_6	ϵ_7	ϵ_8	ϵ_9
0.0087	0.0103	0.0119	0.0135	0.0151

(e)

Table 6.19: Thresholds values on (a) ADULT (b) BANK (c) COMPAS (d) GERMAN (e) LAW for the In- and the Post-processing strategies.

D Baseline strategy

D.1 Results overview

On the ADULT dataset

Method	Classifier	Accuracy	Corr. Deci.	Corr. Proba.	Dem. Par.	Modif. F1
Baseline	DTC	0.8531	-0.1244	-0.3063	-0.0718	0.8891
Baseline	LR	0.8442	-0.1496	-0.3474	-0.0838	0.8787
Baseline	MLP	0.8400	-0.1090	-0.3282	-0.0661	0.8845
Baseline	RFC	0.8463	-0.0968	-0.3733	-0.0383	0.9003
Baseline	SVM	0.8389	-0.0529	-0.2673	-0.0149	**0.9061**
Omit Z	DTC	0.8531	-0.1244	-0.3027	-0.0718	0.8891
Omit Z	LR	0.8448	-0.1328	-0.2955	-0.0733	0.8839
Omit Z	MLP	0.8469	-0.1253	-0.2814	-0.0755	0.8840
Omit Z	RFC	0.8465	-0.0924	-0.3354	-0.0371	0.9010
Omit Z	SVM	0.8376	-0.0371	-0.2260	-0.0087	**0.9080**

Table 6.20: Results of the Baseline and Omit Z strategies on ADULT.

On the BANK dataset

Method	Classifier	Accuracy	Corr. Deci.	Corr. Proba.	Dem. Par.	Modif. F1
Baseline	DTC	0.9014	-0.1855	-0.2540	-0.2321	0.8293
Baseline	LR	0.9014	-0.1981	-0.2866	-0.2333	0.8286
Baseline	MLP	0.8840	-0.1860	-0.2310	-0.2165	0.8307
Baseline	RFC	0.8957	-0.1344	-0.3171	-0.0952	0.9002
Baseline	SVM	0.8834	-0.0114	-0.1848	-0.0029	**0.9368**
Omit Z	DTC	0.9008	-0.0992	-0.1397	-0.1213	0.8896
Omit Z	LR	0.9010	-0.1282	-0.1971	-0.1502	0.8747
Omit Z	MLP	0.8836	-0.1218	-0.1510	-0.1753	0.8531
Omit Z	RFC	0.8958	-0.1229	-0.1999	-0.0883	0.9037
Omit Z	SVM	0.8830	-0.0197	-0.1636	-0.0021	**0.9369**

Table 6.21: Results of the Baseline and Omit Z strategies on BANK.

On the COMPAS dataset

Method	Classifier	Accuracy	Corr. Deci.	Corr. Proba.	Dem. Par.	Modif. F1
Baseline	DTC	0.6683	-0.2572	-0.2592	-0.2596	0.7025
Baseline	LR	0.6785	-0.2382	-0.3005	-0.2375	0.7181
Baseline	MLP	0.6834	-0.2397	-0.2902	-0.2398	**0.7198**
Baseline	RFC	0.6808	-0.2392	-0.3182	-0.2404	0.7181
Baseline	SVM	0.6785	-0.2459	-0.3030	-0.2457	0.7144
Omit Z	DTC	0.6719	-0.2213	-0.2637	-0.2226	**0.7208**
Omit Z	LR	0.6789	-0.2421	-0.3048	-0.2413	0.7166
Omit Z	MLP	0.6776	-0.2317	-0.2880	-0.2320	0.7199
Omit Z	RFC	0.6784	-0.2382	-0.2946	-0.2387	0.7175
Omit Z	SVM	0.6790	-0.2416	-0.3052	-0.2414	0.7166

Table 6.22: Results of the Baseline and Omit Z strategies on COMPAS.

On the GERMAN dataset

Method	Classifier	Accuracy	Corr. Deci.	Corr. Proba.	Dem. Par.	Modif. F1
Baseline	DTC	0.7290	-0.0442	-0.1074	-0.0478	0.8258
Baseline	LR	0.7520	-0.1997	-0.2392	-0.2186	0.7664
Baseline	MLP	0.6950	-0.0187	-0.1033	-0.0149	0.8150
Baseline	RFC	0.7160	0.0058	-0.2628	0.0032	**0.8334**
Baseline	SVM	0.7030	-0.0200	-0.0266	-0.0028	0.8246
Omit Z	DTC	0.7290	-0.0442	-0.0976	-0.0478	0.8258
Omit Z	LR	0.7350	-0.1156	-0.1470	-0.1242	0.7993
Omit Z	MLP	0.7090	0.0139	0.0433	0.0067	0.8274
Omit Z	RFC	0.7160	0.0142	-0.1665	0.0082	**0.8316**
Omit Z	SVM	0.7050	-0.0379	-0.0182	-0.0068	0.8246

Table 6.23: Results of the Baseline and Omit Z strategies on GERMAN.

On the LAW dataset

Method	Classifier	Accuracy	Corr. Deci.	Corr. Proba.	Dem. Par.	Modif. F1
Baseline	DTC	0.9491	-0.2087	-0.3668	-0.0721	0.9384
Baseline	LR	0.9514	-0.1982	-0.4357	-0.0534	0.9490
Baseline	MLP	0.9508	-0.1759	-0.4432	-0.0404	0.9551
Baseline	RFC	0.9517	-0.1486	-0.4538	-0.0278	**0.9618**
Baseline	SVM	0.9509	-0.2130	-0.4465	-0.0609	0.9449
Omit Z	DTC	0.9492	-0.2032	-0.3618	-0.0695	0.9397
Omit Z	LR	0.9514	-0.1835	-0.3975	-0.0490	0.9512
Omit Z	MLP	0.9511	-0.1415	-0.3989	-0.0277	0.9616
Omit Z	RFC	0.9515	-0.1444	-0.4261	-0.0279	**0.9617**
Omit Z	SVM	0.9507	-0.2109	-0.4025	-0.0617	0.9445

Table 6.24: Results of the Baseline and Omit Z strategies on LAW.

E Pre-processing strategy

E.1 Results overview

On the ADULT dataset

Classifier	Accuracy	Corr. Deci.	Corr. Proba.	Dem. Par.	Modif. F1
DTC	0.8543	-0.1369	-0.3002	-0.0846	0.8838
LR	0.8397	0.0176	-0.0659	0.0084	**0.9094**
MLP	0.8212	-0.1995	-0.2773	-0.1360	0.8420
RFC	0.8450	-0.0926	-0.3912	-0.0351	0.9010
SVM	0.8406	-0.0356	0.0055	-0.0104	0.9090

Table 6.25: Results of the Pre-processing strategy on ADULT.

On the BANK dataset

Classifier	Accuracy	Corr. Deci.	Corr. Proba.	Dem. Par.	Modif. F1
DTC	0.9012	-0.1574	-0.1916	-0.1978	0.8488
LR	0.8995	-0.0209	-0.0246	-0.0238	0.9363
MLP	0.8964	-0.0389	-0.0456	-0.0469	0.9239
RFC	0.8958	-0.1360	-0.3506	-0.0981	0.8989
SVM	0.8834	0.0033	-0.02840	0.0006	**0.9378**

Table 6.26: Results of the Pre-processing strategy on BANK.

On the COMPAS dataset

Classifier	Accuracy	Corr. Deci.	Corr. Proba.	Dem. Par.	Modif. F1
DTC	0.6654	-0.1860	-0.2144	-0.1858	0.7323
LR	0.6629	-0.0486	-0.0033	-0.0487	**0.7814**
MLP	0.6683	-0.0662	-0.0754	-0.0667	0.7789
RFC	0.6766	-0.2241	-0.2987	-0.2256	0.7222
SVM	0.6633	-0.0548	-0.0053	-0.0550	0.7794

Table 6.27: Results of the Pre-processing strategy on COMPAS.

On the GERMAN dataset

Classifier	Accuracy	Corr. Deci.	Corr. Proba.	Dem. Par.	Modif. F1
DTC	0.7130	-0.1908	-0.1730	-0.2131	0.7481
LR	0.7390	0.0477	0.0169	0.0513	**0.8308**
MLP	0.7120	0.0078	0.0051	0.0053	0.8299
RFC	0.7220	-0.0816	-0.3158	-0.0494	0.8207
SVM	0.7000	0.0217	-0.0027	0.0025	0.8227

Table 6.28: Results of the Pre-processing strategy on GERMAN.

On the LAW dataset

Classifier	Accuracy	Corr. Deci.	Corr. Proba.	Dem. Par.	Modif. F1
DTC	0.9480	-0.1661	-0.2549	-0.0508	0.9486
LR	0.9500	-0.0181	-0.0488	-0.0034	**0.9727**
MLP	0.9507	-0.0981	-0.0931	-0.0180	0.9661
RFC	0.9511	-0.1035	-0.3394	-0.0127	0.9688
SVM	0.9502	-0.0515	-0.0808	-0.0107	0.9694

Table 6.29: Results of the Pre-processing strategy on LAW.

F In-processing strategy

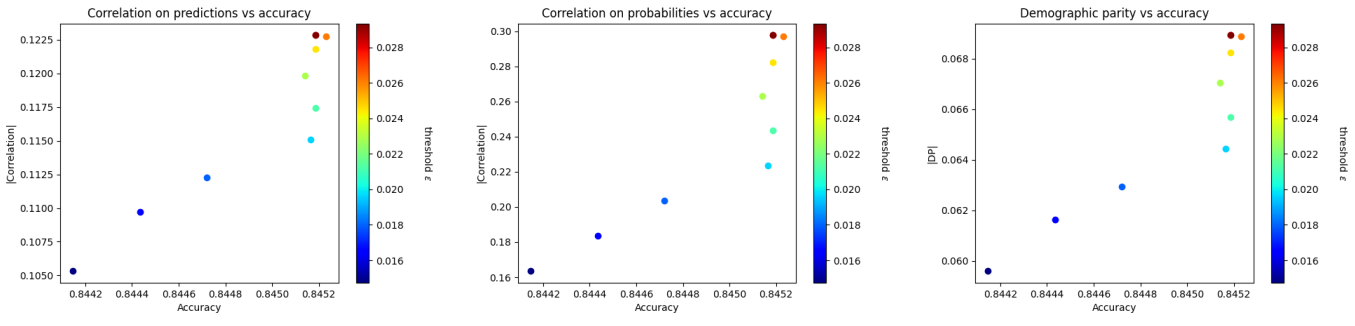
F.1 Results overview

Dataset	Accuracy	Corr. Deci.	Corr. Proba.	Dem. Par.	Modif. F1
ADULT	0.8441	-0.1053	-0.1635	-0.0596	**0.8897**
BANK	0.8984	-0.0192	-0.0233	-0.0234	**0.9359**
COMPAS	0.6636	-0.0298	-0.0001	-0.0300	**0.7880**
GERMAN	0.7470	0.0077	0.0069	0.0085	**0.8521**
LAW	0.9465	-0.0203	-0.0172	-0.0071	**0.9692**

Table 6.30: Results of the In-processing strategy on all datasets. The In-processing strategy uses a Binary Maximum Entropy classifier.

F.2 Graphics overview

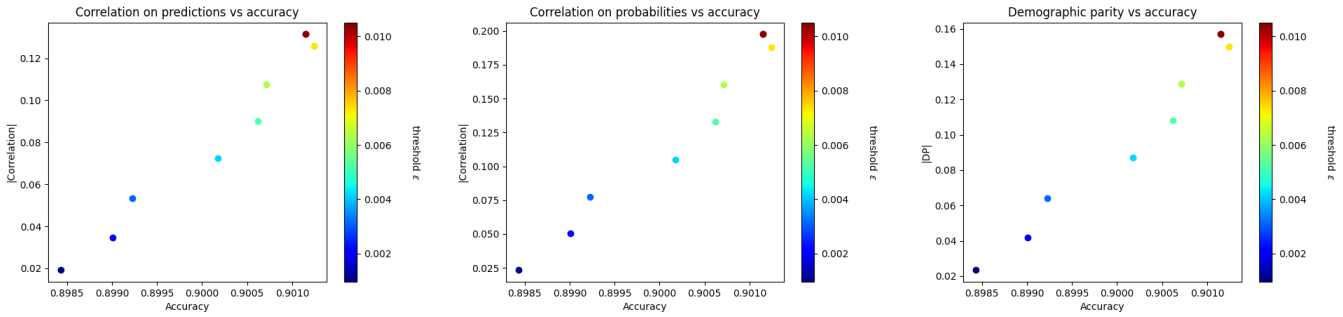
On the ADULT dataset



(a) (b) (c)

Figure 6.1: Epsilon variation - In-processing strategy on ADULT dataset: comparison between accuracy and (a) correlation on the decisions (b) correlation on the predicted probabilities (c) demographic parity.

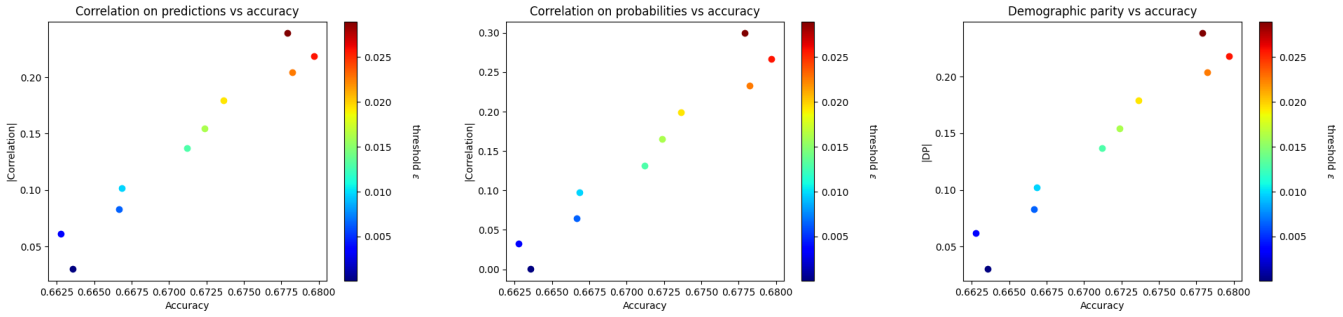
On the BANK dataset



(a) (b) (c)

Figure 6.2: Epsilon variation - In-processing strategy on BANK dataset: comparison between accuracy and (a) correlation on the decisions (b) correlation on the predicted probabilities (c) demographic parity.

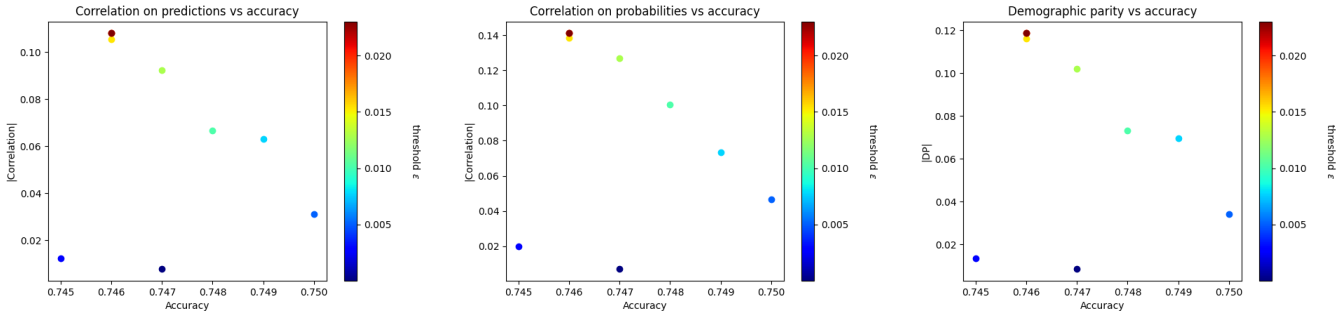
On the COMPAS dataset



(a) (b) (c)

Figure 6.3: Epsilon variation - In-processing strategy on COMPAS dataset: comparison between accuracy and (a) correlation on the decisions (b) correlation on the predicted probabilities (c) demographic parity.

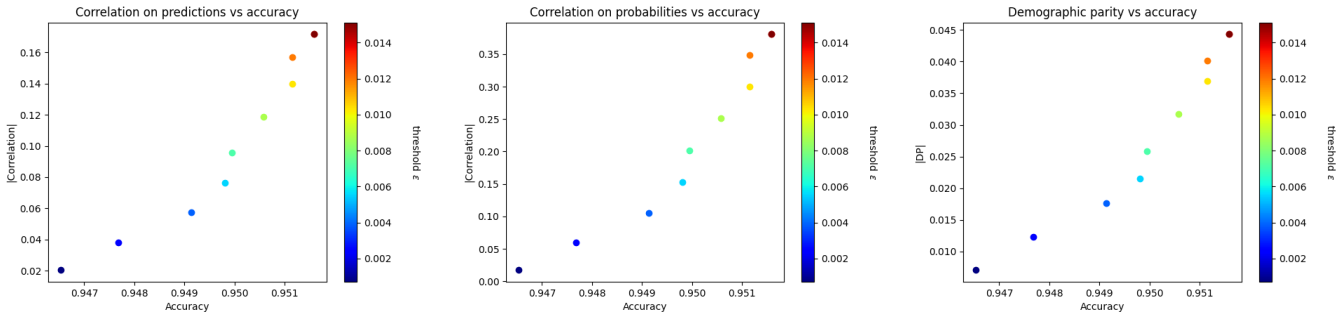
On the GERMAN dataset



(a) (b) (c)

Figure 6.4: Epsilon variation - In-processing strategy on GERMAN dataset: comparison between accuracy and (a) correlation on the decisions (b) correlation on the predicted probabilities (c) demographic parity.

On the LAW dataset



(a) (b) (c)

Figure 6.5: Epsilon variation - In-processing strategy on LAW dataset: comparison between accuracy and (a) correlation on the decisions (b) correlation on the predicted probabilities (c) demographic parity.

G Post-processing strategy

G.1 Norm L2

G.1.1 Results overview

On the ADULT dataset

Method	Classifier	Accuracy	Corr. Deci.	Corr. Proba.	Dem. Par.	Modif. F1
PP_L2	DTC	0.8524	-0.0918	-0.1651	-0.0526	0.8974
PP_L2	LR	0.8415	-0.0909	-0.1811	-0.0461	0.8942
PP_L2	MLP	0.8391	-0.0705	-0.1772	-0.0363	0.8971
PP_L2	RFC	0.8457	-0.0702	-0.2198	-0.0273	0.9047
PP_L2	SVM	0.8377	-0.0450	-0.2388	-0.0119	**0.9067**

Table 6.31: Results of the Post-processing strategy with L2-norm on ADULT.

On the BANK dataset

Method	Classifier	Accuracy	Corr. Deci.	Corr. Proba.	Dem. Par.	Modif. F1
PP_L2	DTC	0.8999	-0.0395	-0.0250	-0.0483	0.9251
PP_L2	LR	0.9002	-0.0562	-0.0256	-0.0648	0.9174
PP_L2	MLP	0.8990	0.0310	-0.0254	0.0410	0.9280
PP_L2	RFC	0.8950	-0.0759	-0.0386	-0.0532	0.9201
PP_L2	SVM	0.8834	-0.0198	-0.0564	-0.0074	**0.9348**

Table 6.32: Results of the Post-processing strategy with L2-norm on BANK.

On the COMPAS dataset

Method	Classifier	Accuracy	Corr. Deci.	Corr. Proba.	Dem. Par.	Modif. F1
PP_L2	DTC	0.6633	-0.0483	-0.0007	-0.0481	**0.7818**
PP_L2	LR	0.6654	-0.0724	-0.0006	-0.0725	0.7749
PP_L2	MLP	0.6761	-0.0913	-0.0006	-0.0918	0.7752
PP_L2	RFC	0.6735	-0.0791	-0.0007	-0.0793	0.7779
PP_L2	SVM	0.6652	-0.0823	-0.0006	-0.0826	0.7712

Table 6.33: Results of the Post-processing strategy with L2-norm on COMPAS.

On the GERMAN dataset

Method	Classifier	Accuracy	Corr. Deci.	Corr. Proba.	Dem. Par.	Modif. F1
PP_L2	DTC	0.7280	-0.0149	-0.0014	-0.0159	0.8369
PP_L2	LR	0.7450	-0.0141	-0.0001	-0.0153	**0.8483**
PP_L2	MLP	0.7110	0.0278	0.0068	0.0177	0.8249
PP_L2	RFC	0.7160	0.0958	0.0008	0.0561	0.8143
PP_L2	SVM	0.7030	0.0343	0.0023	0.0062	0.8235

Table 6.34: Results of the Post-processing strategy with L2-norm on GERMAN.

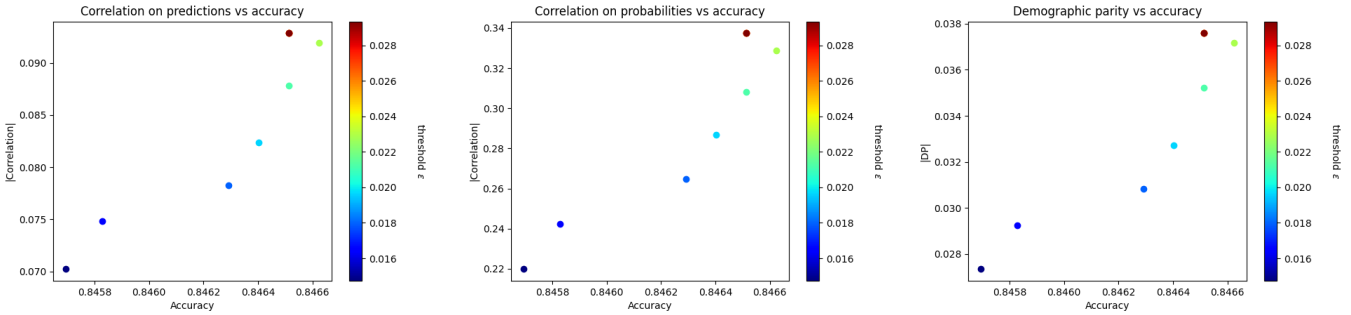
On the LAW dataset

Method	Classifier	Accuracy	Corr. Deci.	Corr. Proba.	Dem. Par.	Modif. F1
PP_L2	DTC	0.9487	-0.1317	-0.0210	-0.0395	0.9546
PP_L2	LR	0.9511	-0.0780	-0.0249	-0.0142	0.9681
PP_L2	MLP	0.9500	0.0143	-0.0264	0.0016	**0.9736**
PP_L2	RFC	0.9509	-0.0566	-0.0274	-0.0069	0.9715
PP_L2	SVM	0.9508	-0.1128	-0.0249	-0.0240	0.9632

Table 6.35: Results of the Post-processing strategy with L2-norm on LAW.

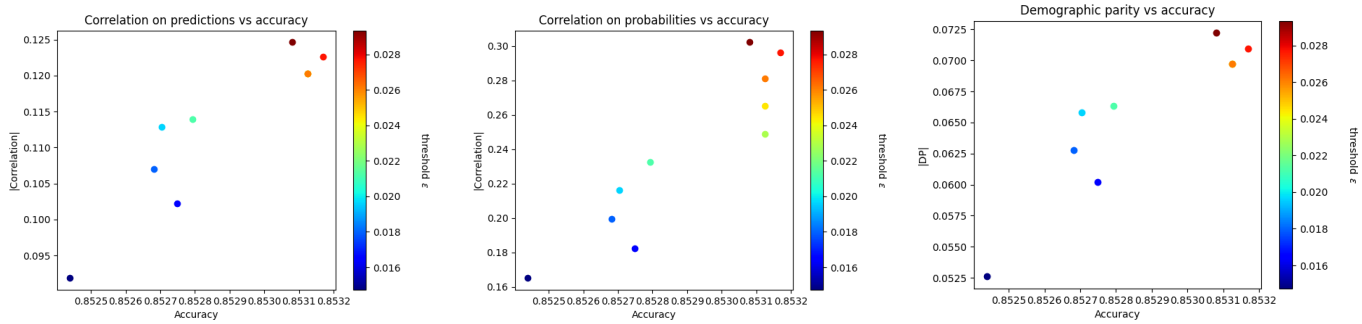
G.1.2 Graphics overview: epsilon variation

On the ADULT dataset



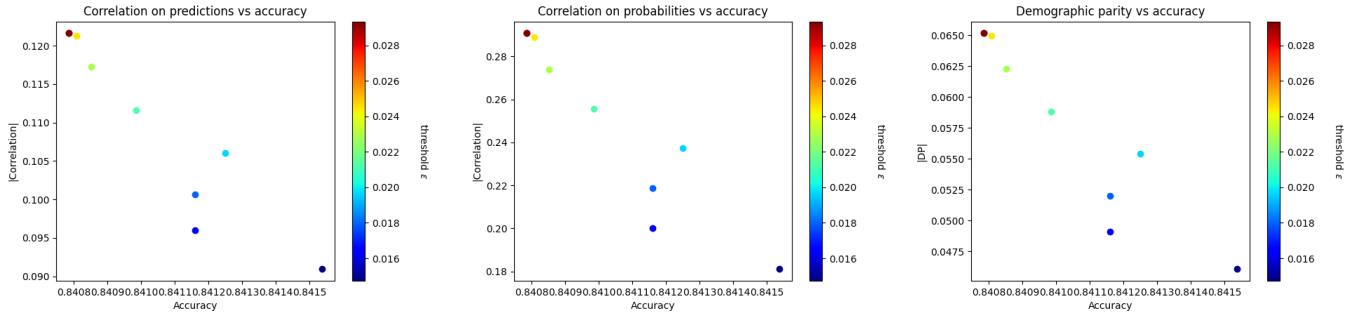
(a) (b) (c)

Figure 6.6: Epsilon variation - Post-processing strategy with L2-norm with a Random Forest Classifier on ADULT dataset: comparison between accuracy and (a) correlation on the decisions (b) correlation on the predicted probabilities (c) demographic parity.



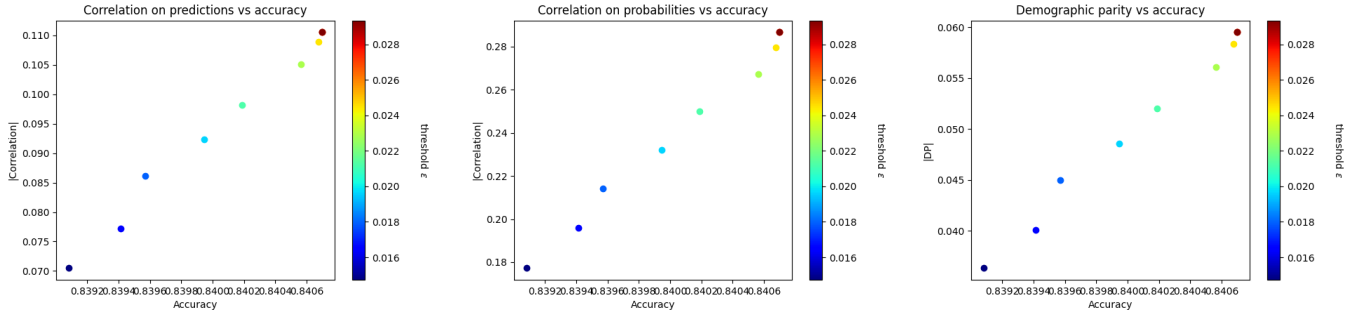
(a) (b) (c)

Figure 6.7: Epsilon variation - Post-processing strategy with L2-norm with a Decision Tree Classifier on ADULT dataset: comparison between accuracy and (a) correlation on the decisions (b) correlation on the predicted probabilities (c) demographic parity.



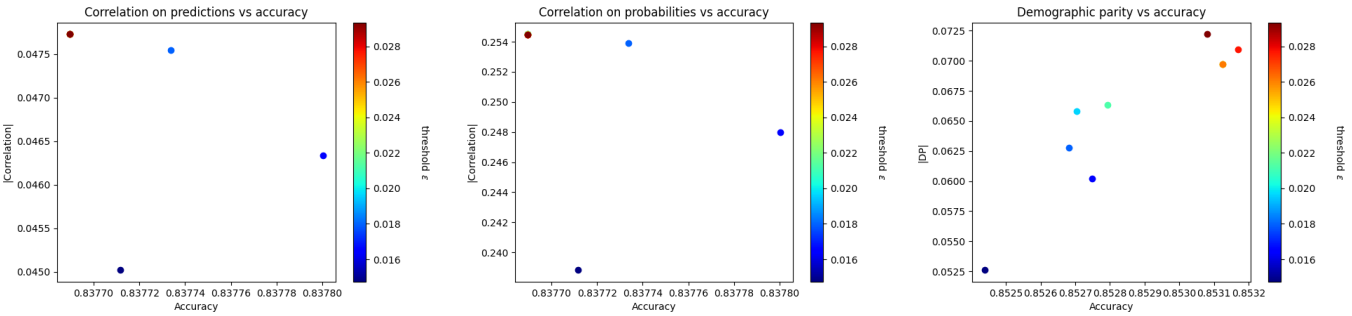
(a) (b) (c)

Figure 6.8: Epsilon variation - Post-processing strategy with L2-norm with a Logistic Regression on ADULT dataset: comparison between accuracy and (a) correlation on the decisions (b) correlation on the predicted probabilities (c) demographic parity.



(a) (b) (c)

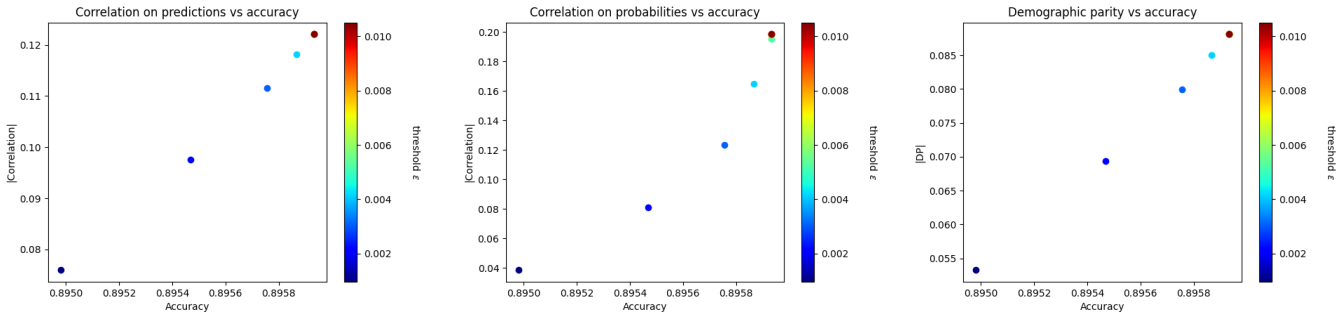
Figure 6.9: Epsilon variation - Post-processing strategy with L2-norm with a Multi Layer Perceptron on ADULT dataset: comparison between accuracy and (a) correlation on the decisions (b) correlation on the predicted probabilities (c) demographic parity.



(a) (b) (c)

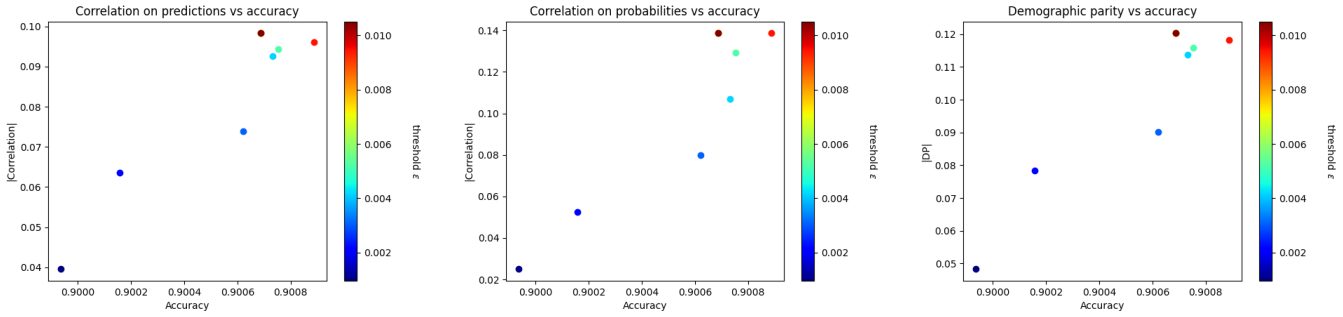
Figure 6.10: Epsilon variation - Post-processing strategy with L2-norm with a Support Vector Machine on ADULT dataset: comparison between accuracy and (a) correlation on the decisions (b) correlation on the predicted probabilities (c) demographic parity.

On the BANK dataset



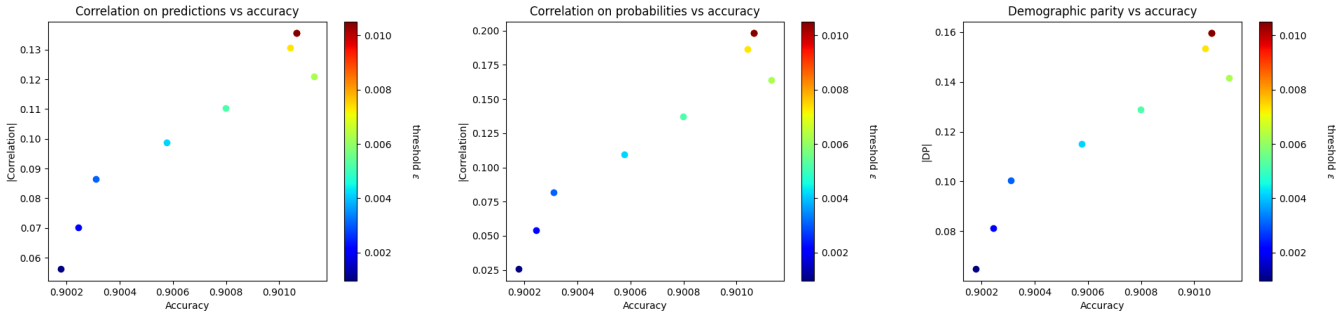
(a) (b) (c)

Figure 6.11: Epsilon variation - Post-processing strategy with L2-norm with a Random Forest Classifier on BANK dataset: comparison between accuracy and (a) correlation on the decisions (b) correlation on the predicted probabilities (c) demographic parity.



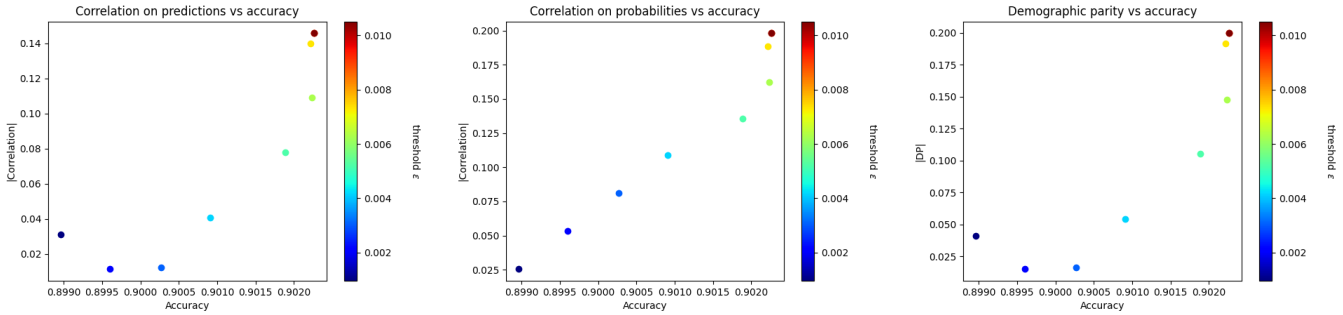
(a) (b) (c)

Figure 6.12: Epsilon variation - Post-processing strategy with L2-norm with a Decision Tree Classifier on BANK dataset: comparison between accuracy and (a) correlation on the decisions (b) correlation on the predicted probabilities (c) demographic parity.



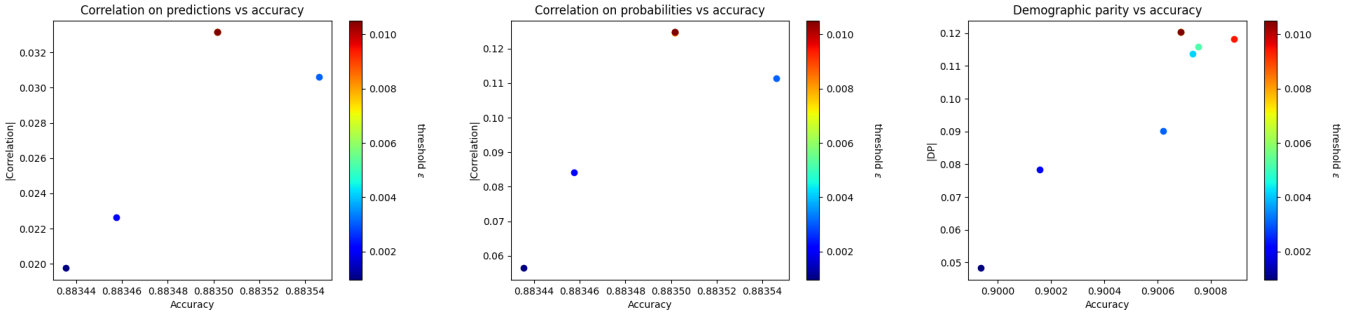
(a) (b) (c)

Figure 6.13: Epsilon variation - Post-processing strategy with L2-norm with a Logistic Regression on BANK dataset: comparison between accuracy and (a) correlation on the decisions (b) correlation on the predicted probabilities (c) demographic parity.



(a) (b) (c)

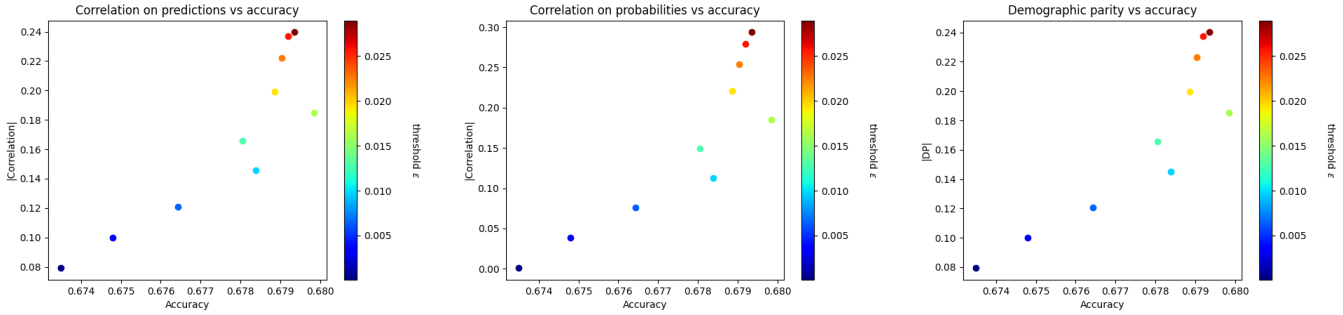
Figure 6.14: Epsilon variation - Post-processing strategy with L2-norm with a Multi Layer Perceptron on BANK dataset: comparison between accuracy and (a) correlation on the decisions (b) correlation on the predicted probabilities (c) demographic parity.



(a) (b) (c)

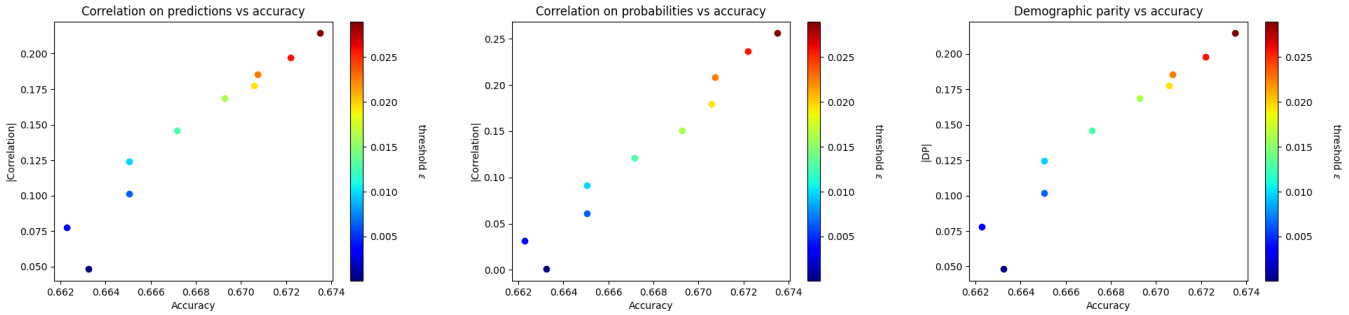
Figure 6.15: Epsilon variation - Post-processing strategy with L2-norm with a Support Vector Machine on BANK dataset: comparison between accuracy and (a) correlation on the decisions (b) correlation on the predicted probabilities (c) demographic parity.

On the COMPAS dataset



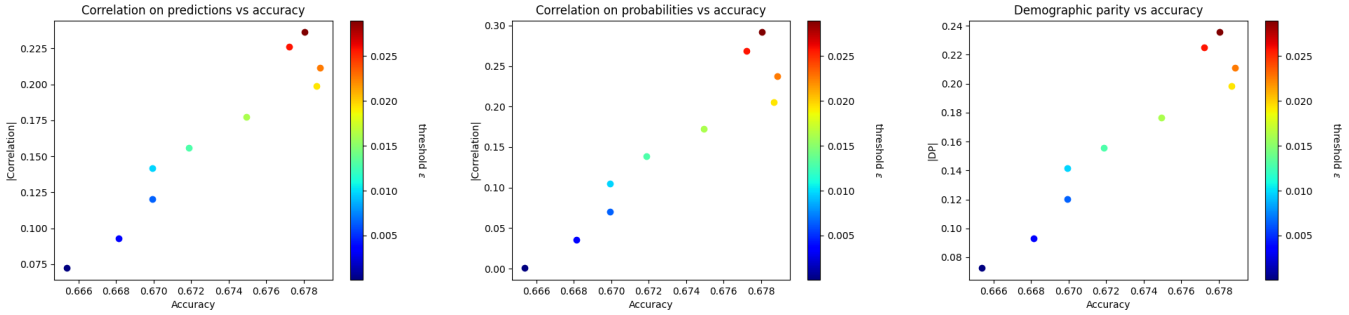
(a) (b) (c)

Figure 6.16: Epsilon variation - Post-processing strategy with L2-norm with a Random Forest Classifier on COMPAS dataset: comparison between accuracy and (a) correlation on the decisions (b) correlation on the predicted probabilities.



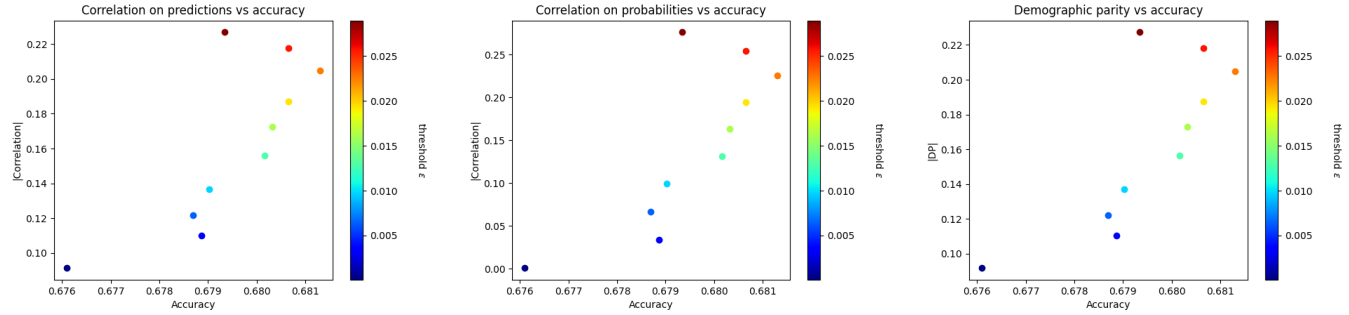
(a) (b) (c)

Figure 6.17: Epsilon variation - Post-processing strategy with L2-norm with a Decision Tree Classifier on COMPAS dataset: comparison between accuracy and (a) correlation on the decisions (b) correlation on the predicted probabilities.



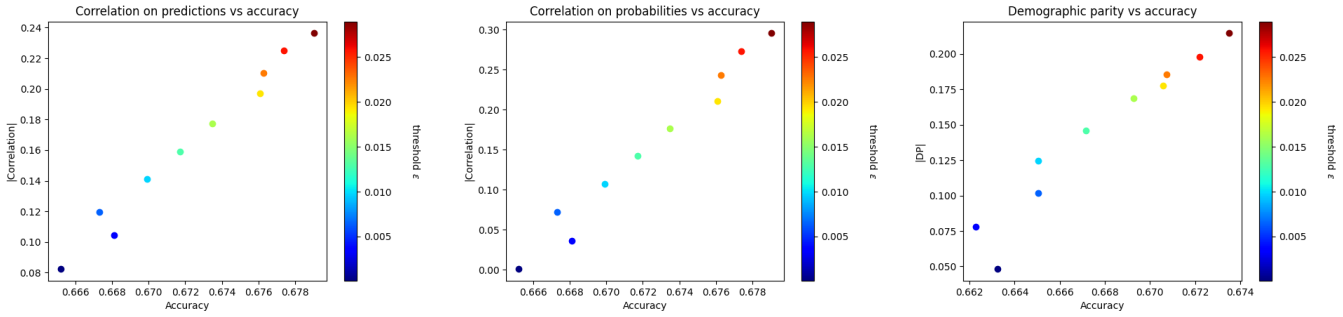
(a) (b) (c)

Figure 6.18: Epsilon variation - Post-processing strategy with L2-norm with a Logistic Regression on COMPAS dataset: comparison between accuracy and (a) correlation on the decisions (b) correlation on the predicted probabilities (c) demographic parity.



(a) (b) (c)

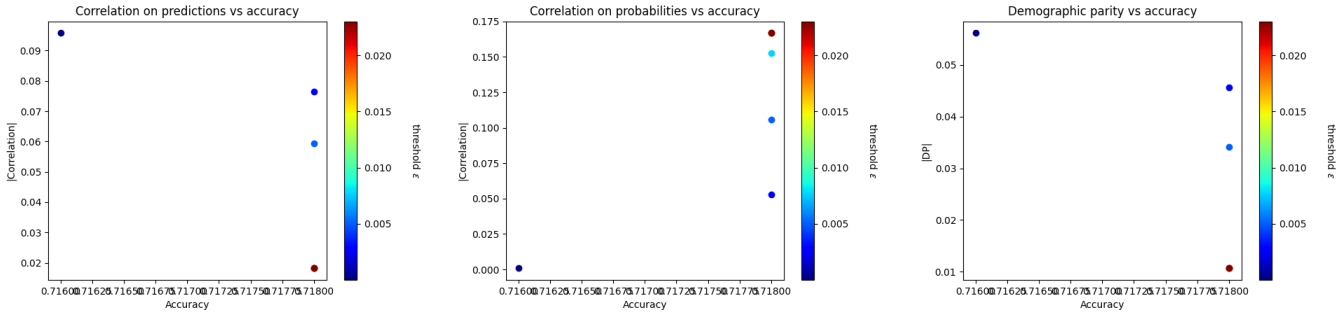
Figure 6.19: Epsilon variation - Post-processing strategy with L2-norm with a Multi Layer Perceptron on COMPAS dataset: comparison between accuracy and (a) correlation on the decisions (b) correlation on the predicted probabilities (c) demographic parity.



(a) (b) (c)

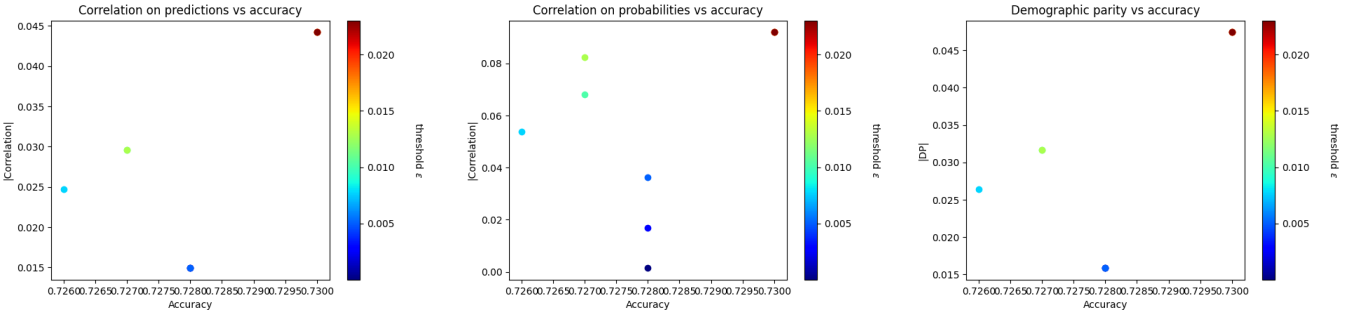
Figure 6.20: Epsilon variation - Post-processing strategy with L2-norm with a Support Vector Machine on COMPAS dataset: comparison between accuracy and (a) correlation on the decisions (b) correlation on the predicted probabilities (c) demographic parity.

On the GERMAN dataset



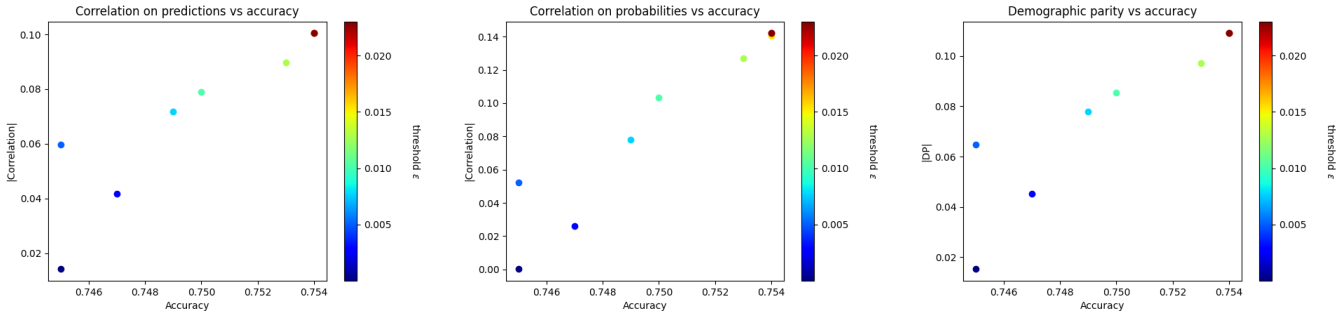
(a) (b) (c)

Figure 6.21: Epsilon variation - Post-processing strategy with L2-norm with a Random Forest Classifier on GERMAN dataset: comparison between accuracy and (a) correlation on the decisions (b) correlation on the predicted probabilities (c) demographic parity.



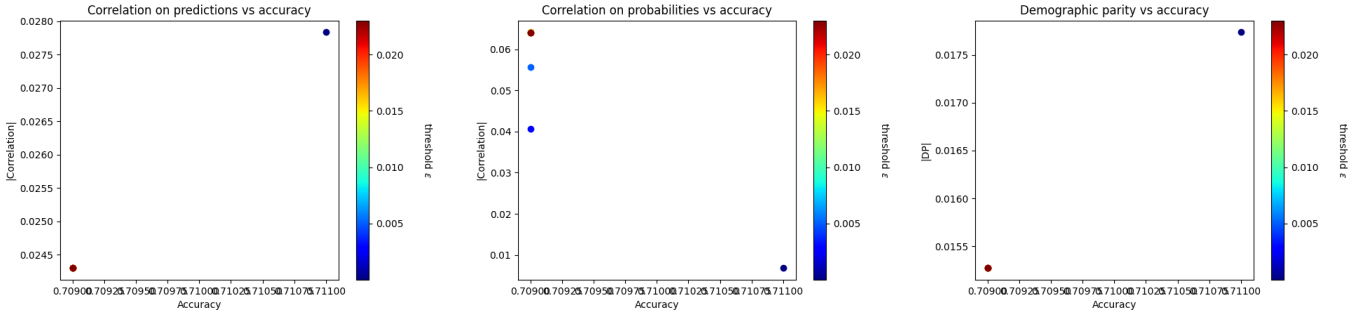
(a) (b) (c)

Figure 6.22: Epsilon variation - Post-processing strategy with L2-norm with a Decision Tree Classifier on GERMAN dataset: comparison between accuracy and (a) correlation on the decisions (b) correlation on the predicted probabilities (c) demographic parity.



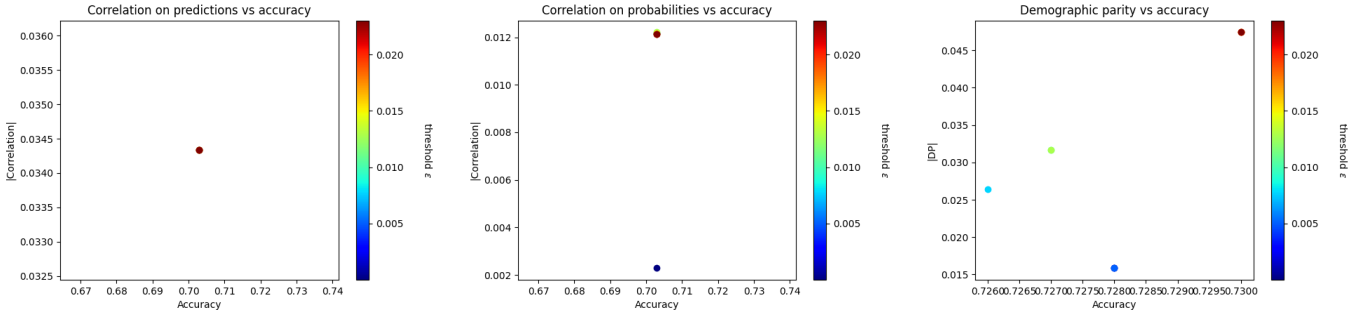
(a) (b) (c)

Figure 6.23: Epsilon variation - Post-processing strategy with L2-norm with a Logistic Regression on GERMAN dataset: comparison between accuracy and (a) correlation on the decisions (b) correlation on the predicted probabilities (c) demographic parity.



(a) (b) (c)

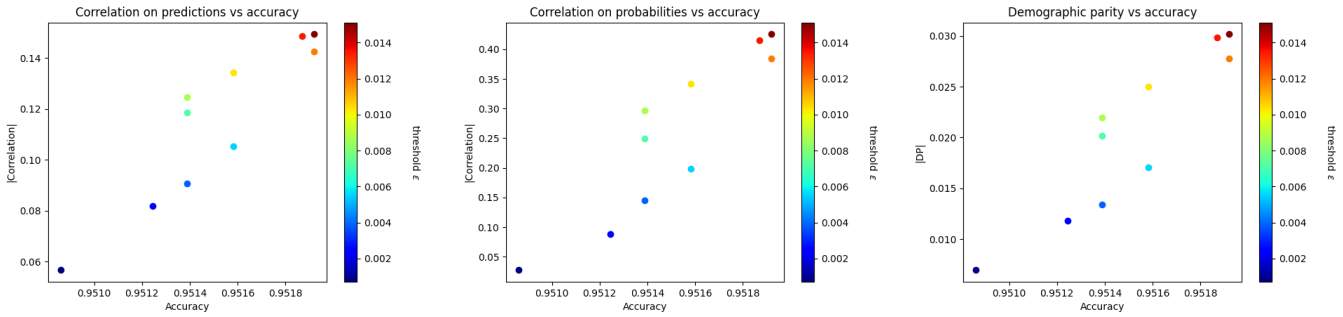
Figure 6.24: Epsilon variation - Post-processing strategy with L2-norm with a Multi Layer Perceptron on GERMAN dataset: comparison between accuracy and (a) correlation on the decisions (b) correlation on the predicted probabilities (c) demographic parity.



(a) (b) (c)

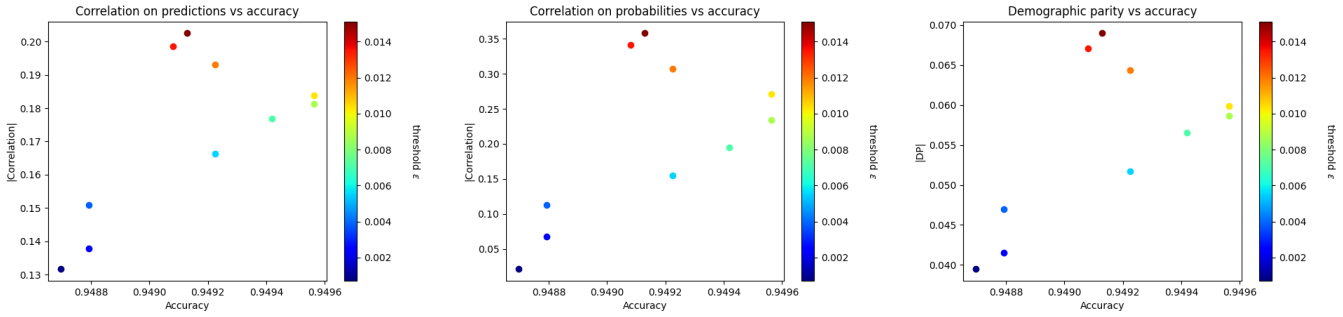
Figure 6.25: Epsilon variation - Post-processing strategy with L2-norm with a Support Vector Machine on GERMAN dataset: comparison between accuracy and (a) correlation on the decisions (b) correlation on the predicted probabilities (c) demographic parity.

On the LAW dataset



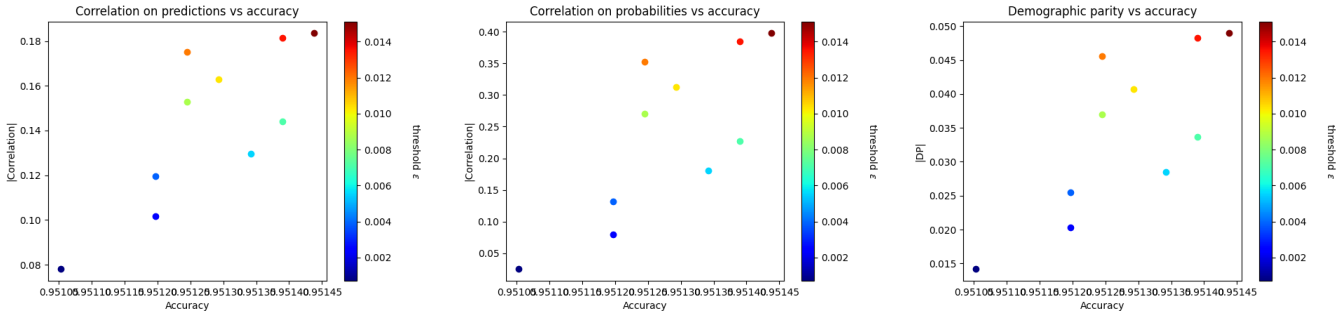
(a) (b) (c)

Figure 6.26: Epsilon variation - Post-processing strategy with L2-norm with a Random Forest Classifier on LAW dataset: comparison between accuracy and (a) correlation on the decisions (b) correlation on the predicted probabilities (c) demographic parity.



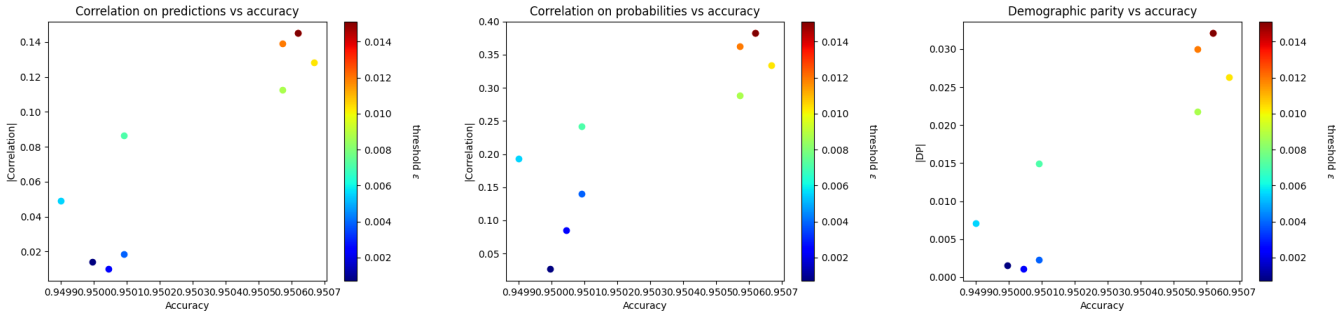
(a) (b) (c)

Figure 6.27: Epsilon variation - Post-processing strategy with L2-norm with a Decision Tree Classifier on LAW dataset: comparison between accuracy and (a) correlation on the decisions (b) correlation on the predicted probabilities (c) demographic parity.



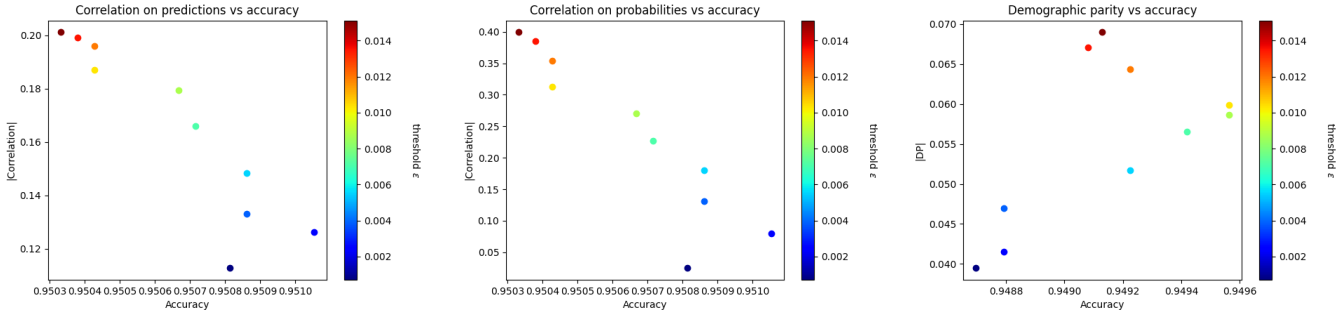
(a) (b) (c)

Figure 6.28: Epsilon variation - Post-processing strategy with L2-norm with a Logistic Regression on LAW dataset: comparison between accuracy and (a) correlation on the decisions (b) correlation on the predicted probabilities (c) demographic parity.



(a) (b) (c)

Figure 6.29: Epsilon variation - Post-processing strategy with L2-norm with a Multi Layer Perceptron on LAW dataset: comparison between accuracy and (a) correlation on the decisions (b) correlation on the predicted probabilities (c) demographic parity.



(a) (b) (c)

Figure 6.30: Epsilon variation - Post-processing strategy with L2-norm with a Support Vector Machine on LAW dataset: comparison between accuracy and (a) correlation on the decisions (b) correlation on the predicted probabilities (c) demographic parity.

G.2 Norm L1L2

G.2.1 Results overview

On the ADULT dataset

Method	Classifier	Accuracy	Corr. Deci.	Corr. Proba.	Dem. Par.	Modif. F1
PP_L1L2	DTC	0.8524	-0.0998	-0.1630	-0.0590	0.8945
PP_L1L2	LR	0.8413	-0.1049	-0.1805	-0.0572	0.8892
PP_L1L2	MLP	0.8406	-0.0842	-0.1761	-0.0466	0.8935
PP_L1L2	RFC	0.8472	-0.0751	-0.2192	-0.0312	0.9039
PP_L1L2	SVM	0.8376	-0.0449	-0.2386	-0.0120	**0.9066**

Table 6.36: Results of the Post-processing strategy on ADULT. L1L2-norm is used with the gamma parameter set to 0.5.

On the BANK dataset

Method	Classifier	Accuracy	Corr. Deci.	Corr. Proba.	Dem. Par.	Modif. F1
PP_L1L2	DTC	0.8999	-0.0370	-0.0250	-0.0442	0.9270
PP_L1L2	LR	0.8998	-0.0518	-0.0257	-0.0588	0.9200
PP_L1L2	MLP	0.8996	0.0371	-0.0255	0.0478	0.9252
PP_L1L2	RFC	0.8948	-0.0707	-0.0386	-0.0490	0.9220
PP_L1L2	SVM	0.8835	-0.0203	-0.0567	-0.0075	**0.9348**

Table 6.37: Results of the Post-processing strategy on BANK. L1L2-norm is used with the gamma parameter set to 0.5.

On the COMPAS dataset

Method	Classifier	Accuracy	Corr. Deci.	Corr. Proba.	Dem. Par.	Modif. F1
PP_L1L2	DTC	0.6624	-0.0679	-0.0007	-0.0691	0.7741
PP_L1L2	LR	0.6680	-0.0484	-0.0005	-0.0494	0.7846
PP_L1L2	MLP	0.6689	-0.0502	-0.0004	-0.0512	**0.7847**
PP_L1L2	RFC	0.6644	-0.0421	-0.0006	-0.0429	0.7843
PP_L1L2	SVM	0.6675	-0.0544	-0.0005	-0.0555	0.7822

Table 6.38: Results of the Post-processing strategy on COMPAS. L1L2-norm is used with the gamma parameter set to 0.5.

On the GERMAN dataset

Method	Classifier	Accuracy	Corr. Deci.	Corr. Proba.	Dem. Par.	Modif. F1
PP_L1L2	DTC	0.7280	-0.0149	-0.0008	-0.0159	0.8369
PP_L1L2	LR	0.7510	-0.0185	-0.0001	-0.0196	**0.8505**
PP_L1L2	MLP	0.7090	0.0243	0.0073	0.0153	0.8244
PP_L1L2	RFC	0.7150	0.0863	0.0010	0.0475	0.8168
PP_L1L2	SVM	0.7030	0.0343	0.0023	0.0062	0.8235

Table 6.39: Results of the Post-processing strategy on GERMAN. L1L2-norm is used with the gamma parameter set to 0.5.

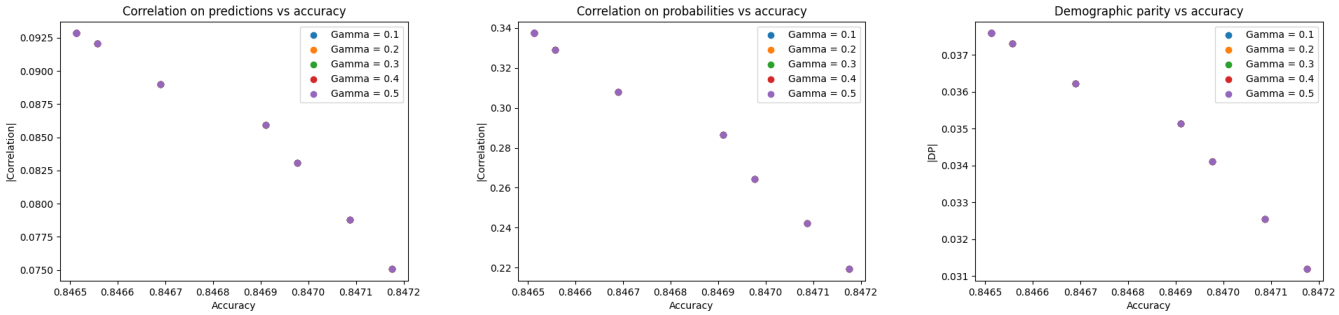
On the LAW dataset

Method	Classifier	Accuracy	Corr. Deci.	Corr. Proba.	Dem. Par.	Modif. F1
PP_L1L2	DTC	0.9493	-0.0693	-0.0238	-0.0163	0.9662
PP_L1L2	LR	0.9504	-0.0270	-0.0274	-0.0036	0.9728
PP_L1L2	MLP	0.9498	0.0154	-0.0287	0.0015	0.9735
PP_L1L2	RFC	0.9502	-0.0140	-0.0298	-0.0011	**0.9739**
PP_L1L2	SVM	0.9504	-0.0631	-0.0278	-0.0102	0.9697

Table 6.40: Results of the Post-processing strategy on LAW. L1L2-norm is used with the gamma parameter set to 0.5.

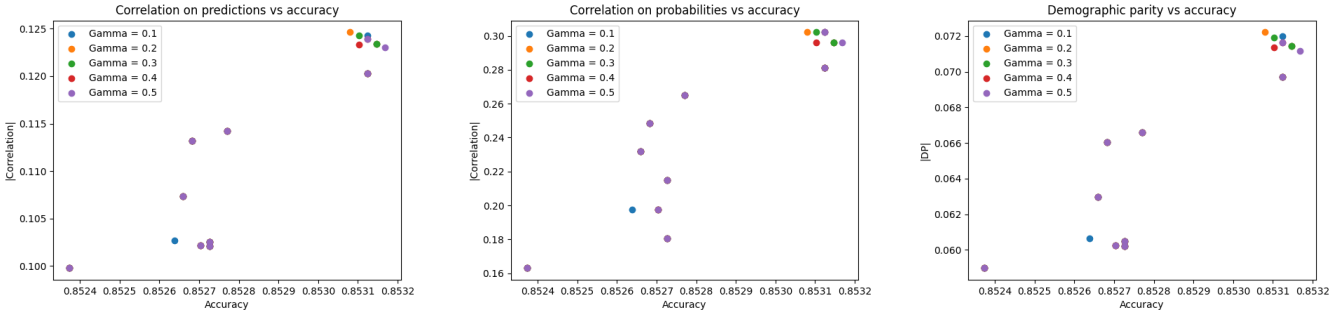
G.2.2 Graphics overview: gamma variation

On the ADULT dataset



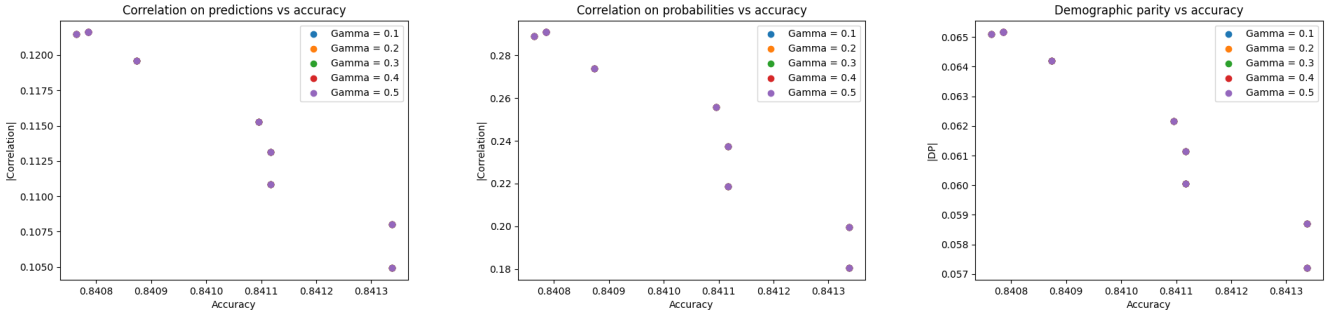
(a) (b) (c)

Figure 6.31: Gamma variation - Post-processing strategy with L1L2 with a Random Forest Classifier on ADULT dataset: comparison between accuracy and (a) correlation on the decisions (b) correlation on the predicted probabilities (c) demographic parity.



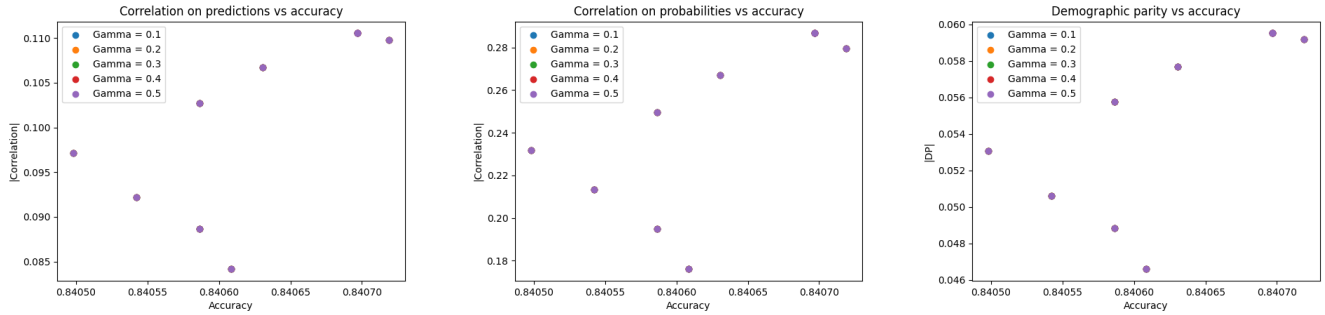
(a) (b) (c)

Figure 6.32: Gamma variation - Post-processing strategy with L1L2 with a Decision Tree Classifier on ADULT dataset: comparison between accuracy and (a) correlation on the decisions (b) correlation on the predicted probabilities (c) demographic parity.



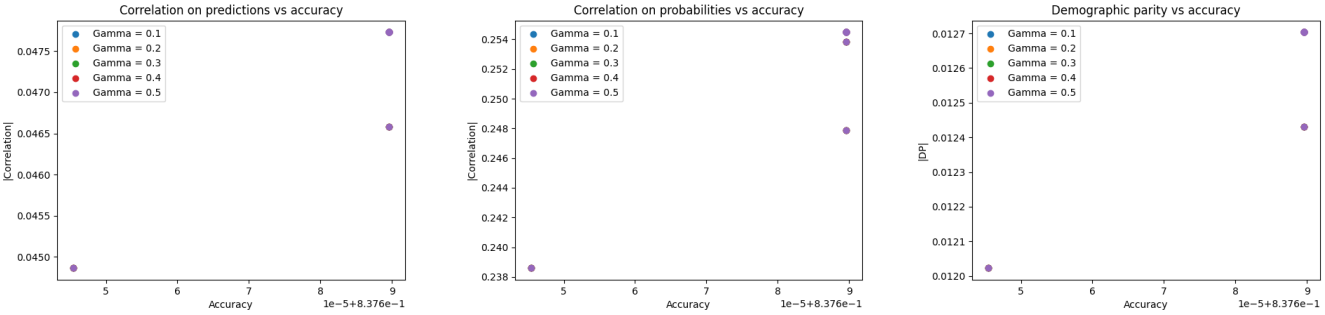
(a) (b) (c)

Figure 6.33: Gamma variation - Post-processing strategy with L1L2 with a Logistic Regression on ADULT dataset: comparison between accuracy and (a) correlation on the decisions (b) correlation on the predicted probabilities (c) demographic parity.



(a) (b) (c)

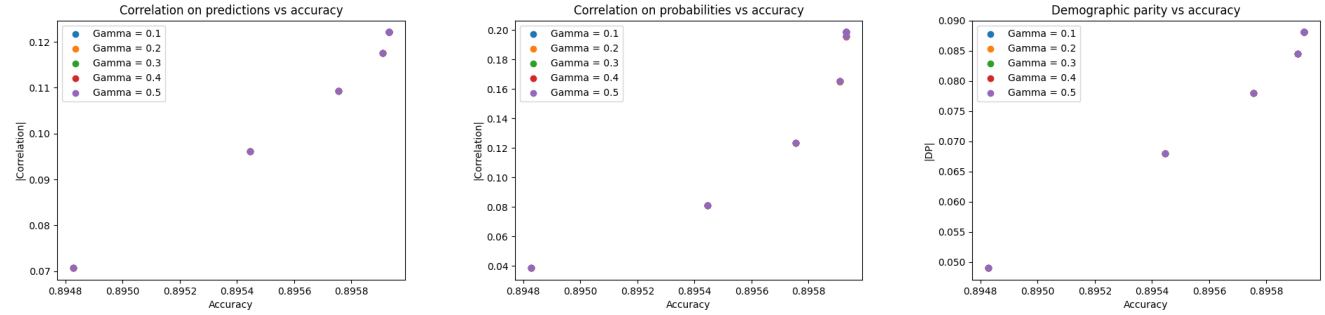
Figure 6.34: Gamma variation - Post-processing strategy with L1L2 with a Multi Layer Perceptron on ADULT dataset: comparison between accuracy and (a) correlation on the decisions (b) correlation on the predicted probabilities (c) demographic parity.



(a) (b) (c)

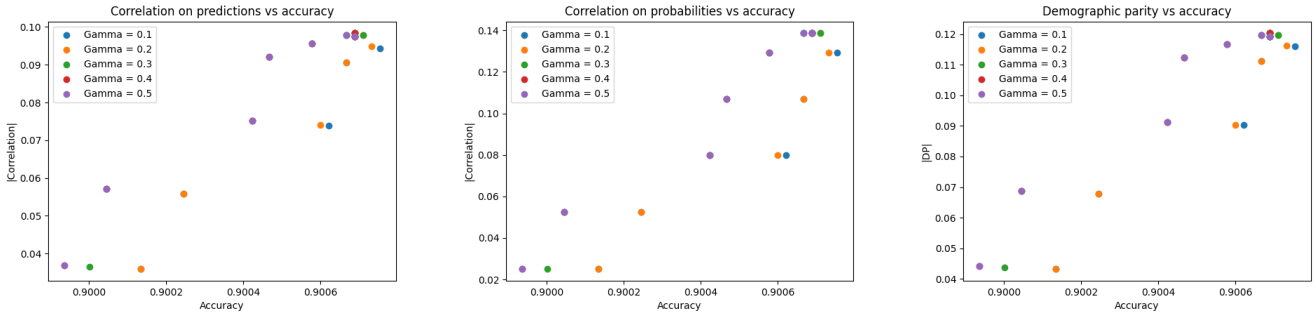
Figure 6.35: Gamma variation - Post-processing strategy with L1L2 with a Support Vector Machine on ADULT dataset: comparison between accuracy and (a) correlation on the decisions (b) correlation on the predicted probabilities (c) demographic parity.

On the BANK dataset



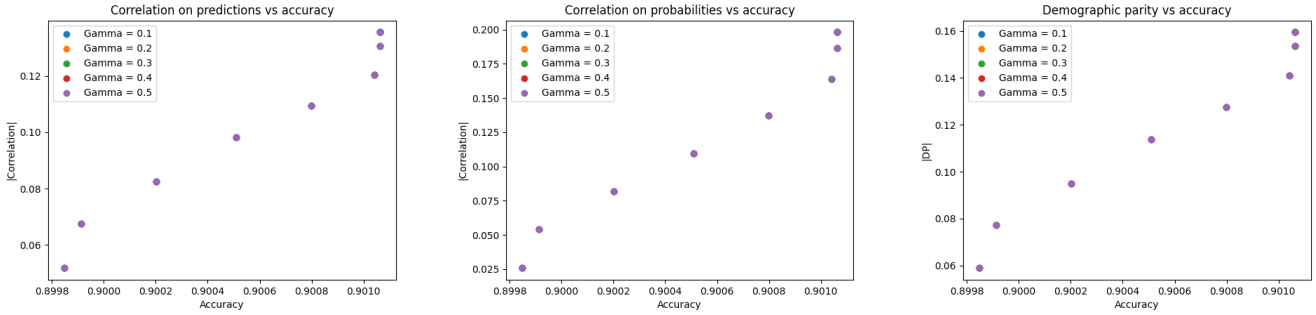
(a) (b) (c)

Figure 6.36: Gamma variation - Post-processing strategy with L1L2 with a Random Forest Classifier on BANK dataset: comparison between accuracy and (a) correlation on the decisions (b) correlation on the predicted probabilities (c) demographic parity.



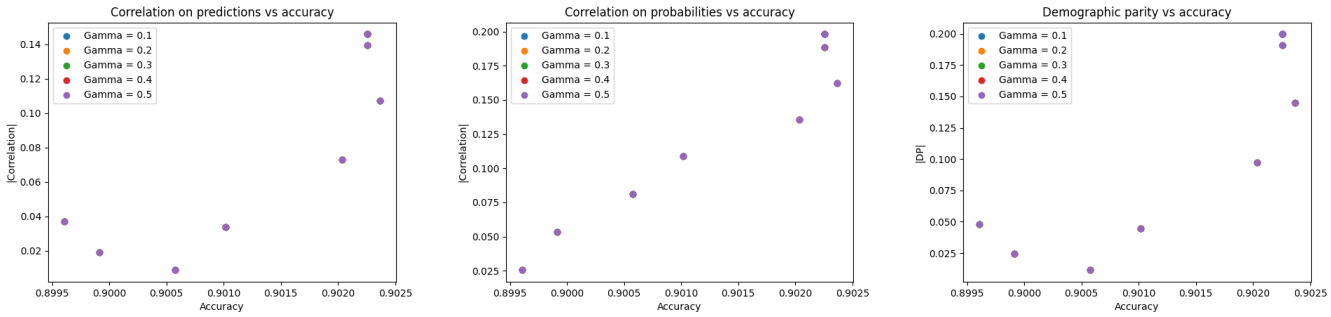
(a) (b) (c)

Figure 6.37: Gamma variation - Post-processing strategy with L1L2 with a Decision Tree Classifier on BANK dataset: comparison between accuracy and (a) correlation on the decisions (b) correlation on the predicted probabilities (c) demographic parity.



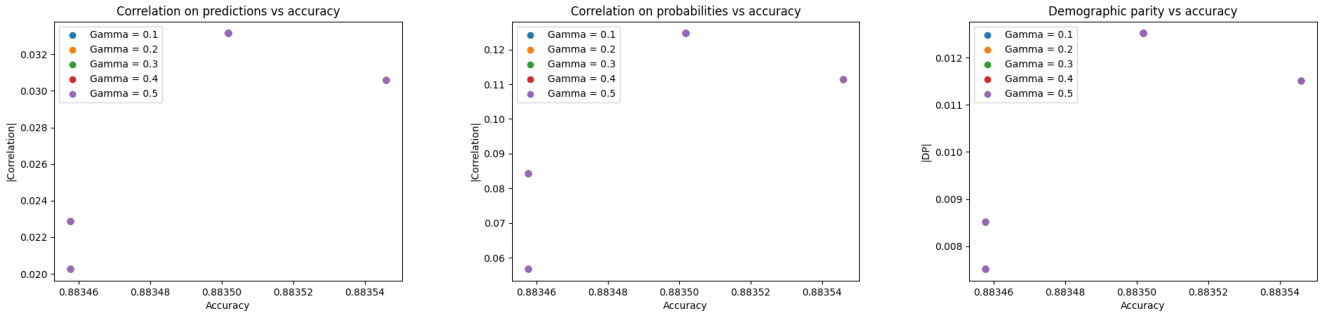
(a) (b) (c)

Figure 6.38: Gamma variation - Post-processing strategy with L1L2 with a Logistic Regression on BANK dataset: comparison between accuracy and (a) correlation on the decisions (b) correlation on the predicted probabilities (c) demographic parity.



(a) (b) (c)

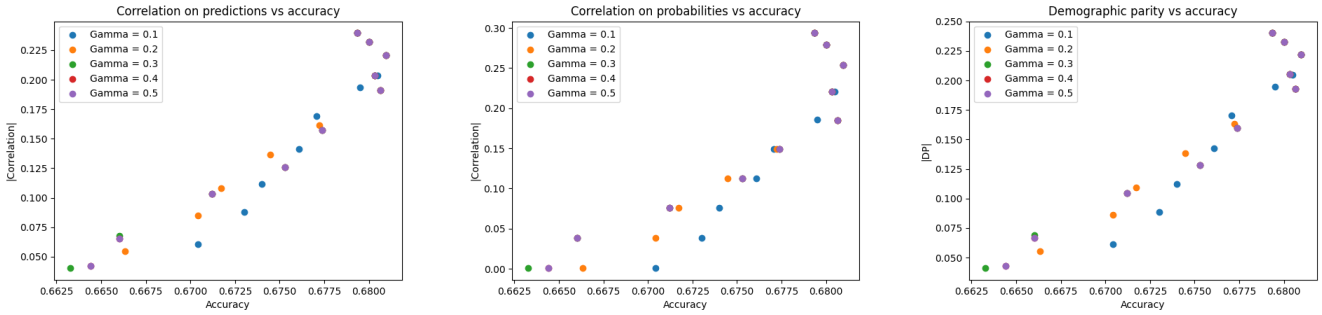
Figure 6.39: Gamma variation - Post-processing strategy with L1L2 with a Multi Layer Perceptron on BANK dataset: comparison between accuracy and (a) correlation on the decisions (b) correlation on the predicted probabilities (c) demographic parity.



(a) (b) (c)

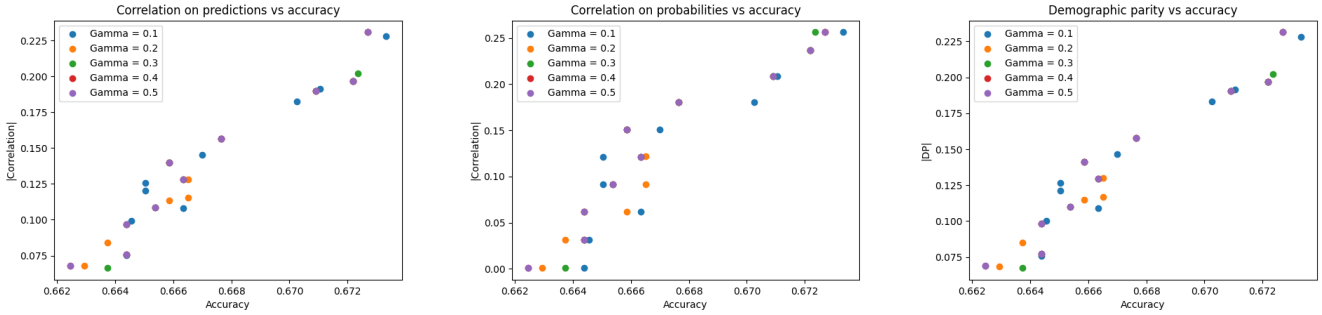
Figure 6.40: Gamma variation - Post-processing strategy with L1L2 with a Support Vector Machine on BANK dataset: comparison between accuracy and (a) correlation on the decisions (b) correlation on the predicted probabilities (c) demographic parity.

On the COMPAS dataset



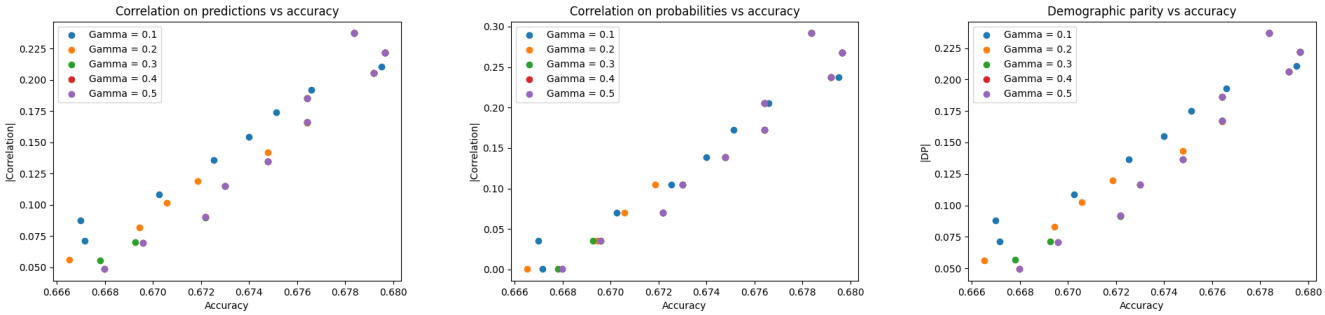
(a) (b) (c)

Figure 6.41: Gamma variation - Post-processing strategy with L1L2 with a Random Forest Classifier on COMPAS dataset: comparison between accuracy and (a) correlation on the decisions (b) correlation on the predicted probabilities (c) demographic parity.



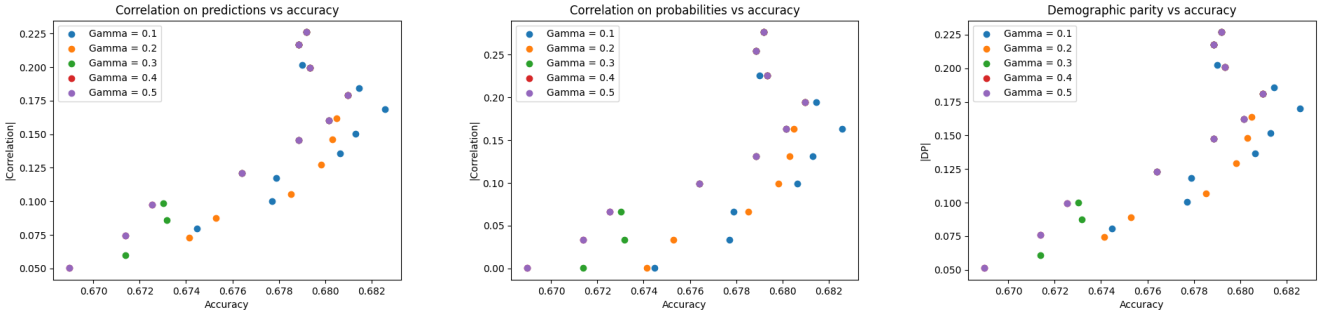
(a) (b) (c)

Figure 6.42: Gamma variation - Post-processing strategy with L1L2 with a Decision Tree Classifier on COMPAS dataset: comparison between accuracy and (a) correlation on the decisions (b) correlation on the predicted probabilities (c) demographic parity.



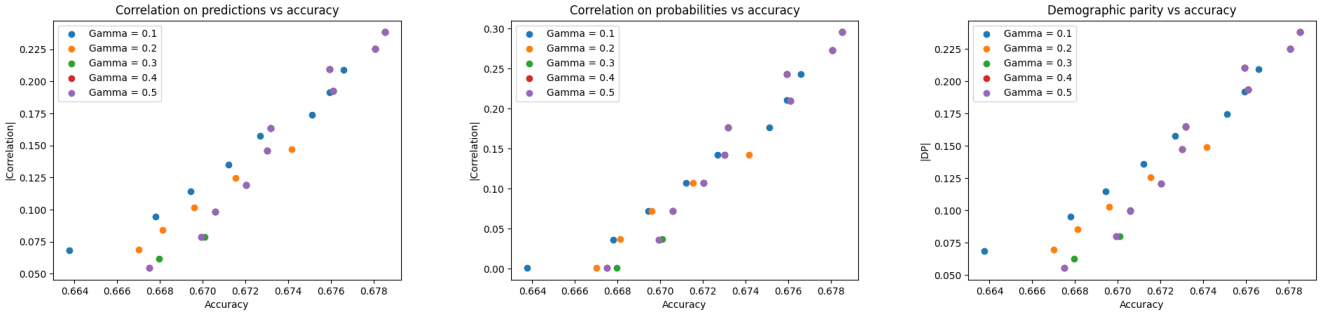
(a) (b) (c)

Figure 6.43: Gamma variation - Post-processing strategy with L1L2 with a Logistic Regression on COMPAS dataset: comparison between accuracy and (a) correlation on the decisions (b) correlation on the predicted probabilities (c) demographic parity.



(a) (b) (c)

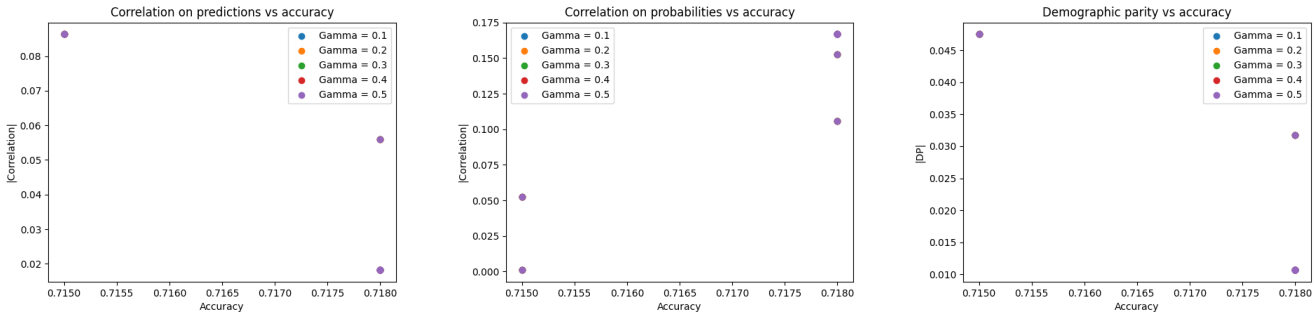
Figure 6.44: Gamma variation - Post-processing strategy with L1L2 with a Multi Layer Perceptron on COMPAS dataset: comparison between accuracy and (a) correlation on the decisions (b) correlation on the predicted probabilities (c) demographic parity.



(a) (b) (c)

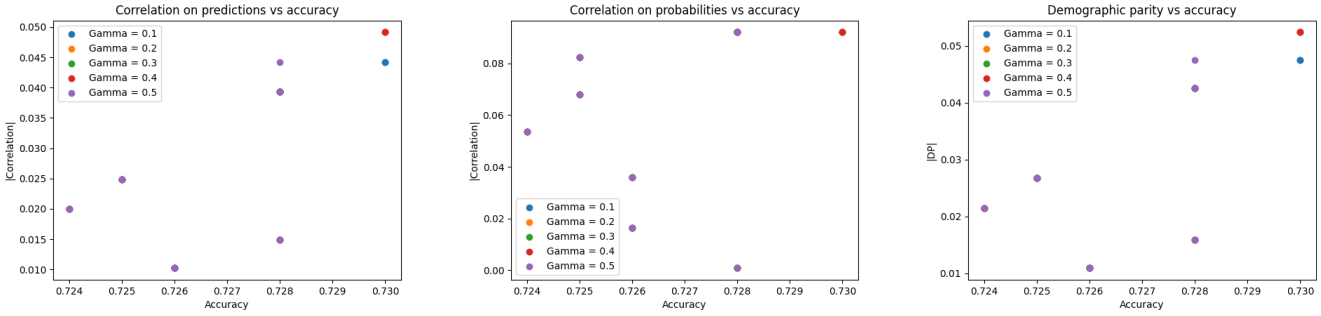
Figure 6.45: Gamma variation - Post-processing strategy with L1L2 with a Support Vector Machine on COMPAS dataset: comparison between accuracy and (a) correlation on the decisions (b) correlation on the predicted probabilities (c) demographic parity.

On the GERMAN dataset



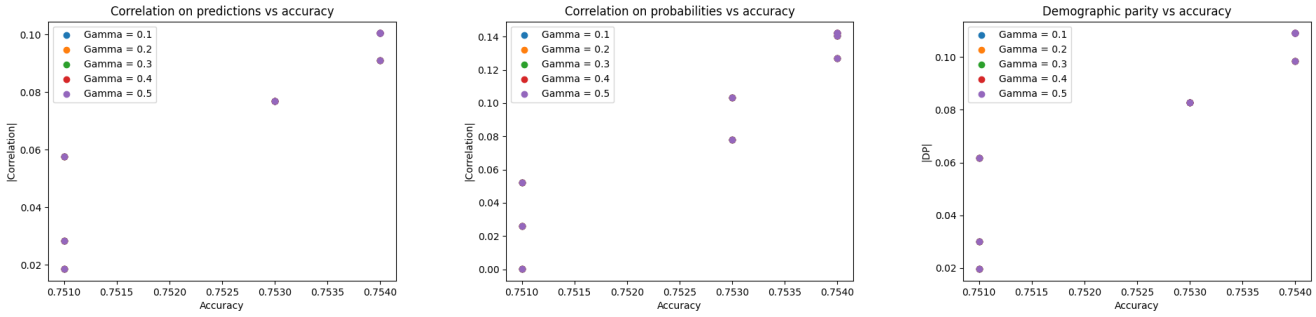
(a) (b) (c)

Figure 6.46: Gamma variation - Post-processing strategy with L1L2 with a Random Forest Classifier on GERMAN dataset: comparison between accuracy and (a) correlation on the decisions (b) correlation on the predicted probabilities (c) demographic parity.



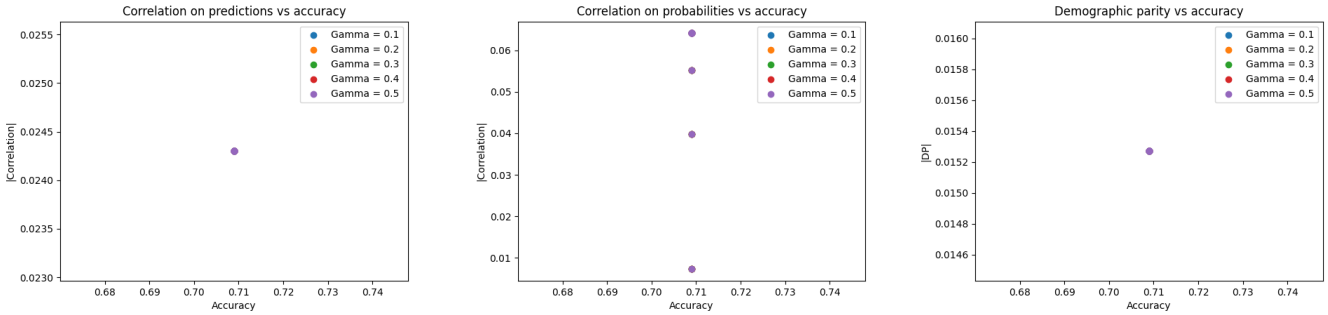
(a) (b) (c)

Figure 6.47: Gamma variation - Post-processing strategy with L1L2 with a Decision Tree Classifier on GERMAN dataset: comparison between accuracy and (a) correlation on the decisions (b) correlation on the predicted probabilities (c) demographic parity.



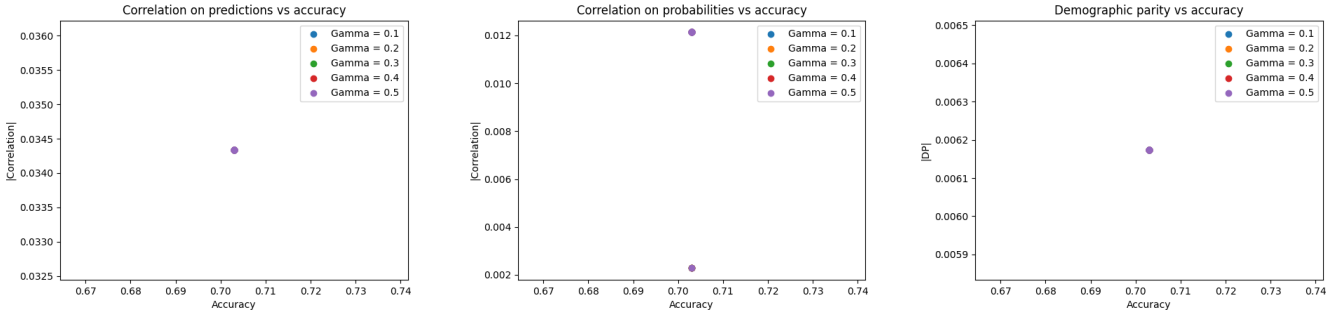
(a) (b) (c)

Figure 6.48: Gamma variation - Post-processing strategy with L1L2 with a Logistic Regression on GERMAN dataset: comparison between accuracy and (a) correlation on the decisions (b) correlation on the predicted probabilities (c) demographic parity.



(a) (b) (c)

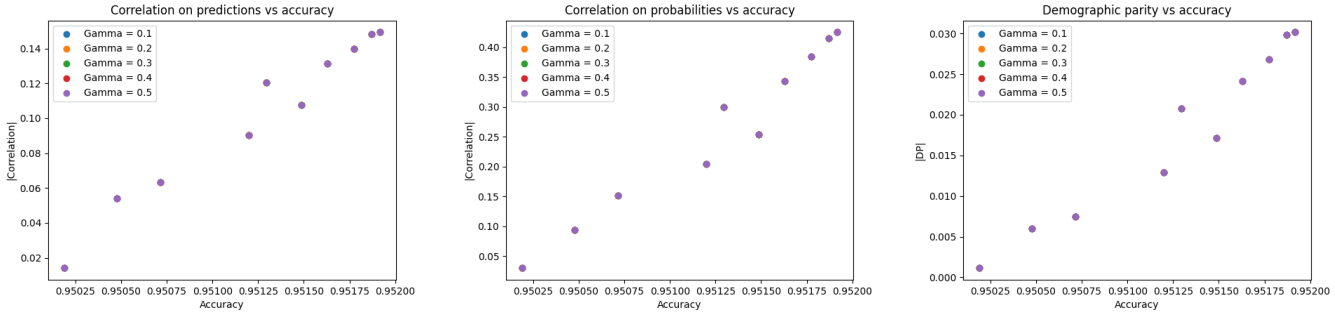
Figure 6.49: Gamma variation - Post-processing strategy with L1L2 with a Multi Layer Perceptron on GERMAN dataset: comparison between accuracy and (a) correlation on the decisions (b) correlation on the predicted probabilities (c) demographic parity.



(a) (b) (c)

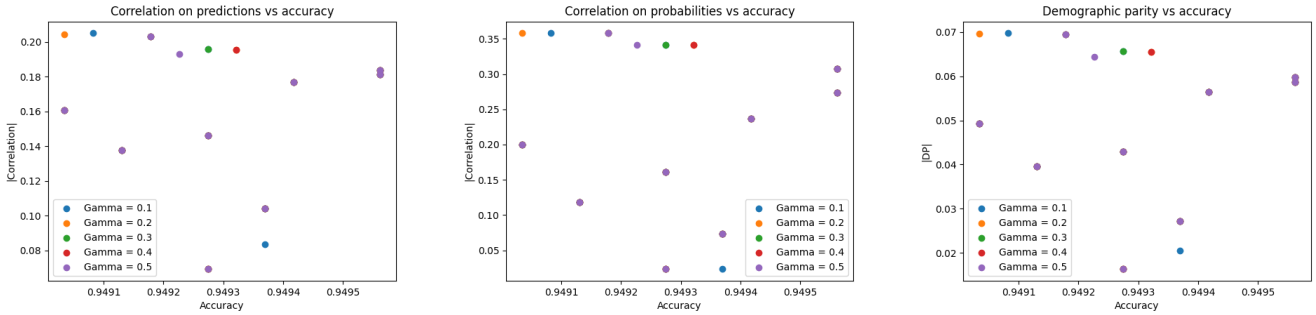
Figure 6.50: Gamma variation - Post-processing strategy with L1L2 with a Support Vector Machine on GERMAN dataset: comparison between accuracy and (a) correlation on the decisions (b) correlation on the predicted probabilities (c) demographic parity.

On the LAW dataset



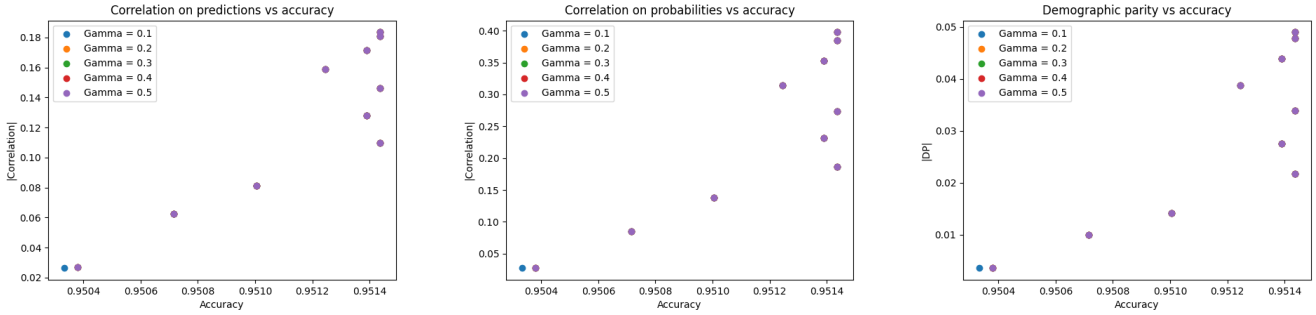
(a) (b) (c)

Figure 6.51: Gamma variation - Post-processing strategy with L1L2 with a Random Forest Classifier on LAW dataset: comparison between accuracy and (a) correlation on the decisions (b) correlation on the predicted probabilities (c) demographic parity.



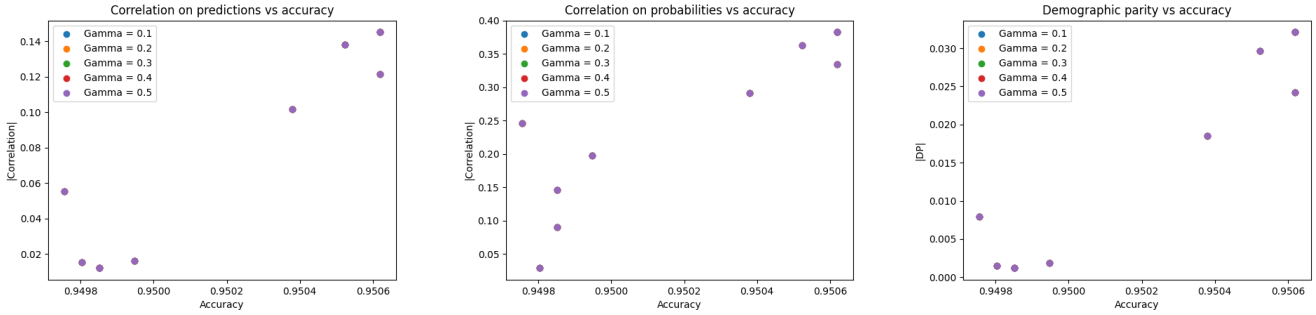
(a) (b) (c)

Figure 6.52: Gamma variation - Post-processing strategy with L1L2 with a Decision Tree Classifier on LAW dataset: comparison between accuracy and (a) correlation on the decisions (b) correlation on the predicted probabilities (c) demographic parity.



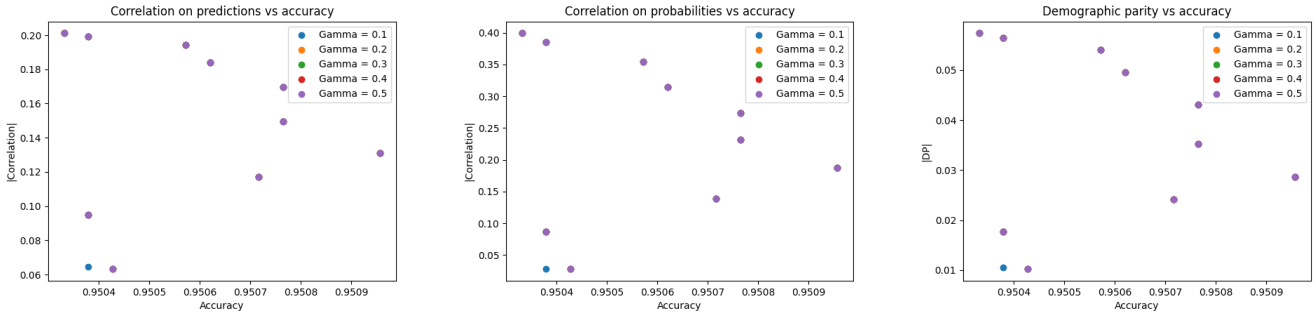
(a) (b) (c)

Figure 6.53: Gamma variation - Post-processing strategy with L1L2 with a Logistic Regression on LAW dataset: comparison between accuracy and (a) correlation on the decisions (b) correlation on the predicted probabilities (c) demographic parity.



(a) (b) (c)

Figure 6.54: Gamma variation - Post-processing strategy with L1L2 with a Multi Layer Perceptron on LAW dataset: comparison between accuracy and (a) correlation on the decisions (b) correlation on the predicted probabilities (c) demographic parity.



(a) (b) (c)

Figure 6.55: Gamma variation - Post-processing strategy with L1L2 with a Support Vector Machine on LAW dataset: comparison between accuracy and (a) correlation on the decisions (b) correlation on the predicted probabilities (c) demographic parity.

H Naive strategy

H.1 Results overview

On the ADULT dataset

Method	Classifier	Accuracy	Corr. Deci.	Corr. Proba.	Dem. Par.	Modif. F1
NaiveAvg	DTC	0.8531	-0.1240	-0.3008	-0.0717	0.8891
NaiveAvg	LR	0.8405	-0.0397	-0.2203	-0.0182	0.9056
NaiveAvg	MLP	0.8366	0.0053	-0.1832	0.0023	**0.9101**
NaiveAvg	RFC	0.8467	-0.0927	-0.3265	-0.0372	0.9010
NaiveAvg	SVM	0.8405	-0.0467	-0.1944	-0.0149	0.9071
Naive05	DTC	0.8531	-0.1240	-0.3008	-0.0717	0.8891
Naive05	LR	0.8398	-0.0315	-0.2156	-0.0128	0.9076
Naive05	MLP	0.8373	0.0172	-0.1756	0.0066	**0.9087**
Naive05	RFC	0.8467	-0.0927	-0.3265	-0.0372	0.9010
Naive05	SVM	0.8402	-0.0460	-0.1923	-0.0142	0.9072
NaivePrior	DTC	0.8531	-0.1240	-0.2981	-0.0717	0.8891
NaivePrior	LR	0.8394	0.0094	-0.1513	0.0036	**0.9112**
NaivePrior	MLP	0.8362	0.0552	-0.1204	0.0202	0.9023
NaivePrior	RFC	0.8430	-0.0632	-0.2750	-0.0215	0.9057
NaivePrior	SVM	0.8398	-0.0386	-0.1578	-0.0117	0.9080
Naive2	DTC	0.8401	-0.1026	-0.3206	-0.0437	0.8945
Naive2	LR	0.8366	-0.0170	-0.2194	-0.0071	0.9081
Naive2	MLP	0.8340	NaN	0.0008	NaN	NaN
Naive2	RFC	0.8355	-0.0300	-0.3373	-0.0044	**0.9085**
Naive2	SVM	0.8341	0.0401	-0.1737	0.0027	0.9084
Naive5	DTC	0.8337	0.0025	0.0426	0.0001	**0.9093**
Naive5	LR	0.8340	0.0168	-0.1335	0.0038	0.9079
Naive5	MLP	0.8340	NaN	0.0008	NaN	NaN
Naive5	RFC	0.8340	NaN	-0.1899	NaN	NaN
Naive5	SVM	0.8339	0.0179	0.0227	0.0005	0.9092

Table 6.41: Results of all Naive strategies on ADULT.

On the BANK dataset

Method	Classifier	Accuracy	Corr. Deci.	Corr. Proba.	Dem. Par.	Modif. F1
NaiveAvg	DTC	0.9013	-0.0895	-0.1207	-0.1079	0.8967
NaiveAvg	LR	0.9014	-0.1053	-0.1559	-0.1209	0.8901
NaiveAvg	MLP	0.8946	-0.1510	-0.1929	-0.1999	0.8448
NaiveAvg	RFC	0.8957	-0.1220	-0.1873	-0.0867	0.9044
NaiveAvg	SVM	0.8837	-0.0165	-0.1137	-0.0061	**0.9356**
Naive05	DTC	0.8426	-0.0860	-0.1032	-0.1495	0.8465
Naive05	LR	0.9016	-0.1118	-0.1644	-0.1468	0.8768
Naive05	MLP	0.8937	-0.1543	-0.1957	-0.2157	0.8354
Naive05	RFC	0.8964	-0.1200	-0.1888	-0.0915	0.9024
Naive05	SVM	0.8841	-0.0196	-0.1166	-0.0079	**0.9350**
NaivePrior	DTC	0.9014	-0.1855	-0.2541	-0.2321	0.8293
NaivePrior	LR	0.9018	-0.1936	-0.2825	-0.2306	0.8303
NaivePrior	MLP	0.8947	-0.1908	-0.2384	-0.2559	0.8125
NaivePrior	RFC	0.8959	-0.1323	-0.3123	-0.0947	0.9006
NaivePrior	SVM	0.8837	-0.0270	-0.1789	-0.0101	**0.9338**
Naive2	DTC	0.8920	-0.1278	-0.1469	-0.1238	0.8840
Naive2	LR	0.8921	-0.1190	-0.1976	-0.1005	0.8958
Naive2	MLP	0.8832	-0.1899	-0.2505	-0.2027	0.8381
Naive2	RFC	0.8937	-0.1296	-0.2251	-0.1032	0.8952
Naive2	SVM	0.8824	-0.0069	-0.1499	-0.0011	**0.9371**
Naive5	DTC	0.8916	-0.1253	-0.1462	-0.1210	0.8853
Naive5	LR	0.8925	-0.1156	-0.1931	-0.0947	**0.8989**
Naive5	MLP	0.8898	-0.2159	-0.2694	-0.2092	0.8374
Naive5	RFC	0.8937	-0.1306	-0.2183	-0.1022	0.8958
Naive5	SVM	0.8830	NaN	-0.1731	NaN	NaN

Table 6.42: Results of all Naive strategies on BANK.

On the COMPAS dataset

Method	Classifier	Accuracy	Corr. Deci.	Corr. Proba.	Dem. Par.	Modif. F1
NaiveAvg	DTC	0.6646	-0.2253	-0.2639	-0.2282	0.7142
NaiveAvg	LR	0.6789	-0.2421	-0.3049	-0.2413	0.7166
NaiveAvg	MLP	0.6808	-0.2253	-0.2816	-0.2255	0.7246
NaiveAvg	RFC	0.6800	-0.2427	-0.2930	-0.2435	0.7162
NaiveAvg	SVM	0.6793	-0.2393	-0.3055	-0.2388	**0.7179**
Naive05	DTC	0.6715	-0.2199	-0.2505	-0.2201	0.7217
Naive05	LR	0.6790	-0.2423	-0.3050	-0.2416	0.7165
Naive05	MLP	0.6798	-0.2280	-0.2824	-0.2280	**0.7230**
Naive05	RFC	0.6776	-0.2374	-0.2947	-0.2384	0.7171
Naive05	SVM	0.6793	-0.2393	-0.3055	-0.2388	0.7179
NaivePrior	DTC	0.6683	-0.2572	-0.2594	-0.2596	0.7025
NaivePrior	LR	0.6785	-0.2412	-0.3041	-0.2405	0.7168
NaivePrior	MLP	0.6805	-0.2287	-0.2821	-0.2288	**0.7230**
NaivePrior	RFC	0.6793	-0.2376	-0.3166	-0.2386	0.7180
NaivePrior	SVM	0.6792	-0.2388	-0.3051	-0.2384	0.7180
Naive2	DTC	0.5816	-0.1851	-0.1863	-0.1880	0.6778
Naive2	LR	0.5738	-0.1482	-0.1712	-0.1504	0.6850
Naive2	MLP	0.5525	-0.1531	-0.1964	-0.1495	0.6699
Naive2	RFC	0.6275	-0.2307	-0.2557	-0.2303	**0.6914**
Naive2	SVM	0.5771	-0.1470	-0.1693	-0.1483	0.6880
Naive5	DTC	0.5233	-0.0544	-0.1017	-0.0553	0.6735
Naive5	LR	0.5800	-0.1240	-0.0699	-0.1026	0.7046
Naive5	MLP	0.4681	0.0239	-0.0526	0.0174	0.6341
Naive5	RFC	0.5839	-0.1221	-0.1466	-0.1023	**0.7076**
Naive5	SVM	0.5793	-0.1196	-0.0687	-0.0984	0.7054

Table 6.43: Results of all Naive strategies on COMPAS.

On the GERMAN dataset

Method	Classifier	Accuracy	Corr. Deci.	Corr. Proba.	Dem. Par.	Modif. F1
NaiveAvg	DTC	0.7280	-0.0394	-0.0913	-0.0425	**0.8271**
NaiveAvg	LR	0.7370	-0.0777	-0.1258	-0.0843	0.8167
NaiveAvg	MLP	0.4560	-0.0312	-0.0312	-0.0390	0.6185
NaiveAvg	RFC	0.7160	0.0418	-0.1551	0.0218	0.8268
NaiveAvg	SVM	0.7030	0.0343	0.0052	0.0062	0.8235
Naive05	DTC	0.7280	-0.0394	-0.0913	-0.0425	**0.8271**
Naive05	LR	0.7300	-0.1007	-0.1281	-0.1135	0.8007
Naive05	MLP	0.4560	-0.0312	-0.0312	-0.0390	0.6185
Naive05	RFC	0.7160	0.0418	-0.1551	0.0218	0.8268
Naive05	SVM	0.7030	0.0343	0.0052	0.0062	0.8235
NaivePrior	DTC	0.7280	-0.0394	-0.0607	-0.0425	0.8271
NaivePrior	LR	0.7260	-0.0295	-0.0574	-0.0337	**0.8291**
NaivePrior	MLP	0.4560	-0.0312	-0.0312	-0.0390	0.6185
NaivePrior	RFC	0.7210	0.0702	-0.0469	0.0428	0.8225
NaivePrior	SVM	0.7030	0.0343	0.0082	0.0062	0.8235
Naive2	DTC	0.7270	-0.0398	-0.0943	-0.0389	**0.8278**
Naive2	LR	0.7440	-0.0874	-0.1133	-0.0923	0.8178
Naive2	MLP	0.4560	-0.0312	-0.0312	-0.0390	0.6185
Naive2	RFC	0.7070	0.0243	-0.1487	0.0080	0.8256
Naive2	SVM	0.7000	NaN	-0.0428	NaN	NaN
Naive5	DTC	0.7020	-0.0375	-0.1161	-0.0313	0.8141
Naive5	LR	0.7260	-0.1051	-0.1347	-0.1102	0.7996
Naive5	MLP	0.4560	-0.0312	-0.0312	-0.0390	0.6185
Naive5	RFC	0.7020	0.0217	-0.1567	0.0025	**0.8241**
Naive5	SVM	0.7000	NaN	-0.1572	NaN	NaN

Table 6.44: Results of all Naive strategies on GERMAN.

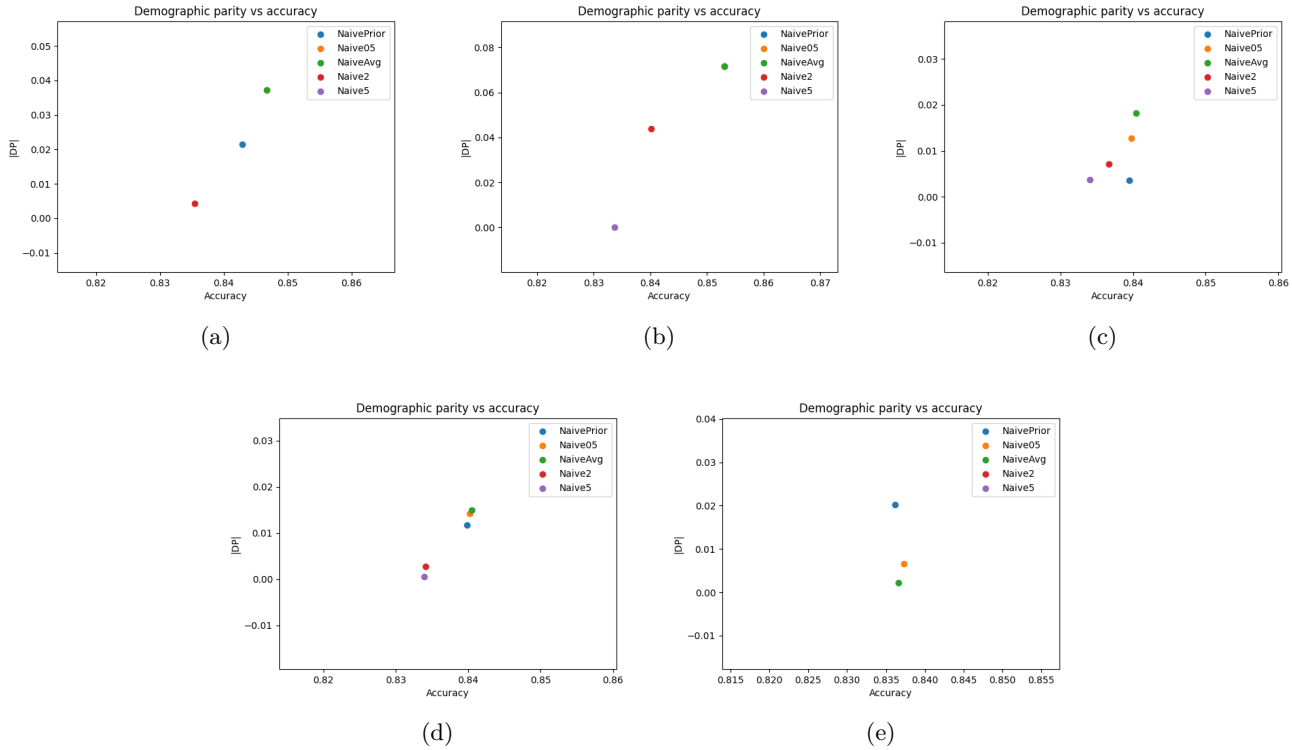
On the LAW dataset

Method	Classifier	Accuracy	Corr. Deci.	Corr. Proba.	Dem. Par.	Modif. F1
NaiveAvg	DTC	0.9494	-0.2025	-0.3622	-0.0690	0.9401
NaiveAvg	LR	0.9516	-0.1611	-0.3866	-0.0381	0.9567
NaiveAvg	MLP	0.9506	-0.1362	-0.3891	-0.0254	0.9625
NaiveAvg	RFC	0.9514	-0.1370	-0.4252	-0.0239	**0.9636**
NaiveAvg	SVM	0.9508	-0.1780	-0.3909	-0.0453	0.9527
Naive05	DTC	0.9494	-0.2025	-0.3622	-0.0690	0.9401
Naive05	LR	0.9512	-0.1702	-0.3881	-0.0435	0.9539
Naive05	MLP	0.9509	-0.1414	-0.3904	-0.0282	0.9613
Naive05	RFC	0.9514	-0.1370	-0.4252	-0.0239	**0.9636**
Naive05	SVM	0.9508	-0.1921	-0.3938	-0.0531	0.9488
NaivePrior	DTC	0.9490	-0.1947	-0.3572	-0.0680	0.9404
NaivePrior	LR	0.9512	-0.1532	-0.3530	-0.0375	0.9568
NaivePrior	MLP	0.9506	-0.1225	-0.3484	-0.0245	0.9629
NaivePrior	RFC	0.9514	-0.1305	-0.3933	-0.0236	**0.9638**
NaivePrior	SVM	0.9509	-0.1698	-0.3536	-0.0446	0.9531
Naive2	DTC	0.9499	-0.1992	-0.3595	-0.0667	0.9415
Naive2	LR	0.9504	-0.0802	-0.3755	-0.0086	0.9705
Naive2	MLP	0.9501	-0.0272	-0.3497	-0.0012	**0.9739**
Naive2	RFC	0.9500	NaN	-0.4393	NaN	NaN
Naive2	SVM	0.9398	-0.2897	-0.3964	-0.1568	0.8889
Naive5	DTC	0.9500	NaN	-0.1136	NaN	NaN
Naive5	LR	0.9500	NaN	-0.0936	NaN	NaN
Naive5	MLP	0.9500	NaN	-0.0339	NaN	NaN
Naive5	RFC	0.9500	NaN	-0.1897	NaN	NaN
Naive5	SVM	0.9500	NaN	-0.0996	NaN	NaN

Table 6.45: Results of all Naive strategies on LAW.

H.2 Graphics overview

On the ADULT dataset

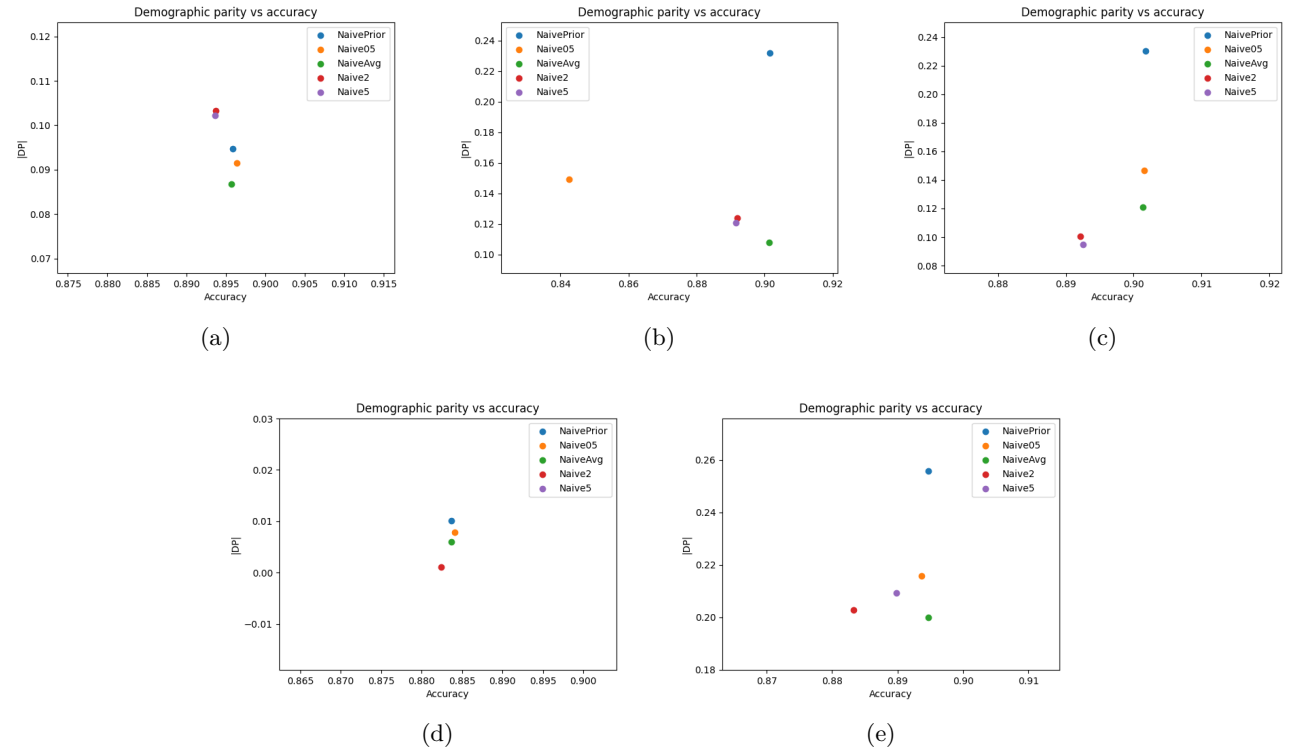


(a) (b) (c)

(d) (e)

Figure 6.56: Comparison between accuracy and the absolute value of demographic parity on the ADULT dataset for all Naive strategies with a (a) Random Forest Classifier (b) Decision Tree Classifier (c) Logistic Regression (d) Support Machine Vector (e) Multi Layer Perceptron.

On the BANK dataset

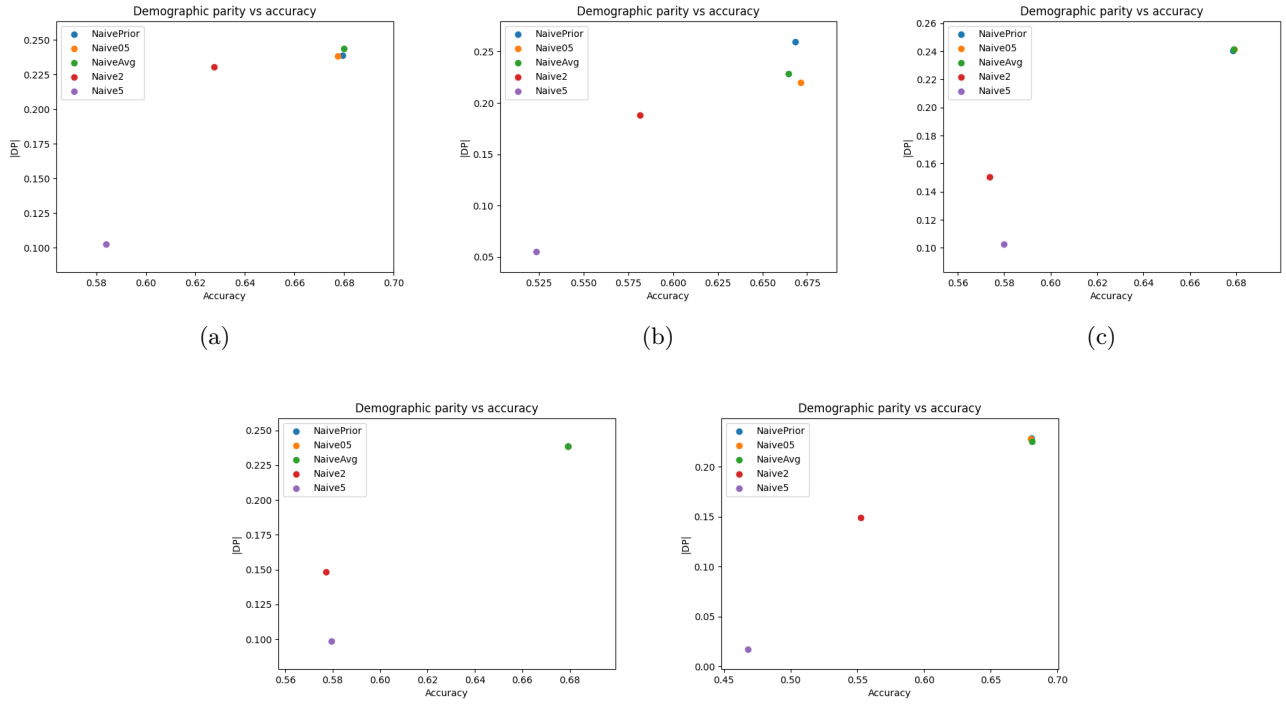


(a) (b) (c)

(d) (e)

Figure 6.57: Comparison between accuracy and the absolute value of demographic parity on the BANK dataset for all Naive strategies with a (a) Random Forest Classifier (b) Decision Tree Classifier (c) Logistic Regression (d) Support Machine Vector (e) Multi Layer Perceptron.

On the COMPAS dataset

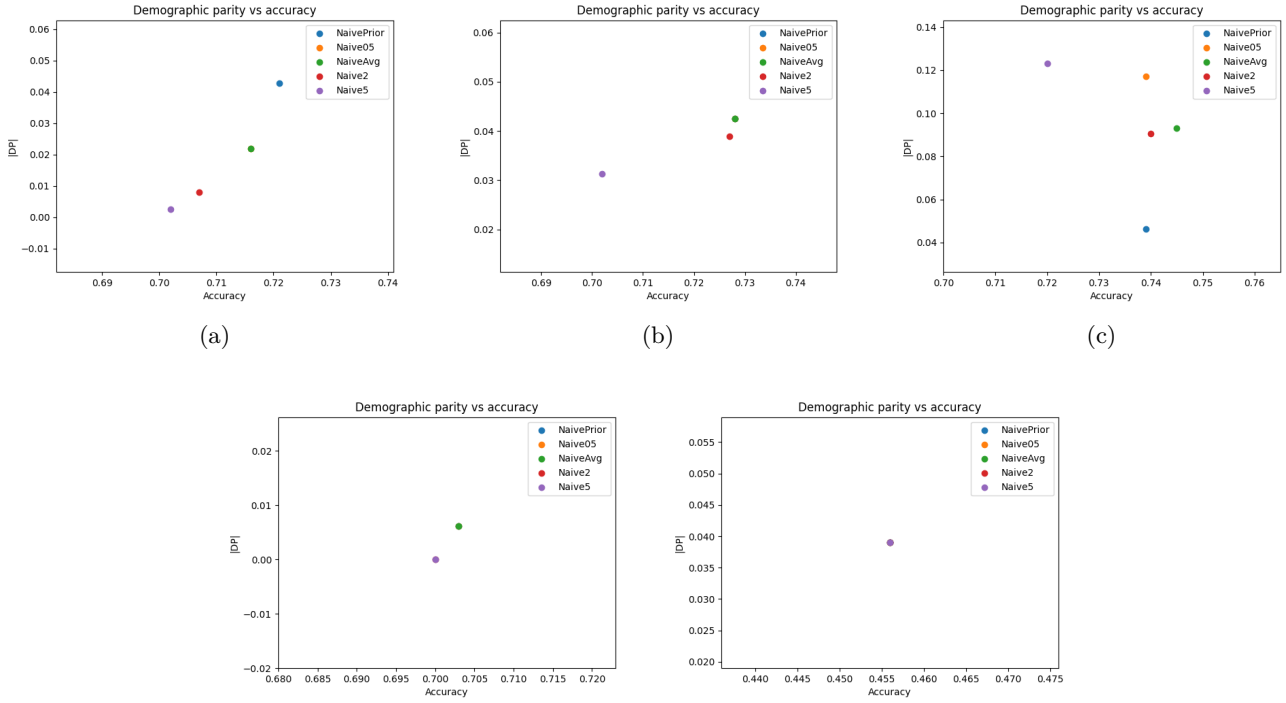


(a) (b) (c)

(d) (e)

Figure 6.58: Comparison between accuracy and the absolute value of demographic parity on the COMPAS dataset for all Naive strategies with a (a) Random Forest Classifier (b) Decision Tree Classifier (c) Logistic Regression (d) Support Machine Vector (e) Multi Layer Perceptron.

On the GERMAN dataset



(a) (b) (c)

(d) (e)

Figure 6.59: Comparison between accuracy and the absolute value of demographic parity on the GERMAN dataset for all Naive strategies with a (a) Random Forest Classifier (b) Decision Tree Classifier (c) Logistic Regression (d) Support Machine Vector (e) Multi Layer Perceptron.

On the LAW dataset

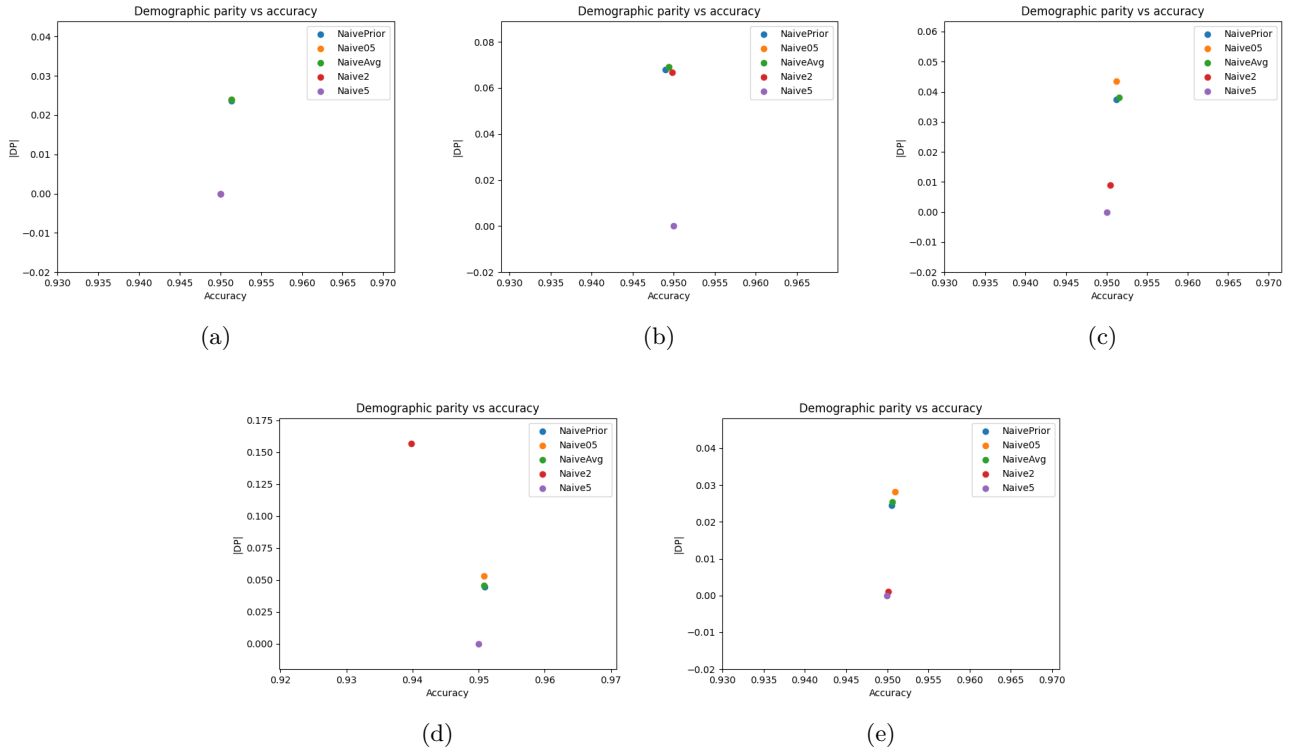


Figure 6.60: Comparison between accuracy and the absolute value of demographic parity on the LAW dataset for all Naive strategies with a (a) Random Forest Classifier (b) Decision Tree Classifier (c) Logistic Regression (d) Support Machine Vector (e) Multi Layer Perceptron.

I GAN-based strategy

I.1 Results overview without fairness

On the ADULT dataset

Classifier	Accuracy	Corr. Deci.	Corr. Proba.	|DP|	Modif. F1
DTC	0.8190 ± 0.0148	-0.2058 ± 0.0141	-0.3178 ± 0.0358	0.1716 ± 0.0236	0.8236 ± 0.0190
LR	0.8293 ± 0.0085	-0.2080 ± 0.0280	-0.3252 ± 0.0250	0.1735 ± 0.0306	0.8279 ± 0.0190
MLP	0.8209 ± 0.0080	-0.2069 ± 0.0172	-0.3192 ± 0.0114	0.1782 ± 0.0200	0.8214 ± 0.0137
RFC	0.8384 ± 0.0077	-0.2012 ± 0.0273	-0.3874 ± 0.0153	0.1517 ± 0.0291	**0.8433** ± 0.0177
SVM	0.8307 ± 0.0085	-0.2044 ± 0.0286	-0.3142 ± 0.0240	0.1697 ± 0.0310	0.8305 ± 0.0191

Table 6.46: Average and standard deviation for the results of five runs of the GAN-based strategy on ADULT. This strategy does not include fairness. Accuracy training steps: 350.

On the BANK dataset

Classifier	Accuracy	Corr. Deci.	Corr. Proba.	|DP|	Modif. F1
DTC	0.8900 ± 0.0061	-0.2590 ± 0.0655	-0.2730 ± 0.0468	0.3285 ± 0.0627	0.7655 ± 0.0428
LR	0.8844 ± 0.0062	-0.2661 ± 0.0446	-0.3021 ± 0.0426	0.4116 ± 0.0482	0.7066 ± 0.0368
MLP	0.8773 ± 0.0097	-0.2617 ± 0.0537	-0.2901 ± 0.0384	0.4181 ± 0.0705	0.6997 ± 0.0539
RFC	0.8920 ± 0.0029	-0.2888 ± 0.0510	-0.3821 ± 0.0440	0.2761 ± 0.0514	**0.7992** ± 0.0311
SVM	0.8856 ± 0.0056	-0.2850 ± 0.0459	-0.3197 ± 0.0410	0.4286 ± 0.0536	0.6946 ± 0.0425

Table 6.47: Average and standard deviation for the results of five runs of the GAN-based strategy on BANK. This strategy does not include fairness. Accuracy training steps: 400.

On the COMPAS dataset

Classifier	Accuracy	Corr. Deci.	Corr. Proba.	|DP|	Modif. F1
DTC	0.6377 ± 0.0253	-0.2081 ± 0.1068	-0.2729 ± 0.0897	0.1996 ± 0.1035	0.7099 ± 0.0290
LR	0.6566 ± 0.0165	-0.2726 ± 0.0750	-0.3283 ± 0.0992	0.2527 ± 0.0724	0.6990 ± 0.0316
MLP	0.6468 ± 0.0134	-0.2115 ± 0.0599	-0.2873 ± 0.0998	0.1941 ± 0.0549	**0.7176** ± 0.0201
RFC	0.6576 ± 0.0169	-0.2444 ± 0.0473	-0.3529 ± 0.0777	0.2225 ± 0.0502	0.7125 ± 0.0151
SVM	0.6602 ± 0.0151	-0.2654 ± 0.0833	-0.3277 ± 0.0958	0.2452 ± 0.0814	0.7043 ± 0.0336

Table 6.48: Average and standard deviation for the results of five runs of the GAN-based strategy on COMPAS. This strategy does not include fairness. Accuracy training steps: 4000.

On the GERMAN dataset

Classifier	Accuracy	Corr. Deci.	Corr. Proba.	|DP|	Modif. F1
DTC	0.7300 ± 0.0415	-0.1191 ± 0.1827	-0.1273 ± 0.1462	0.1304 ± 0.0559	0.7937 ± 0.0252
LR	0.7420 ± 0.0319	-0.1738 ± 0.1539	-0.1708 ± 0.1530	0.1636 ± 0.1031	0.7864 ± 0.0362
MLP	0.7420 ± 0.0331	-0.1395 ± 0.1603	-0.1353 ± 0.1553	0.1335 ± 0.0961	0.7995 ± 0.0306
RFC	0.7220 ± 0.0412	-0.1276 ± 0.1474	-0.1907 ± 0.1511	0.0510 ± 0.0208	**0.8201** ± 0.0234
SVM	0.7420 ± 0.0319	-0.1523 ± 0.1398	-0.1709 ± 0.1532	0.1456 ± 0.0962	0.7942 ± 0.0358

Table 6.49: Average and standard deviation for the results of five runs of the GAN-based strategy on GERMAN. This strategy does not include fairness. Accuracy training steps: 5000.

On the LAW dataset

Classifier	Accuracy	Corr. Deci.	Corr. Proba.	|DP|	Modif. F1
DTC	0.9514 ± 0.0030	-0.1199 ± 0.0014	-0.1561 ± 0.0692	0.0278 ± 0.0021	0.9617 ± 0.0003
LR	0.9511 ± 0.0025	-0.0570 ± 0.0106	-0.2593 ± 0.0974	0.0075 ± 0.0065	**0.9713** ± 0.0031
MLP	0.9512 ± 0.0027	-0.0952 ± 0.0302	-0.2793 ± 0.0893	0.0171 ± 0.0048	0.9668 ± 0.0026
RFC	0.9517 ± 0.0033	NaN	-0.3522 ± 0.0455	NaN	NaN
SVM	0.9511 ± 0.0025	-0.0625 ± 0.0144	-0.2622 ± 0.1018	0.0100 ± 0.0072	0.9701 ± 0.0035

Table 6.50: Average and standard deviation for the results of five runs of the GAN-based strategy on LAW. This strategy does not include fairness. Accuracy training steps: 1250.

I.2 Results overview with fairness

On the ADULT dataset

Classifier	Accuracy	Corr. Deci.	Corr. Proba.	|DP|	Modif. F1
DTC	0.8285 ± 0.0119	-0.1822 ± 0.0209	-0.2906 ± 0.0065	0.1395 ± 0.0289	0.8442 ± 0.0200
LR	0.8358 ± 0.0049	-0.1562 ± 0.0103	-0.2329 ± 0.0159	0.1228 ± 0.0076	0.8560 ± 0.0040
MLP	0.8275 ± 0.0050	-0.1698 ± 0.0133	-0.2393 ± 0.0194	0.1398 ± 0.0108	0.8435 ± 0.0034
RFC	0.8464 ± 0.0032	-0.1524 ± 0.0116	-0.3312 ± 0.0075	0.1017 ± 0.0101	**0.8716** ± 0.0039
SVM	0.8378 ± 0.0054	-0.1482 ± 0.0096	-0.2254 ± 0.0154	0.1155 ± 0.0065	0.8605 ± 0.0039

Table 6.51: Average and standard deviation for the results of five runs of the GAN-based strategy on ADULT. This strategy include fairness. Accuracy training steps: 350. Fairness training steps: 200.

On the BANK dataset

Classifier	Accuracy	Corr. Deci.	Corr. Proba.	|DP|	Modif. F1
DTC	0.8849 ± 0.0093	-0.0151 ± 0.0515	0.0488 ± 0.0319	0.0605 ± 0.0463	0.9114 ± 0.0244
LR	0.8802 ± 0.0073	0.0672 ± 0.0075	0.1393 ± 0.0158	0.1008 ± 0.0218	0.8896 ± 0.0140
MLP	0.8734 ± 0.0084	0.0654 ± 0.0133	0.1251 ± 0.0175	0.1011 ± 0.0277	0.8859 ± 0.0171
RFC	0.8889 ± 0.0028	0.0286 ± 0.0168	0.0209 ± 0.0208	0.0245 ± 0.0106	**0.9302** ± 0.0058
SVM	0.8819 ± 0.0069	0.0556 ± 0.0098	0.1231 ± 0.0183	0.0803 ± 0.0196	0.9004 ± 0.0121

Table 6.52: Average and standard deviation for the results of five runs of the GAN-based strategy on BANK. This strategy include fairness. Accuracy training steps: 400. Fairness training steps: 250.

On the COMPAS dataset

Classifier	Accuracy	Corr. Deci.	Corr. Proba.	|DP|	Modif. F1
DTC	0.6491 ± 0.0168	-0.1910 ± 0.0446	-0.2690 ± 0.0462	0.1777 ± 0.0426	**0.7255** ± 0.0103
LR	0.6563 ± 0.0143	-0.2402 ± 0.0330	-0.2867 ± 0.0347	0.2229 ± 0.0335	0.7116 ± 0.0175
MLP	0.6491 ± 0.0080	-0.2043 ± 0.0322	-0.2631 ± 0.0461	0.1885 ± 0.0361	0.7213 ± 0.0110
RFC	0.6540 ± 0.0138	-0.2452 ± 0.0516	-0.3208 ± 0.0489	0.2268 ± 0.0485	0.7086 ± 0.0131
SVM	0.6546 ± 0.0145	-0.2447 ± 0.0305	-0.2891 ± 0.0349	0.2263 ± 0.0313	0.7092 ± 0.0128

Table 6.53: Average and standard deviation for the results of five runs of the GAN-based strategy on COMPAS. This strategy include fairness. Accuracy training steps: 4000. Fairness training steps: 2500.

On the GERMAN dataset

Classifier	Accuracy	Corr. Deci.	Corr. Proba.	|DP|	Modif. F1
DTC	0.7120 ± 0.0449	0.0867 ± 0.0836	0.2329 ± 0.2231	0.0710 ± 0.0305	0.8062 ± 0.0361
LR	0.7300 ± 0.0482	0.1065 ± 0.0956	0.2407 ± 0.1773	0.0902 ± 0.0373	0.8101 ± 0.0411
MLP	0.7240 ± 0.0393	0.0911 ± 0.1456	0.2232 ± 0.1875	0.1243 ± 0.0419	0.7927 ± 0.0234
RFC	0.7220 ± 0.0431	0.0501 ± 0.0069	0.2852 ± 0.2629	0.0125 ± 0.0006	**0.8341** ± 0.0308
SVM	0.7320 ± 0.0542	0.0844 ± 0.1370	0.2181 ± 0.1642	0.1053 ± 0.0230	0.8052 ± 0.0331

Table 6.54: Average and standard deviation for the results of five runs of the GAN-based strategy on GERMAN. This strategy include fairness. Accuracy training steps: 5000. Fairness training steps: 3000.

On the LAW dataset

Classifier	Accuracy	Corr. Deci.	Corr. Proba.	|DP|	Modif. F1
DTC	0.9444 ± 0.0086	-0.0861 ± 0.0287	-0.2188 ± 0.1147	0.0241 ± 0.0203	0.9599 ± 0.0141
LR	0.9513 ± 0.0027	-0.0953 ± 0.0489	-0.3740 ± 0.0601	0.0174 ± 0.0190	0.9667 ± 0.0090
MLP	0.9468 ± 0.0094	-0.1350 ± 0.0507	-0.3404 ± 0.0780	0.0412 ± 0.0326	0.9528 ± 0.0209
RFC	0.9517 ± 0.0031	-0.0505 ± 0.0001	-0.4549 ± 0.1006	0.0030 ± 0.0001	**0.9738** ± 0.0014
SVM	0.9511 ± 0.0030	-0.1022 ± 0.0388	-0.3805 ± 0.0672	0.0188 ± 0.0175	0.9659 ± 0.0082

Table 6.55: Average and standard deviation for the results of five runs of the GAN-based strategy on LAW. This strategy include fairness. Accuracy training steps: 1250. Fairness training steps: 500.

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