UCLouvain

LOUVAIN
School of Management

Louvain School of Management

Fairness in machine learning : focus on post-processing methods

Auteur·e(s) : François GOUVERNEUR
Promoteur·rice(s) : Marco SAERENS
Année académique 2022-2023
Travail de fin d'études (TFE) en vue d'obtenir le titre de
Master (60) en Sciences de Gestion
Horaire de jour

Abstract

The popularity of machine learning derives from its application to decision-making in all fields. Classification is fundamental, as it allows data to be categorized on the basis of characteristics, with supervised classification learning from labeled data to predict outcomes. The question of fairness is essential to avoid biased results and promote equal treatment for all in many areas such as clinical testing or criminal justice. There are a number of different fairness measures, each more appropriate in certain situations. Promoting fairness in machine learning is crucial for responsible and fair decision-making. This thesis develops a new fair post-processing method, Optimal Swapping, and its greedy version, for binary score predictions in supervised classification. It aims to mitigate unfairness while maintaining overall accuracy. Comparison analyses with other post-processing techniques, using statistical tests such as Friedman, Nemenyi, and Wilcoxon signed-rank, evaluate the method's efficiency in promoting fairness. By contributing to robust and effective fairness techniques, this work advances fairness machine learning applications.

Acknowledgement

I would like to express my warmest thanks to all the people who supported me in the writing of this second master thesis.

First of all, I would like to thank my promoter, Marco Saerens, for his precious advice and availability all year long.

I would also like to deeply thank Flore Vancompernolle Vromman and Sylvain Courtain for their constant support and their pertinent suggestions and comments, which enabled me to accomplish this work.

Finally, I would like to express my special gratitude to my parents and relatives, who have always supported me in my ideas and projects.

François

Contents

1 Introduction

Machine learning is a fast-growing field at the cutting edge of computer science and mathematics. Its popularity is due to its growing application in decision-making processes in various fields. As research continues to evolve, machine learning is constantly opening up new possibilities and developing advances techniques.

One of its best-known tasks is classification, a fundamental process in which data samples are classified according to their features. By training classification models on labeled data, they are fitted to predict unseen sample data. This predictive capability allows to take data-driven decisions, which revolutionizes the way people solve problems and make choices.

Supervised classification is a well-known method in machine learning where a computer learns from labeled data to make predictions (Sen et al., 2020). For instance, in medical diagnosis, the computer can learn from labeled patient data to predict if a new patient has a specific disease. This approach is used in many fields such as healthcare, finance or marketing to solve problems and aid decision-making. However, it is essential to recognize that this process may not always be fair. For instance, if the data used for training contain biases or unfair treatment, the classifier may perpetuate these biases and lead to unfair outcomes.

Fairness in machine learning is crucial to avoid biased outcomes and ensure equal treatment for everyone (Sweeney, 2013) (Caliskan et al., 2017). Biases can arise from multiple sources, like imbalanced data or historical discrimination. Fairness can be achieved through group fairness, where different groups are treated equally, and individual fairness, where similar individuals receive similar treatment. There are a huge number of metrics for measuring the fairness of an algorithm, each more relevant to the goal being pursued. In real-world applications, fairness issues have been evident in scenarios like clinical tests, loan approvals, and criminal justice (Manrai et al., 2016) (Cozarenco & Szafarz, 2018) (Faber, 2017). For example, a biased hiring algorithm might favor one group over others, leading to unfair employment practices. Identifying and addressing these fairness challenges is essential to build responsible and equitable machine learning systems.

In the domain of fairness in machine learning, three main types of techniques can be used: pre-processing, in-processing, and post-processing. This work specifically focuses on post-processing fairness techniques. These post-processing methods are applied after the initial predictions of the model. They aim to modify the predictions in a way that mitigates any unfairness while preserving the model's overall accuracy. This family of methods is attractive because it works with existing machine learning models, and fairness changes are introduced from the outside. Instead of modifying the internal workings of the model, post-processing methods focus on the results it produces. This makes them a flexible and less intrusive way of improving fairness in decision-making.

The main objective of this thesis is to develop a new fair post-processing method for supervised classification that focuses on binary score predictions. This new method called Optimal Swapping and its greedy version will be compared with other post-processing techniques using statistical tests such as Friedman, Nemenyi and Wilcoxon signed-rank. These tests will determine whether the new method performs significantly better than existing methods, and whether it can be a promising option for improving fairness in supervised classification. By conducting these evaluations, the thesis aims at contributing to the development of a robust and efficient post-processing method that promotes fairness in machine learning.

2 State of The Art

2.1 Introduction

In the field of machine learning, fairness plays a critical role to ensure equal treatment and preventing discrimination. Machine learning algorithms significantly impact various domains, including employment, credit scoring, and criminal justice. It is essential to prioritize fairness to avoid perpetuating social inequalities and reinforcing biases present in the data.

To achieve fairness in machine learning, it is necessary to identify and address biases that may result in unfair treatment based on attributes like ethnicity or gender (ProPublica, 2016). Fairness metrics provide a framework to measure and evaluate the potential discrimination. The integration of fairness considerations throughout the development and evaluation stages of machine learning systems maintain ethical standards and respects individual rights.

Unfairness in machine learning has ethical and societal implications (Barocas et al., 2019). Biased outcomes generated by algorithms systematically disadvantage specific individuals or groups, exacerbating social disparities and limiting equal opportunities. Losing trust in automated decision-making systems leads to more discrimination and harm to marginalized communities, which slows down social progress. Ethical frameworks are important for identifying biases, being transparent, and reducing harm from unfair algorithms.

This section explains where unfairness and bias in machine learning algorithms come from, and the various techniques that exist to eliminate them, with a particular focus on post-processing algorithms in the literature.

2.2 Bias in supervised classification

2.2.1 Supervised Classification

There are many machine learning techniques (Bishop, 2006), supervised classification is one of them that is used to categorize or classify data into predefined classes or categories based on labeled training examples. The technique is "supervised" because it is provided with a dataset containing instances composed of features and their corresponding output labels corresponding to classes/categories (Sen et al., 2020). This work is based on supervised classification techniques.

2.2.2 Types of bias

In supervised classification, various types of biases can affect the outcomes and performance of algorithms. These biases can arise from different stages of the machine learning pipeline, including data collection, algorithm design, and model evaluation. Here is an overview of different types of bias in machine learning inspired by Mehrabi et al. (2022) and Hellström et al. (2020) :

- **Algorithmic Bias**: refers to biases that are embedded in the design and implementation of machine learning algorithms (Baeza-Yates, 2018). They can arise due to the algorithm's assumptions, limitations, or the specific choices made during its development. For example, if an algorithm assumes that ethnicity is a strong predictor of college graduation, it may discriminate against individuals from minor ethnic group as investigated by Anderson et al. (2019)

- **Evaluation Bias**: refers to biases that occur during the evaluation of machine learning models (Suresh & Guttag, 2019). The choice of evaluation metrics and methods can introduce biases that favor certain outcomes or fail to capture important aspects of performance. For instance, if a model is evaluated only based on overall accuracy, it may ignore disparities in performance across different demographic groups.

- **Historical Bias**: occurs when the data used to build a machine learning model contain systematic distortions (Suresh & Guttag, 2019). This bias exists in the real world and can interfere with the data generation process even if sampling and feature selection are perfect. For example, if historical loan data is used to train a credit scoring model, which includes biases against marginalized groups, the model may perpetuate discrimination by denying loans unfairly.

- **Representation Bias**: arises when the features or variables used to represent the data in a machine learning model are insufficient, incomplete, or biased (Suresh & Guttag, 2019). If important factors that contribute to the problem are not adequately represented, the model may fail to capture the full complexity of the data. For example, if only a person's age and gender are used to predict their job performance, other relevant variables may be neglected, leading to biased results.

- **Sampling Bias**: is similar to representation bias and occurs when there is a non-random sampling of subgroups (Suresh & Guttag, 2019). If the sample is biased or unrepresentative, the resulting model may perform poorly on unseen data.

- **Amplification Bias**: happens when machine learning algorithms exacerbate existing biases in the data or amplify the impact of discriminatory patterns (Hall et al., 2022). For example, if a recommendation system suggests content based on users' past behavior, it can reinforce existing biases by repeatedly recommending similar content.

- **Automation Bias**: refers to the tendency of humans to blindly trust and rely on machine learning models, assuming they are objective and unbiased (Cummings, 2004). This bias can lead to the acceptance of erroneous or biased decisions made by algorithms without appropriate human oversight and intervention.

2.2.3 Real-life examples

Data biases can be particularly risky in sensitive areas like healthcare. For example, in medical research, the data often focuses on specific groups which can have harmful effects on underrepresented communities. A study demonstrated how excluding African-Americans from clinical studies led to their missclassification (Manrai et al., 2016). Another examination (Shaw & Corpas, n.d.) of a genotype dataset revealed a significant imbalance, with the majority (87%) being European while Asian and African populations only represented a small fraction (2% each).

Another area where data biases can have a negative impact concerns the financial institutions. Gender, ethnicity, age and other characteristics of an individual may influence the prediction of an algorithm on defaulting a loan (German dataset) or its tendency to subscribe to a term deposit (Bank dataset). A study of Cozarenco and Szafarz (2018) investigated whether microfinance institutions (MFI) in France display gender bias and therefore create additional barriers for female entrepreneurs

in accessing micro-credit lines for their businesses. Another study by Faber (2017) shows that "black and Latino borrowers are three times more likely to receive high-cost loans compared with whites, a practice that has accelerated since the 2007–8 subprime crisis".

These findings highlight the unfair impact of machine learning algorithms and artificial intelligence and show the importance of making it fairer.

2.3 Fairness techniques and approaches

Fairness in supervised classification can be addressed through a pipeline of techniques, namely pre-processing, in-processing, and post-processing fairness techniques (T. Mahoney, 2020). These techniques can be employed independently or in combination to tackle biases and promote fairness in the development of machine learning models.

2.3.1 Fairness techniques

Pre-processing fairness techniques consist in addressing biases before training machine learning models. These techniques include data augmentation, the creation of more data, data transformation and feature selection.
They offer the advantage of universality (Dunkelau, 2019), as they can be applied before any classification algorithm. However, there are potential drawbacks to consider. The quality of the data may be compromised, new biases might accidentally be introduced, and there is a risk of selecting inappropriate features.

In-processing fairness methods consist in the incorporation of fairness directly during the design of machine learning algorithms (Wan et al., 2023). It can be achieved by the use of fairness constraints in the training objective or by the use of an adversarial learning method to detect sensitive attributes in order to make predictions that are less dependent on such attributes.
On the positive side, in-processing techniques explicitly handle the trade-off between accuracy and fairness. However, there are drawbacks. They can decrease predictive accuracy by imposing constraints to reduce bias. These constraints can also increase training time due to added complexity (Inc, 2019). Additionally, the generalization of these techniques is restricted as they are often linked to specific learning algorithms (Woodworth et al., 2017).

Post-processing techniques propose adjustments to the output of machine learning models to achieve intended fairness goals. These techniques are the most diversified

and include adaptation of classification threshold (Hardt et al., 2016)(Pleiss et al., 2017), rejection of uncertain predictions (Kamiran et al., 2012) and individual bias detector (Lohia et al., 2018)(Lohia, 2021).
The main advantage of post-processing techniques is that they are highly compatible as they can be applied to any classifier. Additionally, they help in mitigating unforeseen discriminatory outcomes. However, there are drawbacks to consider. Those techniques may not be optimal as they only act on information that has already been learned, after fitting the model. They can also have a negative impact on the overall accuracy of the model. Furthermore, there is a risk of disparate treatment, where individuals with similar characteristics are treated differently based on their group membership.
Finally, post-processing techniques have the advantage of not changing the choices of the estimator but instead the way the estimated predictions function is used (Kleinberg et al., 2018).

2.3.2 Fairness pipeline

These techniques can be apply at every step of the fairness pipeline as it can be seen on Figure 2.1. An example of instantiation consists in transforming data into a fairer dataset using a fair pre-processing algorithm, learning a classifier from this transformed dataset and finally obtaining predictions from this classifier.

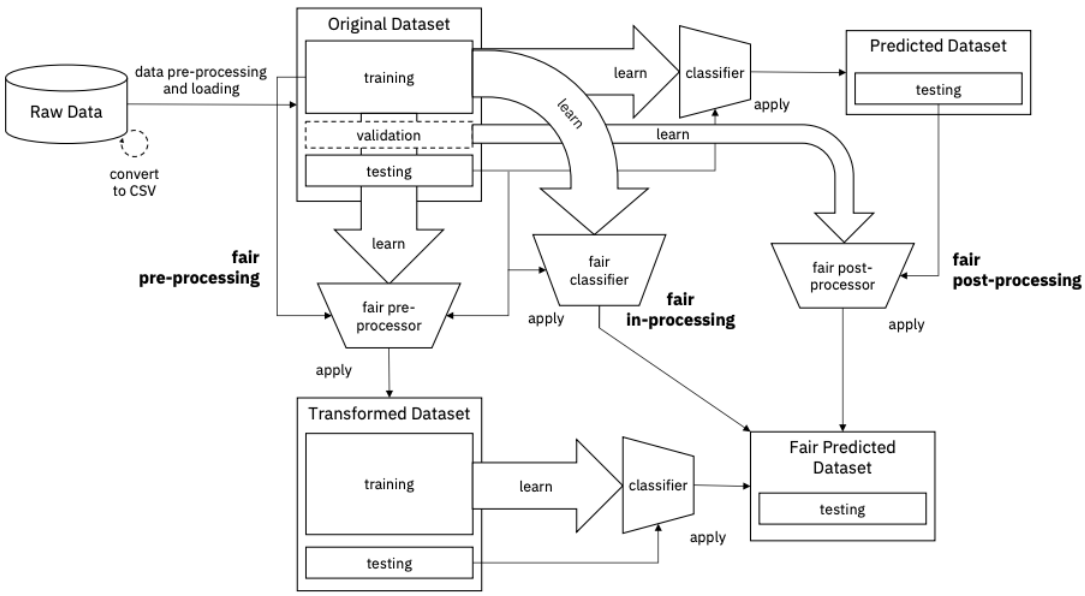

Fig. 2.1.: Fairness pipeline of AIF360 (Bellamy et al., 2018)

2.4 Metrics and definitions

Accuracy is used to quantify how often the model's predictions align with the true labels by measuring the proportion of correctly classified instances out of the total number of instances in a dataset. An instance is correctly classified if its predicted label matches its true label (Carvalho et al., 2019).
In machine learning, accuracy is generally considered as one the best measures to use. But even if it provides a general measure of the model's overall precision it may not be the most suitable metric in some cases. For example, a model with imbalanced labels will achieve a high accuracy by simply predicting the majority label most of the time. This is why other metrics need to be considered, such as fairness metrics.

Fairness is a complex multidimensional concept that has different definitions because each one focuses on different aspects and therefore requires different metrics. The choice of one or another depends on the specific context. For example, in criminal justice, focus is made on avoiding false positives or false negatives for different groups while in the hiring process the representation of different demographic groups is important. Maximizing one metric may result to the decrease in another because they sometimes conflict with each other (Saravanakumar, 2020).

Moreover, the search for fairness in machine learning and therefore the optimization of one of the above metrics may reduce the accuracy of the model.
The tradeoff between accuracy and fairness refers to the challenge of simultaneously achieving high accuracy in predictions while ensuring fairness and mitigating biases in the predictions.

2.4.1 Individual vs Group fairness

Fairness in machine learning may be seen from an individual perspective or a group perspective (Lumenova, n.d.).

Individual fairness ensures that similar individuals receive similar outcomes or decisions. According to Dwork et al. (2011), "a model is individually biased (or unfair) if there is a pair of valid inputs which are close to each other (according to an appropriate metric) but are treated differently by the model (different class label, or large difference in output)". For example, in the criminal justice system, individual fairness could mean that people who have committed similar crimes in similar circumstances should receive similar sentences, independently of their background or social status.

On the other hand, group fairness emphasizes fairness at the group level, focusing on avoiding discrimination or bias against certain groups. In this work, the main focus is on group fairness. For example, imagine a university admissions process that uses an algorithm to evaluate applicants. If the historical data favor certain schools or regions, the algorithm may perpetuate those biases and unfairly disadvantage some groups of applicants from other regions or schools

In the context of group fairness, the concepts of privileged (unprotected) and unprivileged (to be protected) groups play a crucial role in understanding and mitigating biases. Privileged and unprivileged groups refer to specific categories or attributes of individuals that may be associated with disparities or discrimination within a given context.
A privileged group typically refers to a group that benefits from certain advantages or favorable treatment in society. This group may have historically held positions of power, enjoyed social privileges, or experienced fewer barriers and inequalities. Being part of the privileged group can result in preferential treatment or more positive outcomes.
On the other hand, an unprivileged group refers to a group that faces systemic disadvantages or discriminatory treatment. This group may have historically experienced social, economic, or political marginalization, and may be subject to bias or discrimination in various domains. Individuals that belong to the unprivileged group are often the ones most affected by biased algorithms and may experience negative outcomes or further marginalization.

2.4.2 Fairness Metrics

There are several ways of classifying individuals in machine learning. In the case of this work, the classification is binary, meaning that the predicted labels of the individuals will be either positive or negative. Consequently, the definitions of the different metrics that follow are binary.

For the rest of this work, the sensitive attribute of an individual is denoted Z, which is used to distinguish between individuals from privileged and unprivileged groups. This attribute can only take two different values, $Z = \{\text{privileged, unprivileged}\}$. Model predictions consist in assigning a label C to each individual. Labels are binary so they can only take two values, positive or negative, represented respectively by C^+ (=1) and C^- (=0).

Here is a quick presentation of the most used group fairness and accuracy metrics in post-processing algorithms (IBM, n.d.).

		Actual class	
		True	**False**
Predicted class	**True**	True Positive (**TP**)	False Positive (**FP**)
	False	False Negative (**FN**)	True Negative (**TN**)

Tab. 2.1.: Confusion matrix

- *True Positive Rate (TPR)* : measures the proportion of actual positive instances that are correctly identified or classified as positive by the model. This measure is also known as the *sensitivity*.

$$TPR = \frac{TP}{TP + FN} \tag{2.1}$$

- *False Positive Rate (FPR)* : measures the proportion of actual negative instances that are incorrectly identified or classified as positive by the model.

$$FPR = \frac{FP}{TN + FP} \tag{2.2}$$

- *True Negative Rate (TNR)* : also known as *specificity*, the proportion of actual negative instances that are correctly identified or classified as negative by the model.

$$TNR = \frac{TN}{TN + FP} \tag{2.3}$$

- *False Negative Rate (FNR)* : measures the proportion of actual positive instances that are incorrectly identified or classified as negative by the model.

$$FNR = \frac{FN}{TP + FN} \tag{2.4}$$

- *Positive Predictive Value (PPV)* : measures the proportion of correctly classified instances among all the positive predictions. It is usually referred to as *precision*.

$$PPV = \frac{TP}{TP + FP} \tag{2.5}$$

- *Statistical Parity (SP)/Demographic Parity (DP)* : is the difference in the rate of favorable outcomes received by unprivileged group to the privileged group. The ideal value for perfect fairness without biases is 0. This metric does not say anything about the quality of the predictions for each group.

$$\left(\frac{\begin{array}{c}\text{Number of favorable instances}\\ \text{for unprivileged group}\end{array}}{\begin{array}{c}\text{Total number of instances}\\ \text{for unprivileged group}\end{array}}\right) - \left(\frac{\begin{array}{c}\text{Number of favorable instances}\\ \text{for privileged group}\end{array}}{\begin{array}{c}\text{Total number of instances}\\ \text{for privileged group}\end{array}}\right) \tag{2.6}$$

which estimates the following quantity with Z a random variable

$$p(C^+|Z = unprivileged) - p(C^+|Z = privileged) \tag{2.7}$$

- *Disparate Impact (DI)* : ratio of the rate of favorable outcomes received by the unprivileged group to the privileged group.

$$\left(\frac{\begin{array}{c}\text{Number of favorable instances}\\ \text{for unprivileged group}\end{array}}{\begin{array}{c}\text{Total number of instances}\\ \text{for unprivileged group}\end{array}}\right) / \left(\frac{\begin{array}{c}\text{Number of favorable instances}\\ \text{for privileged group}\end{array}}{\begin{array}{c}\text{Total number of instances}\\ \text{for privileged group}\end{array}}\right) \tag{2.8}$$

which estimates the following quantity with Z a random variable

$$\frac{p(C^+|Z = unprivileged)}{p(C^+|Z = privileged)} \tag{2.9}$$

- *Equal Opportunity difference (EOpp)* : difference of the true positives rate (TPR) between unprivileged and privileged groups.

$$\text{TPR}_{unprivileged} - \text{TPR}_{privileged} \tag{2.10}$$

- *Equalized Odds (EO)* : differences between unprivileged and privileged for both TPR and FPR.

$$\text{TPR}_{unprivileged} - \text{TPR}_{privileged} \tag{2.11}$$
$$\text{FPR}_{unprivileged} - \text{FPR}_{privileged} \tag{2.12}$$

The three most intuitive and common definitions of fairness in the literature are *demographic parity* (DP), *predictive parity* (PPV) and *equalized odds* (EO). However, it has been demonstrated that no more than one of these metrics can hold at the

same time for a well calibrated classifier (Saravanakumar, 2020). Therefore, each method needs to focus on optimizing one of these metrics.

2.5 Methods of postprocessing

As explained in Section 2.3, there are three main families of techniques for combating unfairness: pre-, in- and post-processing techniques. This work focuses on post-processing techniques. In this section, two methods are explored in particular as they will be used as points of comparison later on : ROC (Subsection 2.5.1) and Massaging (Subsection 2.5.2) methods.

2.5.1 Defining a critical reject region

The Reject Option-based Classification (ROC) is a post-processing fairness algorithm introduced by Kamiran et al. (2012). It takes into account the uncertainty of certain decisions by defining a critical region around a classification threshold. Instances falling within this band are classified as uncertain rather than being assigned to a specific class. In this zone, protected individuals are automatically selected meaning their label is positive, and unprotected individuals are negatively labeled. Other individual predictions remain unchanged.

The size of this rejection zone varies according to the margin parameter θ. The margin selected at the end is the one that produces results that respect the specified fairness conditions and are as accurate as possible.

2.5.2 Swapping privileged and unprivileged individuals

The Massaging (Mass) post-processing method proposed by the pioneering work of Calders et al. (2009) aims to mitigate bias in classification models by swapping the predicted labels of different groups. This is done by classifying candidates in two ordered lists. Protected (resp. unprotected) candidates are those whose scores is below (resp. above) the classification threshold. The lists are sorted in ascending order according to the distance of each individual between its score and the classification threshold. Finally, labels of the firsts M candidates of each list are swapped.

Experimental evaluations have shown that this approach reduces dependency from the dataset better than simple methods as removing dependent attributes from the training data. An other advantage is that this method is extremely fast. On the other

hand, the situation with final predictions might be far away in terms of fairness than what was requested.

2.5.3 Modified Naïve Bayesian approach

This approach developed in Calders and Verwer (2010) proposes to deal with the discrimination of individuals by using a modified naïve Bayesian classification. This type of classification is based on Bayes' theorem (Swinburne, 2004) with the 'naïve' assumption that all features are independent of each other. To do so, the probabilities of the decision being positive are changed in the model such that the number of positive labels stays relatively close to the count of positive labels in the dataset, with only minor deviations.

2.5.4 Adapting the classification threshold

The principle of thresholding methods is to give each group different threshold values (Hutchinson & Mitchell, 2019)(Zafar et al., 2017). By doing this, various fairness goals can be achieved. However, user intervention is needed to analyze the results and define the thresholds in each case. Reducing user intervention is crucial as their biases can affect the process. To achieve this, employing learnable thresholds is a good approach.

An example of this method is the Equalized Odds (EOP) post-processing method that ensures that the probability of being selected is the same for individuals from privileged and unprivileged group (Hardt et al., 2016). The main drawback of it is that it can potentially give bad calibrated predictions.

2.5.5 Calibration

To counteract potential miscalibration of threshold methods (Subsection 2.5.4), the process of calibration were developed. It ensures that "the probability of positive predictions is equal to that of positive examples" (Dawid, 1982). The main drawback of this process is that it cannot be applied to multiple groups or protected features at once (Hébert-Johnson et al., 2018) (Liu et al., 2017).

To preserve the correct calibration of the EOP method described in Section 2.5.4, the Calibrated Equalized Odds (CEOP) post-processing method has been introduced by Pleiss et al. (2017). In this method, either TPR (Eq. 2.11) or FPR (Eq. 2.12) constraint is relaxed.

2.5.6 Satisfying ranked group fairness

The Fair Top-k ranking post-processing method (Zehlike et al., 2017) solves the problem of determining the k best candidates out of a larger pool of $n >> k$ individuals subject to group fairness criteria. This method differs from others in that it satisfies ranked group fairness, an extension of traditional group fairness.

This notion of ranked group fairness ensures that "the proportion of protected candidates in every prefix of the top-k ranking remains statistically above or indistinguishable from a given minimum" (Zehlike et al., 2017). Thus the main difference is that group fairness will be respected for any value of $k << n$.

2.5.7 Reduction approach

The reduction approach presented by Agarwal et al. (2018) proposes to reduce fair classification to a sequence of cost-sensitive classification problems. The solutions of these problems give a randomized classification subject to the requested fairness constraint with the minimum error. This approach allows to minimize the errors between the original and modified predictions.

2.5.8 Others

There are many other post-processing methods, of which here is a brief, exhaustive presentation. The Multiaccuracy Boost (MB) method adjusts predicted probabilities by combining multiple weak learner, each correcting errors of previous one (Kim et al., 2019). The FairScore Transformer (FST) takes advantage of the low dimensionality of the dual problem of standard probabilistic classifiers such as Logistic Regression or Gradient Boosting to solve the problem (Wei et al., 2020). The Wass-1 Postprocess (WPP) method enforces small Wasserstein distances between FPR and FNR distributions across groups (Jiang et al., 2019).
These methods offer diverse approaches to reduce bias and promote fairness in post-processing, with their effectiveness depending on specific implementation factors and dataset characteristics.

3 New post-processing method

3.1 Introduction

This master's thesis focuses on fair post-processing methods. As mentioned in Section 2.5, there are a huge number of such methods, including Massaging (Mass) (Calders et al., 2009), Reject Option-based Classification (ROC) (Kamiran et al., 2012), Equalized Odds (EOP) (Hardt et al., 2016), and many more. Each has its own strengths and weaknesses, and are therefore more appropriate in certain situations.

This section introduces a new post-processing method, *Optimal Swapping* (OS), and its *greedy* (OSG) version. These methods are designed to solve the problem of sometimes poor results in the least-square method of de Schaetzen (2021) when binary discrete decision are taken based on the observed probabilistic prediction of the classification model.

3.2 Least-Square post-processing method

This method is largely inspired by the work of de Schaetzen (2021)

A supervised classification algorithm uses a classifier to obtain a vector $\hat{y}$ of probabilistic predictions (scores). The goal is to find new predicted scores $\tilde{y}$ verifying the fairness constraints that are closest to the original scores provided by the classification model. To do so, a post-processing method is applied providing a vector $\tilde{\mathbf{y}}$ of new predicted scores. "This method minimizes the difference between new scores $\tilde{y}$ and original scores $\hat{y}$ under the constraint that the covariance (in absolute value) between the sensitive variables z and the new predictions is smaller than a small positive threshold ϵ. If $\tilde{y}_{i1}$ and $\hat{y}_{i1}$ are respectively the new and the original predicted probability for individual i to be positively predicted, the optimisation problem is the following" (Beghein & Kneip, 2022) :

$$
\begin{aligned}
\min_{\{\tilde{y}_{i1}\}} \quad & \sum_{i=1}^{n} (\tilde{y}_{i1} - \hat{y}_{i1})^2 \\
s.t. \quad & \frac{1}{n-1} \sum_{i=1}^{n} \dot{z}_i \tilde{y}_{i1} \leq \epsilon \\
& \frac{1}{n-1} \sum_{i=1}^{n} \dot{z}_i \tilde{y}_{i1} \geq -\epsilon \\
& \tilde{y}_{i1} \geq 0 \quad \forall i \in \{1, \ldots, n\} \\
& \tilde{y}_{i1} \leq 1 \quad \forall i \in \{1, \ldots, n\}
\end{aligned}
\tag{3.1}
$$

with $\dot{\mathbf{z}} = \mathbf{H}\mathbf{z}$ the centered sensitive vector. The sample covariance is :

$$
\text{cov}(\mathbf{z}, \tilde{\mathbf{y}}) = \frac{1}{n-1} \mathbf{z}^\top \mathbf{H} \tilde{\mathbf{y}} = \frac{1}{n-1} \dot{\mathbf{z}}^\top \tilde{\mathbf{y}} \tag{3.2}
$$

This optimization problem produces good demographic parity results in terms of class membership for new predicted scores when variables are continuous. But when taking binary discrete decisions based on the observed probabilistic predictions of this classification model, the new predictions that are supposed to be fair are no longer so. In order to guarantee that fairness constraints are respected when taking decisions, a new post-processing method, Optimal Swapping (OS), and its greedy version (OSG) are developed in this section.

3.3 General label switching approach

The OS and OSG methods are both part of a more general label switching approach that modifies certain model's predictions until this model is considered fair. This can be visualized as a large chessboard in which each point, representing a tuple with the number of positive label for protected and unprotected individuals, is considered either fair or unfair. The three parameters that influence this chessboard are (1) the fairness measure (Subsection 3.3.2), (2) the fairness threshold line (Subsection 3.3.3) and (3) ϵ and γ parameters (Subsections 3.3.4 and 3.3.5).

For the rest of this work, n_0 (resp. n_1) denotes the number of positively classified privileged (resp. unprivileged) individuals :

$$
n_0 = N(\tilde{y}^d = 1 \wedge z = 0) \qquad n_1 = N(\tilde{y}^d = 1 \wedge z = 1) \tag{3.3}
$$

3.3.1 Chessboard introduction & small example

The points on this chessboard are not all equidistant, as their relative position depends on the score of the individuals whose label is changed. Indeed, although each individual's label is binary, it is defined in terms of a continuous score between 0 and 1. Below 0.5, the label is negative (=0) and above it is rounded up to 1. If an individual's label is switched, the closer its score is to 0.5, the smaller the distance of the corresponding points on the chessboard. Here is an example to illustrate the concept.

Suppose an initial situation (blue dot on Figure 3.1) with two individuals, $i1$ and $i2$, that have a predicted score of respectively 0.4 and 0.7. According to the rule defined above, the first individual will have a negative prediction (0) because its score is below 0.5 and the second a positive prediction because its score is above the 0.5 threshold.
If the label of $i1$ is modified to become positive, the situation becomes that of the orange point in Figure 3.1 and the distance between the initial and current point is then calculated as the distance between $i1$'s score, 0.4, and the threshold 0.5. The same logic applies when calculating the distance between the initial situation and that of the i2 label swapped (green dot). This is how the distances between the different points on the chessboard are calculated.

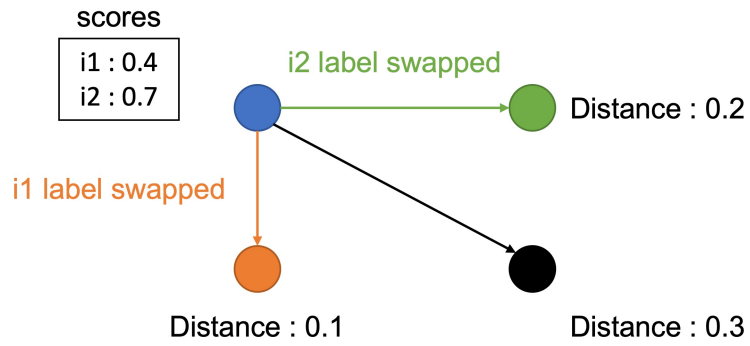

Fig. 3.1.: Chessboard with 2 individuals whose labels are swapped

3.3.2 Fairness measure

The metric used to measure the fairness of a point on the chessboard is the *Disparate Impact (DI)* (Eq. 2.9). The ratio of the rate of favorable outcomes received by the unprivileged group to the privileged group.

Before the post-processing phase, if a method is unfair, this ratio will be below 1. The goal is to swap labels so that DI becomes closer and closer to 1.

3.3.3 Fairness threshold line

From a given fairness measurement, e.g. $DI = 0.9$, a straight line can be drawn on the chessboard. This line corresponds to the following equation where the left term is the proportion of positively classified unprivileged on the positively classified privileged individuals :

$$\frac{n_1/N(z=1)}{n_0/N(z=0)} = 1 - \epsilon \tag{3.4}$$

with ϵ parameter defined in Subsection 3.3.4. This line is called the *Fairness Threshold Line (FTL)*.

3.3.4 Epsilon parameter

Achieving perfect fairness, $DI = 1$, is almost always impossible in practice. The distance between the perfect fairness line and fairness threshold line (FTL) mentioned above depends on the parameter $\epsilon \geq 0$. The closer this parameter is to 0, the closer the FTL will be to perfect fairness line and therefore the fewer chessboard points considered fair.

3.3.5 Gamma parameters

As previously mentioned, points on the chessboard are not equidistant since they depend on a certain distance of individual scores to 0.5. There are several ways to compute the total distance between two points on this chessboard. In this work, the two norms used are L1 ($d_1 + d_2$) and L2 ($\sqrt{d_1^2 + d_2^2}$). The parameter $\gamma \in [0, 1]$ is used to define the proportion of L1 and L2 used to calculate the total distance between 2 points :

$$\text{Distance} = \gamma * \sum_i d_i + (1 - \gamma) * \sqrt{\sum_i d_i^2} \tag{3.5}$$

On Figure 3.1 only L1 distance is used to compute the black dot. Indeed, the 0.3 distance is equal to the sum of the two other distances. Below is an example with two individuals per category, protected and unprotected.

The modification of a protected individual label leads to a horizontal shift of the situation on the chessboard, whereas that of an unprotected individual label results in a vertical shift. The distance between the initial situation (blue dot) and the

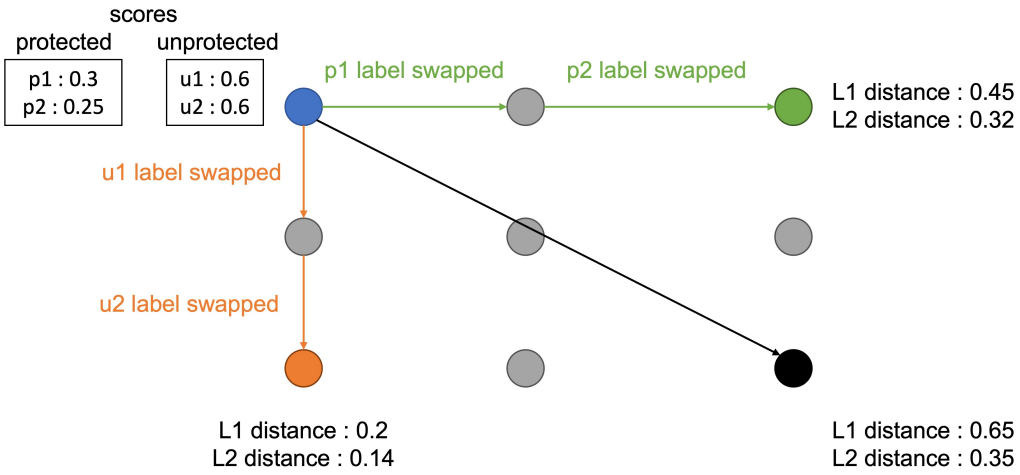

Fig. 3.2.: Chessboard with 4 individuals, 2 protected and 2 unprotected, whose labels are changed and the corresponding distances

final one (black dot) on Figure 3.2 depends on the γ parameter : distance $= \gamma * 0.65 + (1 - \gamma) * 0.35$.

3.3.6 Chessboard presentation

For a better visualization, Figure 3.3 shows an example of a situation incorporating the presented elements. The vertical axis corresponds to the number of positively classified privileged individuals, n_0, and the horizontal axis to the number of positively classified unprivileged individuals, n_1.

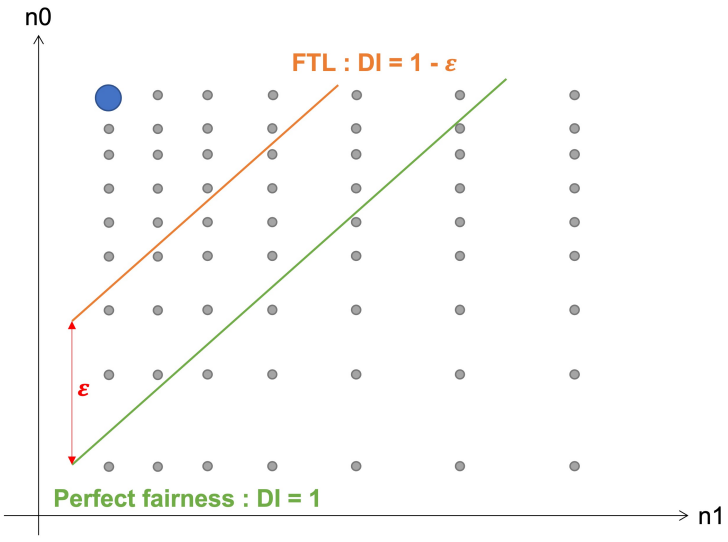

Fig. 3.3.: Chessboard presentation

The blue dot at top left represents the baseline situation before post-processing. Note that this situation is not fair, as its Disparate Impact is lower than that defined by

FTL, 1 - ϵ. In order to improve fairness, the various post-processing methods will have to switch labels, which means moving from gray point to gray point on the chessboard, until it situates below the FTL.

3.4 New post-processing methods

The 2 new post-processing methods presented below have the common goal of going below FTL to a situation considered fair.

3.4.1 Optimal Swapping (OS)

The OS method may be seen as an optimization problem with a single objective function and one constraint :

$$c^* = \min_{\{\tilde{y}^d\}} \quad \sum_{i=1}^{n} \frac{|\hat{y}_i - \tilde{y}_i^d|}{n} \tag{3.6}$$

$$s.t. \quad \frac{n_1/N(z=1)}{n_0/N(z=0)} \geq 1 - \epsilon \tag{3.7}$$

with $\tilde{y}_i^d$ being equal to 1 if original prediction $\hat{y}_i \geq 0.5$ and 0 otherwise.

The OS method satisfies this constraint (Eq. 3.7) by choosing a certain amount of individuals original predictions to swap among two swapping types. The first type concerns the swapping of protected individuals labels, sensitive variables $z = 1$, from unselected to selected (Eq. 3.8). While the second type concerns the swapping of unprotected individuals labels, sensitive variables $z = 0$, from selected to unselected (Eq. 3.9) :

$$\tilde{y}_i^d = 0 \rightarrow \tilde{y}_i^{d'} = 1 \quad \Rightarrow \quad \Delta c = c' - c^* = 2 * (0.5 - \hat{y}_i) \tag{3.8}$$

$$\tilde{y}_j^d = 1 \rightarrow \tilde{y}_j^{d'} = 0 \quad \Rightarrow \quad \Delta c = c' - c^* = 2 * (\hat{y}_j - 0.5) \tag{3.9}$$

In both cases, the objective value increases.

More formally, this is how the OS method works :

1. **Classify** all candidates (protected & unprotected) increasingly from closer score to 0.5 in two different lists. Candidates are unprotected (resp. protected) individuals that have a higher (resp. lower) score than 0.5.

2. **Identify** all fair candidates (yellow on Figure 3.4) corresponding to the points just below FTL and calculate their distance (as a function of the γ parameter) from the initial situation (blue dot).

3. **Select** the fair candidate with the lowest distance to initial situation (blue dot). As this means that the total accuracy of the model is the highest compared with the other fair candidates.

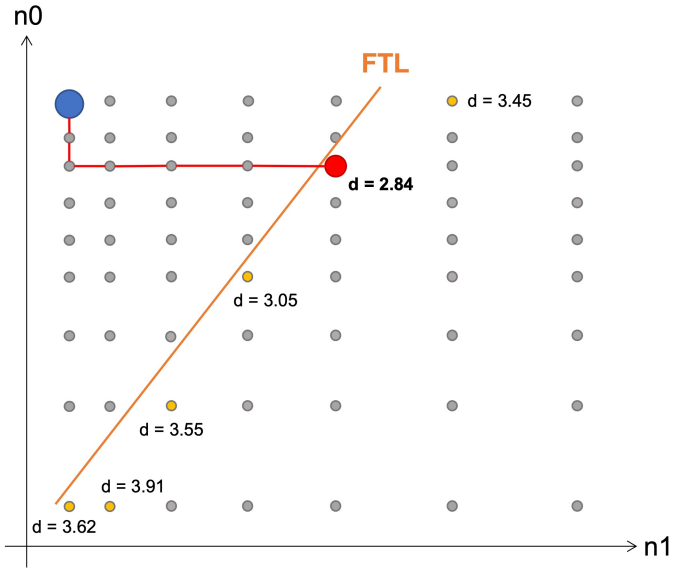

Fig. 3.4.: Optimal Swapping situation. The blue dot is the initial situation and the red dot the final fair one selected by the Optimal Swapping method

This method has the advantage to select the best point corresponding to a fair situation as accurate as possible.

3.4.2 Optimal Swapping Greedy (OSG)

The OSG method also resolves the (3.6)-(3.7) optimization problem but in a simpler way. Instead of comparing the distances of all the fair points below the FTL, this method moves from point to point until a fair situation is found, without worrying about whether it is optimal or not. This is the formal OSG method :

1. **Classify** all candidates (protected & unprotected) increasingly from closer score to 0.5 in a joint list. Candidates are unprotected (resp. protected) individuals that have a higher (resp. lower) score than 0.5.

2. **Chose**, **switch** its label and **remove** from the joint list the first candidate, whose score is closest to 0.5 and whose label has not yet been modified.

3. **Repeat** step 2.

4. **Stop** when it is in a situation below the FTL, meaning it is fair.

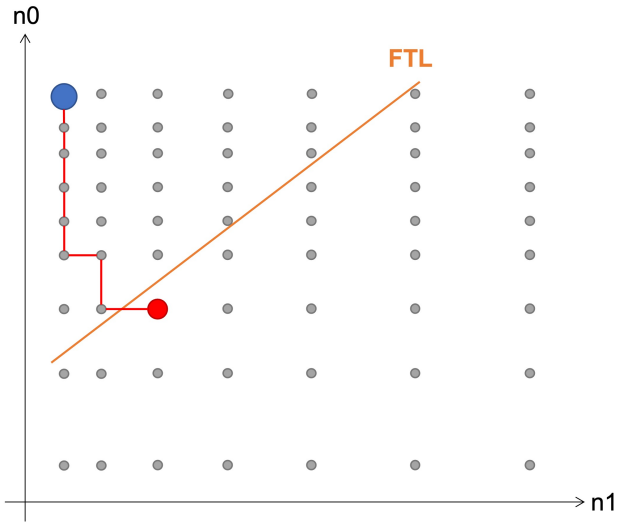

Fig. 3.5.: Optimal Swapping Greedy situation. The blue dot is the initial situation and the red dot the final fair one selected by the Optimal Swapping Greedy method

This method has the advantage of being faster than Optimal Swapping method, but its main drawback is that it regularly delivers a fair but sub-optimal solution. Indeed, on Figure 3.5 there are other fair points that are closer in distance to the initial blue point and therefore offer a more accurate while still fair solution.

3.5 Link with other methods

This section presents two post-processing methods based on a swapping technique, Optimal Swapping (OS) and its greedy version (OSG). These methods both classify individuals into two lists, protected and unprotected, based on their original predictions scores. This list separation and then classification is a distinctive characteristic also found in Massaging (Mass) (Subsection 2.5.2) and Reject Option-based Classification (ROC) (Subsection 2.5.1) methods. However, there are certain differences between the different ways of proceeding.

For the Massaging method, the labels of the first M individuals on top of each list are swapped. The value of M has been determined by Calders et al. (2009) and is worth

$$M = \frac{N(z=1) * n_0 + N(z=0) * n_1}{N(z=0) + N(z=1)} \tag{3.10}$$

The value of M does not depend on any parameters, neither ϵ nor γ, but guarantees that the final situation has a fairness measure $DI > 1$. Nevertheless, the final

situation is often suboptimal, as can be seen in Figure 3.6, where the point selected in red is very far from the FTL and will therefore have poorer accuracy than other fair points.

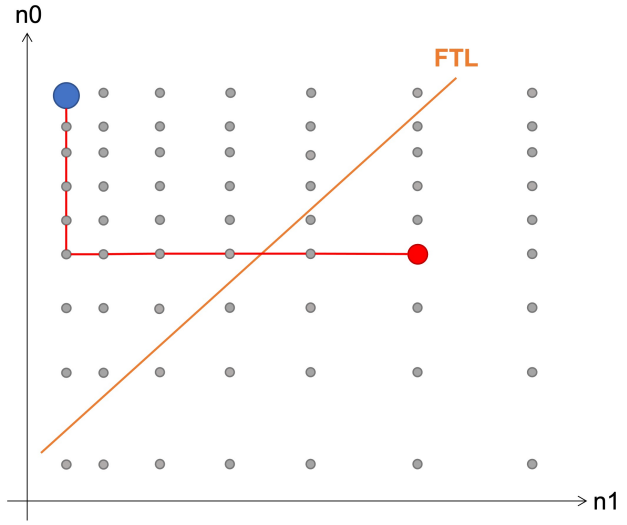

Fig. 3.6.: Massaging situation with $M = 5$. The blue dot is the initial situation and the red dot the final fair one selected by the Massaging method

For the ROC method, individuals are also classified in two lists, protected and unprotected. The main difference of ROC is that, unlike OS and OSG that only modify protected (resp. unprotected) predictions whose scores is below (resp. above) 0.5, here it swaps individuals in the two groups above and below the threshold classification. Graphically, this is represented on Figure 3.7 by a zone of uncertainty around 0.5 in which all protected individuals (green) are classified positively, those unprotected (red) are classified negatively and the predictions of individuals outside this zone remain unchanged.

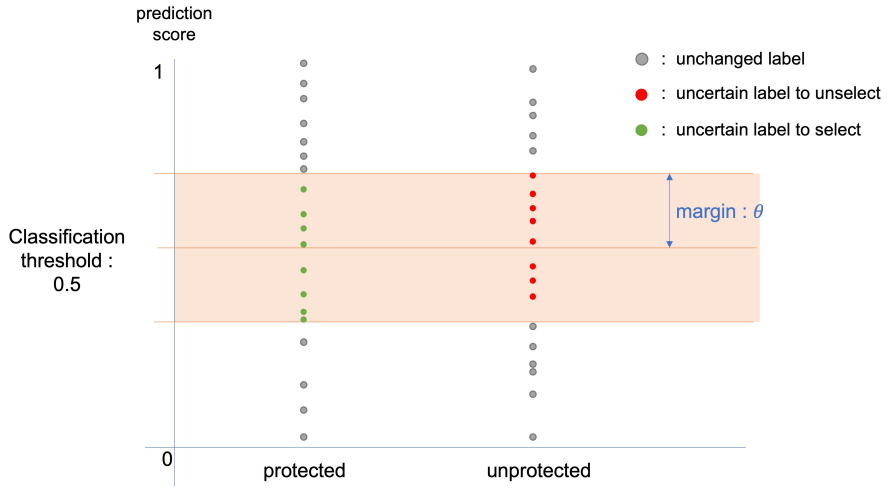

Fig. 3.7.: Reject Option-based Classification (ROC) situation with 0.5 classification threshold

3.6 Additional constraint

In many real-life cases, the number of individuals selected and therefore positively labeled is limited. Indeed, in the case of the loan bank, for example, even if it wants its lending process to be fairer, it nevertheless doesn't want to suddenly accept significantly more loan applications. The bank would like the number of people selected to be limited to a maximum value.

This limitation on the number of positively labeled people, P, corresponds to an additional constraint in the (3.6)-(3.7) optimization problem :

$$n_0 + n_1 \leq P \tag{3.11}$$

Visually, on the chessboard, this means that candidate points must be located to the right of a new line called $MaxSelect$ (purple on Figure 3.8).

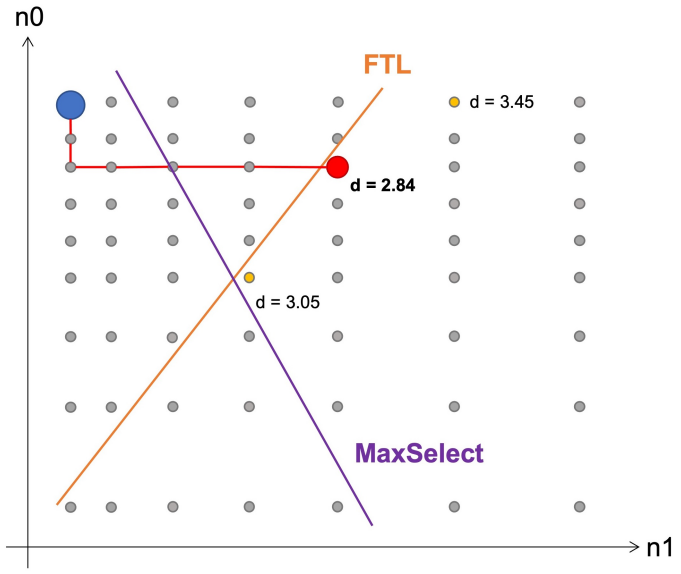

Fig. 3.8.: MaxSelect line in Optimal Swapping method situation. The blue dot is the initial situation and the red dot the final fair one selected by the Optimal Swapping method. Yellow points are fair candidates.

4 Implementation and methodology

4.1 Datasets

The experimentations are conducted on five datasets often used in the litterature when focusing on fairness in classification (D.Pessach & Shmueli, 2022) based on real-world data. The following explanation of the different datasets is based on the master thesis work of Beghein and Kneip (2022)

4.1.1 German Credit

The German Credit dataset is a widely used dataset in the field of credit risk assessment used to predict the solvability of individuals and determine their likelihood of defaulting on a loan.

This dataset contains 1,000 individuals and the target variable is binary, 1 if the person is considered has a good client, 0 otherwise. The sensitive attribute (Z) is equal to 1 if the client is under the age of 25, this represents nearly 20% as seen on Table 4.1.

	Class 0	Class 1
Target (Y)	0.3	0.7
Age (Z)	0.81	0.19

Tab. 4.1.: A priori probabilities of Y and Z on German Credit dataset

4.1.2 COMPAS

The COMPAS dataset is derived from real data collected in the USA and is used to predict the risk of criminals re-offending. The dataset has been anonymized to protect privacy and contains details such as demographics, criminal history and factors used in pretrial assessments.

This dataset contains 6,150 interesting individuals for this work and the target variable is binary, 0 if the person has re-offended within 2 years and 1 otherwise. Both classes in the dataset have an equal chance of occurring, with a 47% probability of recidivism (as shown in Table 4.2). The dataset includes a sensitive attribute, ethnicity, which is represented as a binary variable: 1 for African-American and 0 for Caucasian.

	Class 0	Class 1
Target (Y)	0.4662	0.5338
Ethnicity (Z)	0.3991	0.6009

Tab. 4.2.: A priori probabilities of Y and Z on COMPAS dataset

4.1.3 Law

The Law dataset is derived from a survey conducted across 163 law schools in the U.S. It focuses on a binary classification task: predicting whether a candidate will pass the bar exam (1) or not (0) based on their admission record. The two classes of Y are very unbalanced (95% for 1 and 5% for 0 as seen on Table 4.3) which can considerably reduce the model's performances. There is 20,798 individuals in the dataset and the sensitive attribute (Z) is the ethnicity, 1 for non-white and 0 for white individuals.

	Class 0	Class 1
Target (Y)	0.0501	0.9499
Ethnicity (Z)	0.8410	0.1590

Tab. 4.3.: A priori probabilities of Y and Z on Law dataset

4.1.4 Bank

The Bank dataset contains 45,211 individuals and is used to predict whether a customer will subscribed to a term deposit (1) or not (0). This variable is indicative of customer response to the bank's marketing efforts. The sensitive variable is the age, Class 1 contains individuals between the ages of 25 and 60 and Class 0 all the others (see on Table 4.4).

	Class 0	Class 1
Target (Y)	0.8830	0.1170
Age (Z)	0.0442	0.9558

Tab. 4.4.: A priori probabilities of Y and Z on Bank dataset

4.1.5 Adult

The Adult dataset, also know as Census Income comes from the UCI machine learning repository (Dua & Graff, 2017) and is composed of 45,222 instances. This dataset is used to see whether women, sensitive variable (Z) equal to 1, receive the same salary as men. The target variable (Y) is the annual income, 1 if greater than 50k$ and 0 otherwise.

	Class 0	Class 1
Target (Y)	0.8340	0.1660
Gender (Z)	0.6750	0.3250

Tab. 4.5.: A priori probabilities of Y and Z on Adult dataset

4.2 Dataset's fairness

To get a better idea of the unfairness of these different datasets, the *Disparate Impact* (DI) is calculated for each situation and shown in Table 4.6. A ratio of less than 1 means that class 1 is discriminated. As a reminder,

$$DI = \frac{p(Y=1|Z=1)}{p(Y=1|Z=0)} \tag{4.1}$$

It can be seen on Table 4.6 that all datasets have a disparate impact of less than 1, reflecting unfairness in their data. Bank and Adult datasets seems to be the most unfair datasets but this can change after post-processing techniques are applied in next section.

	German	**Compas**	**Law**	**Bank**	**Adult**
$p(Y=1\|Z=1)$	0.5789	0.4857	0.8549	0.1060	0.0757
$p(Y=1\|Z=0)$	0.7284	0.6064	0.9680	0.3550	0.2095
Disparate Impact	**0.7948**	**0.8009**	**0.8831**	**0.2985**	**0.3612**

Tab. 4.6.: Disparate Impact of the datasets

4.3 Baseline Classifiers

Post-processing methods are evaluated on the five datasets presented in Section 4.1, as well as on six different classifiers (Raschka & Mirjalili, 2019) (Hastie et al., 2009). On the one hand, this makes it possible to compare their performance and analyze which classifier offers the best results. On the other hand, testing the methods on

different classifiers ensures that they are providing good results no matter which classifier is used.

4.3.1 Logistic Regression

Logistic Regression (LogReg) is a linear classification algorithm used for binary or multi-class classification problems. It models the relationship between the independent variables and the probability of a certain outcome by using a logistic function (Murphy., 2012).
In order to prevent overfitting in the model a penalty term might be added, it is not the case in this work. If the algorithm has not converged after reaching 2,000 iterations, it will stop and return the current solution.

4.3.2 Decision Tree

Decision Tree builds a tree-like model by splitting the data based on different attributes, creating branches that best separate the classes. Each leaf node represents a class prediction.
The maximum depth of this tree is set to 7 and the minimum number of samples that must be present in a leaf node for the splitting process to continue is 10. To identify the best split, the Gini index is computed to measure node impurity (Alpaydin, 2020) (Tan et al., 2018).

4.3.3 Random Forest

Random Forest is an ensemble learning method that combines multiple decision trees. It creates a set of decision trees on random subsets of the data and combines their predictions to make the final prediction handling high-dimensional data well (Tan et al., 2018). Parameters of each tree are the one presented above and the maximum number of tree included is 10,000.

4.3.4 Bagging

Bagging is an ensemble learning technique which creates multiple subsets of the original dataset and trains individual classifiers on each subset. The final prediction is determined by combining the predictions of all the classifiers (Tan et al., 2018). The base estimator used to train individual estimators is the decision tree with the same parameters as presented above.

4.3.5 Boosting

Boosting classification algorithm combines weak learners to create a strong classifier. It trains the weak learners sequentially, with each subsequent learner focusing on the instances that the previous ones misclassified (Tan et al., 2018) (Murphy., 2012). Boosting algorithms can improve overall prediction accuracy. In this work the boosting algorithm used is Gradient Boosting.

4.3.6 Neural Network

Neural Network (NN) is a machine learning model inspired by the structure of the human brain. It consists of interconnected nodes (neurons) organized into layers. It can handle large amounts of data but requires careful architecture design and training (Tan et al., 2018).
The hidden layers architecture consists of 3 layers which contain respectively 64, 32 and 16 neurons. Moreover, this method is limited to 20,000 iterations.

4.4 Experimental methodology

This section is largely inspired by the thesis of Beghein and Kneip (2022).

A 5-fold cross-validation technique is employed to build the models. For a given dataset, it is divided into five subsets of equal size, with each subset serving as the test set once while the remaining four subsets are combined to form the training set. This approach ensures that each data point is used for both training and testing purposes.

During each fold of the cross-validation process, the results obtained from the model are stored and recorded. To evaluate the performance of the model across different parameter values, such as γ and ϵ, the micro-average is computed. The micro-average calculates the overall performance metric by considering the aggregate results of all the folds. This approach allows for robust evaluation and comparison of different parameter settings, aiding in the selection of the optimal configuration for the given task.

4.4.1 Experimental tests

The baseline models for each classifier are those that predict the target classes before post-processing techniques are applied. They serve as a benchmark for evaluating the different techniques applied afterwards.

Some methods modify only one of the two parameters, ϵ or γ, while others vary both. Table 4.7 shows the different scenarios. When modified, the ϵ parameter varies from 0.5 to 0.005 with progressively smaller steps and parameter γ takes values between 0 and 1 in steps of 0.25.

	OS	OSG	Massaging	ROC
Epsilon	✓	✓	✗	✓
Gamma	✓	✗	✗	✗

Tab. 4.7.: Modified parameters in each of the 4 comparison methods

4.4.2 Comparison metric

In order to compare the predictions of different folds, it is necessary to use an appropriate measure. *Accuracy* is a good way of measuring a model's efficiency, but it does not reflect the model's fairness and this metric can be biased if labels are not balanced. On the other hand, the fairness measure used in each method is the *Disparate Impact* but this says nothing about the quality of predictions.

A tradeoff between accuracy and fairness must therefore be found. To this end, the *F1-score* is used to measure the performance of the different methods.

$$F1 = \frac{2 * (1 - |DP|) * Accuracy}{(1 - |DP|) + Accuracy} \tag{4.2}$$

with DP the Demographic Parity, the difference in the rate of favorable outcomes received by unprivileged group to the privileged group. This measure increases proportionally with Accuracy and inversely with Demographic Parity.

4.4.3 Comparison statistical tests

After running the experimental tests, results are available for each method, depending on the dataset, the ϵ and γ parameters and the fold used. In order to compare the performance of different methods, the best parameters must be selected for each method. First, the best ϵ and γ parameters for each classifier are selected via a

Friedman-Nemeyni test (Friedman, 1937) (Nemenyi, 1963). Then, the best classifier of each method is chosen again with a Friedman-Nemenyi test. Finally, the methods are compared using Friedman-Nemenyi and Wilcoxon tests (Wilcoxon, 1945). As explained above, a parameter combination is considered best on the basis of its F1-score.

Figure 4.1 shows the structure of the experimental results files. The very large number of parameter/classifier combinations explains why statistical tests are performed in several stages rather than all at once.

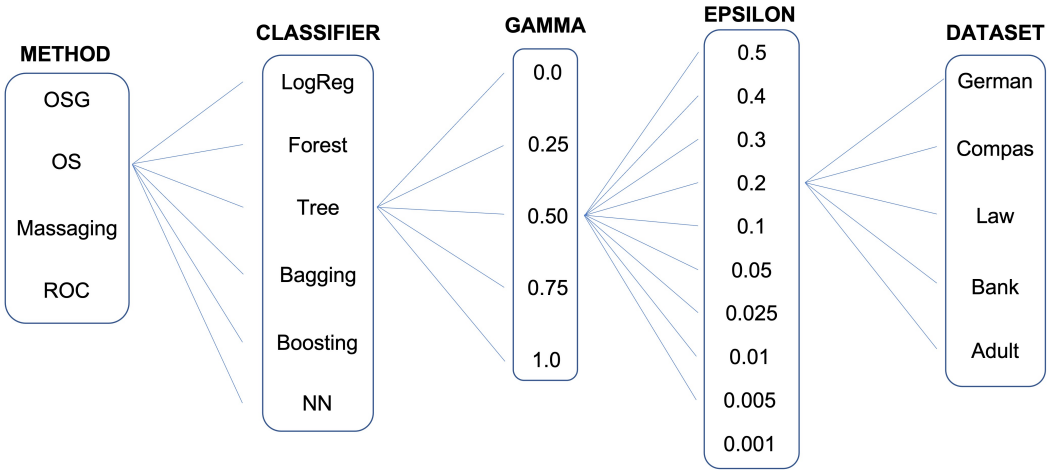

Fig. 4.1.: Structure of experimental results files

The three statistical tests are performed multiple times at different stages (parameters/classifiers) and they make no assumptions on the distribution of parameters across the population samples. Especially, they do not require the differences between the two random variables to be normally distributed (Demsar, 2006). However, they do require that the data samples are independent of each other. This condition is not satisfied since the results are computed on the same data.
As a result, these tests can only be used for informational purposes and no completely reliable conclusions can be drawn. However, they can still provide some evidence either supporting or opposing some ideas, which is valuable for this exploratory study.

Friedman test

The Friedman test, introduced by Friedman (1937), is a non-parametric alternative to the repeated-measures ANOVA. The test assigns ranks to the parameters/classifiers in descending order of their performance for each dataset. Then, the test compares the average ranks of parameters/classifiers defined by $R_j = \frac{1}{N} \sum_{i=1}^{N} r_i^j$ with r_i^j being the rank of the j^{th} out of k results of the N datasets. The number of k results vary

but is equal to the number of parameters or classifiers times the number of folds (5). The Friedman statistics is therefore :

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

It is distributed according to a chi-square distribution with $k-1$ degrees of freedom when N and k are large enough ($N > 10$ and $k > 5$) which is not the case here since there is only five datasets (N) (Demsar, 2006).

Nemenyi test

If the null hypothesis of the Friedman test is rejected (p-value is below a threshold α) then a second test developed by Nemenyi (1963) can be performed. It is used to compare all parameters/classifiers to each other. The performance of two parameters/classifiers is different if their mean ranks vary by at least the critical distance (CD) (Demsar, 2006) :

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

with q_α based on the studentised range statistic at a significance level α divided by the square root of 2 (Demsar, 2006).

Wilcoxon signed-rank test

The Wilcoxon signed-rank test, developed by Wilcoxon (1945), is a non-parametric version of the paired t-test. It determines if two algorithms perform equally well. It does this by ranking and comparing the performance differences between two parameters/classifiers for each dataset, regardless of their sign. R^+ is defined as the sum of the ranks of the datasets on which the second algorithm outperforms the first, R^- is the sum of the ranks for the opposite scenario. In addition, the ranks of ties are split among both sums (Demsar, 2006)(Zar, 2010). Variables are computed as follow :

$$R^+ = \sum_{d_i>0} \text{rank}(d_i) + \frac{1}{2} \sum_{d_i=0} \text{rank}(d_i) \qquad R^+ = \sum_{d_i<0} \text{rank}(d_i) + \frac{1}{2} \sum_{d_i=0} \text{rank}(d_i) \tag{4.5}$$

with d_i the difference of performance between the 2 parameters/classifiers on the i^{th} dataset. The minimum of the two sums, $T = min(R+, R-)$, is used to calculate the statistic test :

$$z = \frac{T - \frac{1}{4}N(N+1)}{\sqrt{\frac{1}{24}N(N+1)(2N+1)}} \tag{4.6}$$

5 Experimental Results and discussions

5.1 Introduction

In order to compare the different methods, a selection of the best ϵ and γ parameters and the best classifier for each of them has to be done. To do so, the best settings will be selected in this order : ϵ, γ and classifier.

It is important to note that the classification threshold has been set at 0.5 for all methods. This could affect their performances, particularly the ROC method which, in its original version (Kamiran et al., 2012), has a variable classification threshold.

5.2 Comparison between epsilon parameter

The first step is to select the best ϵ parameter for each tuple (γ, classifier, method). Rather than performing a statistical test for every combination, it is more interesting to observe how the F1-score evolves as the ϵ parameter decreases. Indeed, there are more than 2,000 parameter combinations of ϵ and tuples. If there is a clear correlation between the F1-score and the ϵ with a peak at some point, a large number of unnecessary statistical tests can be avoided.

F1-scores have been normalized between 0 and 1 for each dataset to make it easier to visualize the effect of ϵ parameter. Figure 5.1 clearly shows that the closer ϵ is to 0, the higher the F1-score for every dataset. The same conclusions can be reached for the other γ values, classifiers and methods.

However, it is important to note that there is a higher F1-score peak on the German dataset. This observation occurs in certain combinations (γ, classifier, method) but not systematically. It is probably due to instabilities in the German-folds results due to the small size of this dataset.

Consequently, the smallest ϵ value tested, 0.001, is chosen as the optimal value for the rest of this work.

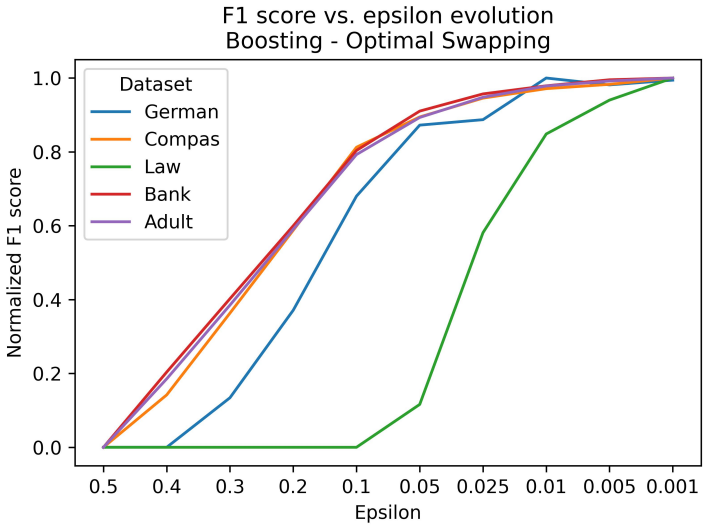

Fig. 5.1.: Normalized F1-score depending on ϵ values for each dataset. Method : Optimal Swapping - Classifier : Boosting - γ parameter : 0.0

5.3 Comparison between gamma parameter

The second step consists in selecting the best γ parameter for each pair (classifier, method). Since only the *Optimal Swapping* method has a parameter γ, it is the only method tested in this section.

A Friedman test is performed on each of the classifiers to identify whether there is a significant difference between the γ or not. The threshold below which the p-value is considered significant is $\alpha = 0.05$.

	LogReg	Forest	Tree	**Bagging**	Boosting	**NN**
Friedman p-value	0.8754	0.9927	0.1591	**0.0124**	0.0984	**0.0009**
Best γ selected	1.0	0.5	0.5	0.75	0.5	0.25

Tab. 5.1.: Friedman test p-value & best γ selected for 6 classifiers. Method : Optimal Swapping - ϵ parameter : 0.001

Since the independence hypothesis is not fully respected (Subsection 4.4.3) it is possible that some results may not be significant and should therefore be taken with caution. Table 5.1 shows the Friedman test p-value for each classifier. Only two of them have γ parameters significantly different : Bagging and NN classifiers. A Nemenyi test is then performed to identify where these differences lie (see Figure 5.2). For other classifiers, the best γ is nevertheless selected, even if it is not significant.

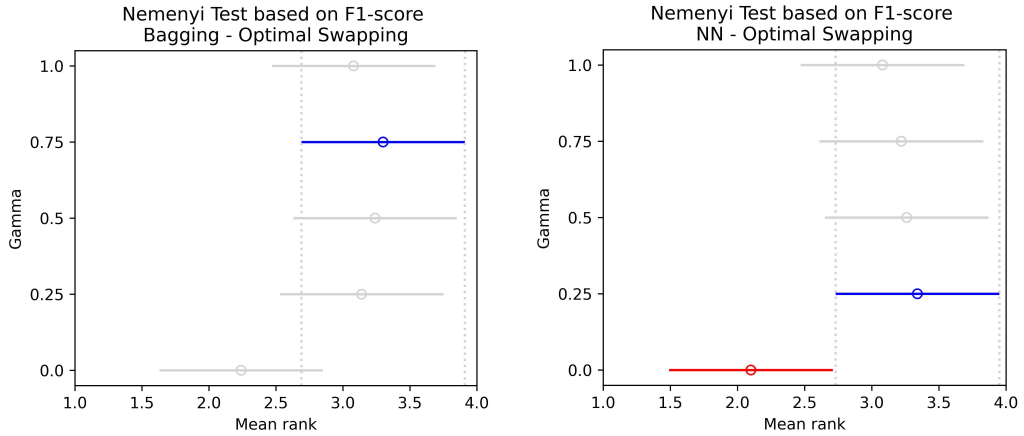

Fig. 5.2.: Mean ranks and 95% Nemenyi confidence intervals for 5 γ values according to the F1-score. Blue horizontal line is the best γ and red one is significantly worse. Classifier : Bagging (left) & NN (right)

5.4 Comparison between classifiers

At this stage, all pairs (methods, classifier) are associated with their optimal ϵ/γ parameters. The next step finds the best classifier for each of the four methods compared. Again, a Friedman test is performed on each method to identify if there are classifiers that are significantly better than others. The same threshold of $\alpha = 0.05$ is applied.

	Optimal Swapping	**Optimal Swapping Greedy**	**Massaging**	**ROC**
Friedman p-value	**1.023e-7**	**6.569e-10**	**1.049e-8**	**8.171e-8**
Best classifier selected	Boosting	Bagging	Boosting	Forest

Tab. 5.2.: Friedman test p-value & best classifier selected for 4 methods. ϵ parameter : 0.001

Table 5.2 resumes the Friedman p-values and best classifiers selected for each method. In this case, since every p-value is smaller than 0.05 it means that each method has significantly different results between its classifiers. Consequently, Nemenyi and Wilcoxon signed-rank tests are performed to identify which classifiers are significantly different. Nemenyi tests are visible on Figures 5.3-5.4 and Wilcoxon results on Tables A.1-A.4.

It is important to specify that the performances of the NN are significantly lower than that of other classifiers even before a post-processing method is applied. In fact, parameters of this classifier need to be tuned in order to obtain optimal performances but these are arbitrary fixed values in this case. Moreover, this classifier works better on big datasets which is not always the case here.

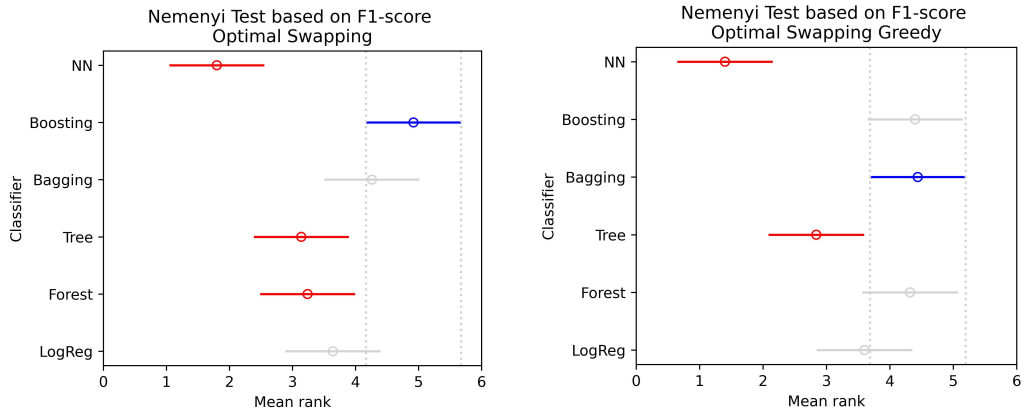

Fig. 5.3.: Mean ranks and 95% Nemenyi confidence intervals for 6 classifiers according to the F1-score. Blue horizontal line is the best classifier and red ones are significantly worse. Methods : Optimal Swapping (left) & Optimal Swapping Greedy (right)

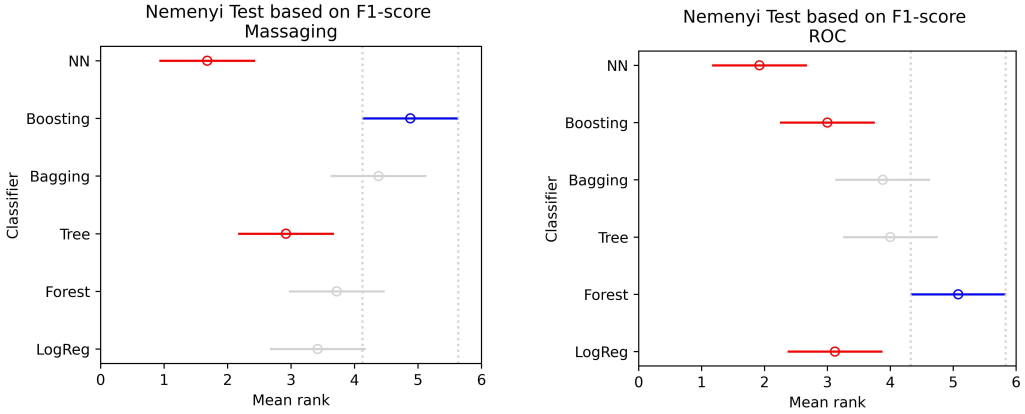

Fig. 5.4.: Mean ranks and 95% Nemenyi confidence intervals for 6 classifiers according to the F1-score. Blue horizontal line is the best classifier and red ones are significantly worse. Methods : Massaging (left) & ROC (right)

For the three methods OS, OSG and Massaging, the ranking of the best classifiers is quite similar. The two best are always either Bagging or Boosting, the next two are either Forest or LogReg and the last two are respectively Tree and NN. By taking into account the results of the Wilcoxon signed-rank tests, there are significant differences between the methods.

The new method, Optimal Swapping, has Boosting as best classifier significantly outperforming every others except Bagging. Then in second place Bagging classifier has significant differences with all the others except LogReg. Then come respectively Forest, LogReg and Tree which all significantly outperform NN classifier (see Table A.1)

For the greedy version of the new method, Optimal Swapping Greedy, Bagging and Boosting are the two best classifiers and they are significantly outperforming Tree and NN. There is no significant differences between Forest, LogReg and Tree but all three have better significant results than NN. (see Table A.2)

For the Massaging method, Boosting and Bagging classifiers are respectively the first and second best ones. There is no significant difference between the two but their results both outperforms significantly those of NN, Forest, LogReg and Tree. As for the following three classifiers, there are no significant differences between them but rather between each of them and NN. (see Table A.3)

The last method, ROC, gives very different results from the others (see Table A.4). Indeed, the classifier with the best rank is Forest, which has significant differences in ranks with all the other classifiers. Then comes the Tree classifier only significantly outperforming NN and in third place the Bagging one which has significant positive differences with LogReg, Boosting and NN. The three last classifiers, respectively LogReg, Boosting and NN have all three significant differences.

5.5 Comparison between methods

The final step is to compare the methods themselves. Table 5.3 summarizes the best parameters and classifiers per method as well as the corresponding mean F1-score. Since only Optimal Swapping has a γ parameter it is the only method whose corresponding value is present in Table 5.3. The same reason explains why Massaging method has no value for best ϵ parameter. Again, a Friedman test ($\alpha = 0.05$) is performed that gives a p-value of $1.105e-6$, meaning the results are significantly different.

	Optimal Swapping	**Optimal Swapping Greedy**	**Massaging**	**ROC**
Best ϵ parameter	0.001	0.001	✗	0.001
Best γ parameter	0.5	✗	✗	✗
Best classifier	Boosting	Bagging	Boosting	Forest
Mean F1-score	0.89833	0.89573	0.89691	0.86389

Tab. 5.3.: Best classifiers and parameters for 4 methods.

Before any statistical test is performed, it can be seen on Table 5.3 that the average F1-scores of the three methods OS, OSG and Massaging are very close and that the ROC is more than 3% lower. A Nemenyi test is then completed revealing that the *Optimal Swapping* method is significantly better than its *greedy* version and the *ROC* method (see on Figure 5.5). Even if it is not significant, the *Massaging* method is

also worse in terms of F1-score than the *Optimal Swapping* one on the investigated datasets.

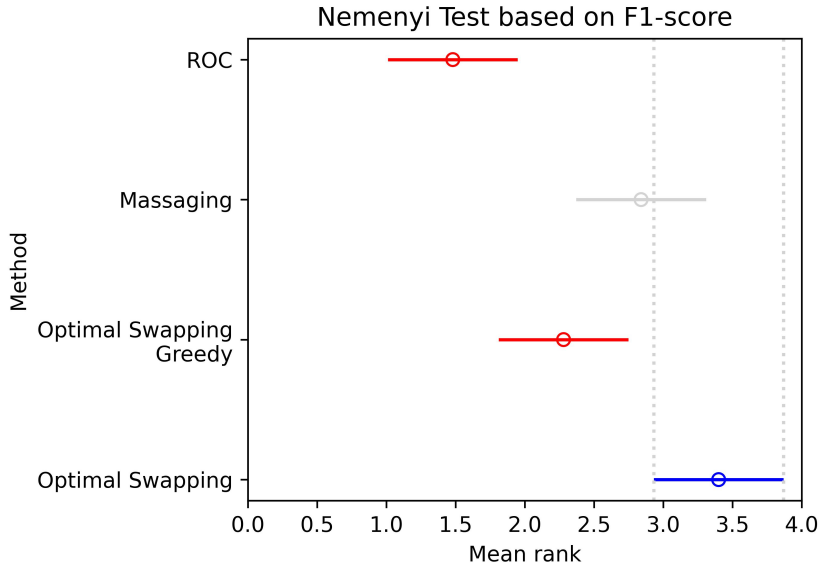

Fig. 5.5.: Mean ranks and 95% Nemenyi confidence intervals for 4 post-processing methods according to the F1-score. Blue horizontal line is the best method and red ones are significantly worse.

Finally, non-parametric Wilcoxon signed-ranks tests are performed to compare the methods two by two. Table 5.4 shows that all the methods are significantly different ($\alpha < 0.05$) except between methods *Optimal Swapping Greedy* and *Massaging*. These tests seem to confirm all the significant differences of the Nemenyi test (Figure 5.5) but also reveal other differences. Indeed, they find that *OS* performs significantly better than the *Massaging* method, which itself obtains results significantly better than those of *ROC*. Moreover, the *OSG* method significantly outperforms the *ROC* one.

	OS	OSG	Massaging	ROC
OS	1.0	**1.603e-2**	**1.767e-2**	**6.451e-5**
OSG	**1.603e-2**	1.0	6.934e-2	**5.133e-5**
Massaging	**1.767e-2**	6.934e-2	1.0	**7.224e-5**
ROC	**6.451e-5**	**5.133e-5**	**7.224e-5**	1.0

Tab. 5.4.: P-values of the Wilcoxon signed-ranks tests comparing the 4 post-processing methods.

5.6 Comparison of speed

Although the *Optimal Swapping* method is the best, it is also interesting to compare the speed of prediction calculation for each method. This includes predictions for

each of the combinations of five folds, six classifiers and, where applicable, five γ-values. Calculation times for all epsilons are not taken into account.

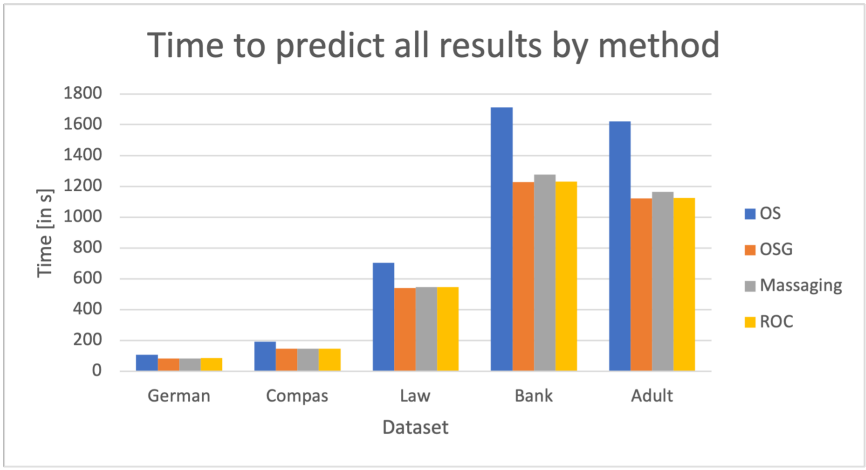

Fig. 5.6.: Speed of each method to predict each combinations of folds, ϵ and γ parameters and classifiers for each dataset.

Figure 5.6 shows that the best method, *Optimal Swapping*, is actually the slowest one. Nevertheless, it should not be forgotten that this method varies its γ parameter while the others do not. Overall, the four methods operate at speeds in the same order of magnitude.

6 Conclusion and further work

Conclusion

This thesis provides two new post-processing methods to address the problem of discrimination against minority groups in machine learning supervised classification predictions.

In the state of the art, it has been seen that discrimination can come from many different types of bias and can be measured with various fairness metrics. To combat this group discrimination in favor of the privileged over the non-privileged, a pipeline of techniques is available, namely pre-processing, in-processing, and post-processing fairness techniques. This work focuses on fair post-processing techniques.

The post-processing method based on least-square optimization problem developed by de Schaetzen (2021) provides very good results in terms of Demographic Parity class membership when predictions are continuous. However, this method performs poorly when binary decisions are taken based on continuous scores. In order to solve this problem, an Optimal Swapping post-processing method and its greedy version have been developed and compared to other similar techniques.

The two post-processing methods used for comparison are the Massaging one developed by Calders et al. (2009) and the Reject Option-based Classification presented by Kamiran et al. (2012). They are both part of a more general swapping method where individuals are sorted into 2 lists, according to their prediction scores, then their labels are swapped following certain rules specific to each method.

The comparison of techniques was based on five datasets (Section 4.1) and six classifiers (Section 4.3) in order to be as general as possible. Moreover, the classification threshold for each method is fixed at 0.5. The results show that the Optimal Swapping method significantly outperforms the other three, and that its greedy version is significantly better than the ROC one.

Further work

To confirm the good results of the Optimal Swapping method, it should be performed on other datasets and compared to other post-processing methods. It might also be

interesting to use other fairness measures and see if the Optimal Swapping method still outperforms the others.

The Optimal Swapping method includes a parameter γ which allows the distance between points to be varied according to a norm L1 and L2. An improvement would be to generalize this parameter to be used with other distance norms such as L1.5, L3, L4,...

Finally, it would be interesting to explore the idea presented in Section 3.6 of a limit on the maximum number of candidates to be selected. Indeed, in many real-life situations such constraints are taken into account when making decisions.

Bibliography

Agarwal, A., Beygelzimer, A., Dudík, M., Langford, J., & Wallach, H. (2018). A reductions approach to fair classification. (Cit. on p. 14).

Alpaydin, E. (2020). *Introduction to machine learning*. MIT Press. (Cit. on p. 28).

Anderson, H., Boodhwani, A., & Baker, R. (2019). Assessing the fairness of graduation predictions. *Proceedings of the 12th International Conference on Educational Data Mining*, 488–491 (cit. on p. 4).

Baeza-Yates, R. (2018). Bias on the web. *Communications of the ACM*, 54–61 (cit. on p. 4).

Barocas, S., Hardt, M., & Narayanan, A. (2019). *Fairness and machine learning: Limitations and opportunities*. fairmlbook.org. (Cit. on p. 3).

Beghein, E., & Kneip, A. (2022). *Fairness in supervised classification: Investigation of three different techniques*. (Cit. on pp. 15, 25, 29).

Bellamy, R., Dey, K., Hind, M., Hoffman, S., Houde, S., Kannan, K., Lohia, P., Martino, J., Mehta, S., Mojsilovic, A., Nagar, S., Ramamurthy, K., Richards, J., Saha, D., Sattigeri, P., Singh, M., Varshney, K., & Zhang, Y. (2018). *Ai fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias*. (Cit. on p. 7).

Bishop, C. (2006). *Pattern recognition and machine learning*. Springer. (Cit. on p. 4).

Calders, T., Kamiran, F., & Pechenizkiy, M. (2009). Building classifiers with independency constraints. *IEEE International Conference on Data Mining Workshops*, 13–18 (cit. on pp. 12, 15, 22, 41).

Calders, T., & Verwer, S. (2010). Three naive bayes approaches for discrimination-free classification. *Data Min Knowl Disc*, *21*, 277–292 (cit. on p. 13).

Caliskan, A., Bryson, J., & Narayanan, A. (2017). Semantics derived automatically from language corpora contain human-like biases. *Science*, *356*, 183–186 (cit. on p. 1).

Carvalho, D., Pereira, E., & Cardoso, J. (2019). Machine learning interpretability: A survey on methods and metrics. *Electronics* (cit. on p. 8).

Cozarenco, A., & Szafarz, A. (2018). Gender biases in bank lending: Lessons from microcredit in france. *Journal of Business Ethics*, *147*, 631–650 (cit. on pp. 1, 5).

Cummings, M. (2004). *Automation bias in intelligent time critical decision support systems*. American Institute of Aeronautics; Astronautics. (Cit. on p. 5).

Dawid, A. (1982). The well-calibrated bayesian. *Journal of the American Statistical Association*, *77*, 605–610 (cit. on p. 13).

Demsar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research* (cit. on pp. 31, 32).

de Schaetzen, C. (2021). *Increasing fairness in supervised classification: A study of simple decorrelation methods applied to the logistic regression*. (Cit. on pp. 15, 41).

D.Pessach & Shmueli, E. (2022). A review on fairness in machine learning. *ACM Computing Surveys* (cit. on p. 25).

Dua, D., & Graff, C. (2017). *Uci machine learning repository.* http://archive.ics.uci.edu (accessed: 07.07.2023). (Cit. on p. 27)

Dunkelau, J. (2019). *Fairness-aware machine learning: An extensive overview*. (Cit. on p. 6).

Dwork, C., Hardt, M., Pitassi, T., Reingold, O., & Zemel, R. (2011). Fairness through awareness. *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, 214–226 (cit. on p. 8).

Faber, J. (2017). Segregation and the geography of creditworthiness: Racial inequality in a recovered mortgage market. *Housing Policy Debate*, *28*, 215–247 (cit. on pp. 1, 6).

Friedman, M. (1937). The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association*, *32*, 675–701 (cit. on p. 31).

Hall, M., van der Maaten, L., Gustafson, L., Jones, M., & Adcock, A. (2022). A systematic study of bias amplification. *ArXiv* (cit. on p. 5).

Hardt, M., Price, E., & Srebro, N. (2016). Equality of opportunity in supervised learning. *Proceedings of the 30th International Conference on Neural Information Processing Systems*, 3323–3331 (cit. on pp. 7, 13, 15).

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning*. Springer. (Cit. on p. 27).

Hébert-Johnson, U., Kim, M., Reingold, O., & Rothblum, G. (2018). Calibration for the (computationally-identifiable) masses. (Cit. on p. 13).

Hellström, T., Dignum, V., & Bensch, S. (2020). *Bias in machine learning – what is it good for?* (Cit. on p. 4).

Hutchinson, B., & Mitchell, M. (2019). 50 years of test (un)fairness. *Proceedings of the Conference on Fairness, Accountability, and Transparency* (cit. on p. 13).

IBM. (n.d.). *Ai fairness 360*. https://aif360.mybluemix.net/ (accessed: 02.05.2023). (Cit. on p. 9)

Inc, N. (2019). *Practitioner's guide to compas core*. (Cit. on p. 6).

Jiang, R., Pacchiano, A., Stepleton, T., Jiang, H., & Chiappa, S. (2019). Wasserstein fair classification. *Proceedings of The 35th Uncertainty in Artificial Intelligence Conference*, 862–872 (cit. on p. 14).

Kamiran, F., Karim, A., & Zhang, X. (2012). Decision theory for discrimination-aware classification. *IEEE 12th International Conference on Data Mining*, 924–929 (cit. on pp. 7, 12, 15, 33, 41).

Kim, M., Ghorbani, A., & Zou, J. (2019). Multiaccuracy: Black-box post-processing for fairness in classification. *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, 247–254 (cit. on p. 14).

Kleinberg, J., Ludwig, J., Mullainathan, S., & Rambachan, A. (2018). Advances in big data research in economics algorithmic fairness (cit. on p. 7).

Liu, Y., Radanovic, G., Dimitrakakis, C., Mandal, D., & Parkes, D. (2017). Calibrated fairness in bandits. (Cit. on p. 13).

Lohia, P. (2021). *Priority-based post-processing bias mitigation for individual and group fairness*. (Cit. on p. 7).

Lohia, P., Ramamurthy, K., Bhide, M., Saha, D., Varshney, K., & Puri, R. (2018). Bias mitigation post-processing for individual and group fairness. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (cit. on p. 7).

Lumenova. (n.d.). *Group fairness vs. individual fairness in machine learning*. https://www.lumenova.ai/blog/group-fairness-vs-individual-fairness/ (accessed: 09.07.2023). (Cit. on p. 8)

Manrai, A., Funke, B., Rehm, H., Olesen, M., Maron, B., Szolovits, P., Margulies, D., Loscalzo, J., & Kohane, I. (2016). Genetic misdiagnoses and the potential for health disparities. *New England Journal of Medicine* (cit. on pp. 1, 5).

Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., & Galstyan, A. (2022). A survey on bias and fairness in machine learning. *ACM Comput. Surv.* (cit. on p. 4).

Murphy., K. (2012). *Machine learning: A probabilistic perspective*. MIT Press. (Cit. on pp. 28, 29).

Nemenyi, P. (1963). *Distribution-free multiple comparisons*. (Cit. on pp. 31, 32).

Pleiss, G., Raghavan, M., Wu, F., Kleinberg, J., & Weinberger, K. (2017). On fairness and calibration. *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 5684–5693 (cit. on pp. 7, 13).

ProPublica. (2016). https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing (accessed: 01.07.2023). (Cit. on p. 3)

Raschka, S., & Mirjalili, V. (2019). *Python machine learning: Machine learning and deep learning with python, scikit-learn, and tensorflow 2*. Packt Publishing. (Cit. on p. 27).

Saravanakumar, K. (2020). *The impossibility theorem of machine fairness–a causal perspective*. (Cit. on pp. 8, 12).

Sen, P., Hajra, M., & Ghosh, M. (2020). Supervised classification algorithms in machine learning: A survey and review. *Emerging Technology in Modelling and Graphics*, 99–111 (cit. on pp. 1, 4).

Shaw, R., & Corpas, M. (n.d.). *Further bias in personal genomics*. (Cit. on p. 5).

Suresh, H., & Guttag, J. (2019). A framework for understanding unintended consequences of machine learning. *ArXiv* (cit. on pp. 4, 5).

Sweeney, L. (2013). Discrimination in online ad delivery. (Cit. on p. 1).

Swinburne, R. (2004). Bayes' theorem. *Revue Philosophique de la France Et de l'Etranger*, *194*(2), 250–251 (cit. on p. 13).

T. Mahoney, M. H., K.R. Varshney. (2020). *Ai fairness*. O'Reilly Media, Inc. (Cit. on p. 6).

Tan, P., Steinbach, M., Karpatne, A., & Kumar, V. (2018). *Introduction to data mining*. Pearson Education. (Cit. on pp. 28, 29).

Wan, M., Zha, D., Liu, N., & Zou, N. (2023). In-processing modeling techniques for machine learning fairness: A survey. *ACM Trans. Knowl. Discov. Data* (cit. on p. 6).

Wei, D., Ramamurthy, K., & Calmon, F. (2020). Optimized score transformation for fair classification. *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics*, 1673–1683 (cit. on p. 14).

Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics Bulletin*, (6), 80–83 (cit. on pp. 31, 32).

Woodworth, B., Gunasekar, S., Ohannessian, M., & Srebro, N. (2017). Learning non-discriminatory predictors. *Proceedings of the 2017 Conference on Learning Theory* (cit. on p. 6).

Zafar, M., Valera, I., Rodriguez, M., & Gummadi, K. (2017). Fairness beyond disparate treatment & disparate impact. *Proceedings of the 26th International Conference on World Wide Web* (cit. on p. 13).

Zar, J. (2010). *Biostatistical analysis*. Pearson Prentice-Hall. (Cit. on p. 32).

Zehlike, M., Bonchi, F., Castillo, C., Hajian, S., Megahed, M., & Baeza-Yates, R. (2017). Fa¨ir. *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management* (cit. on p. 14).

A Wilcoxon signed-rank tests

	LogReg	Forest	Tree	Bagging	Boosting	NN
LogReg	1.0	0.2312	0.7533	0.2414	**0.0102**	**0.0004**
Forest	0.2312	1.0	0.7570	**0.0422**	**0.0063**	**0.0128**
Tree	0.7533	0.7570	1.0	**0.0012**	**0.0016**	**0.0347**
Bagging	0.2414	**0.0422**	**0.0012**	1.0	0.0544	**0.0023**
Boosting	**0.0102**	**0.0063**	**0.0016**	0.0544	1.0	**0.0002**
NN	**0.0004**	**0.0128**	**0.0347**	**0.0023**	**0.0002**	1.0

Tab. A.1.: P-values of the Wilcoxon signed-ranks tests comparing the 6 classifiers for the Optimal Swapping method.

	LogReg	Forest	Tree	Bagging	Boosting	NN
LogReg	1.0	0.2418	0.1500	0.2758	0.1658	**0.0001**
Forest	0.2418	1.0	0.0511	0.7982	0.5812	**0.0001**
Tree	0.1500	0.0511	1.0	**0.0025**	**0.0025**	**0.0027**
Bagging	0.2758	0.7982	**0.0025**	1.0	0.8401	**0.0001**
Boosting	0.1658	0.5812	**0.0025**	0.8401	1.0	**0.0001**
NN	**0.0001**	**0.0001**	**0.0027**	**0.0001**	**0.0001**	1.0

Tab. A.2.: P-values of the Wilcoxon signed-ranks tests comparing the 6 classifiers for the Optimal Swapping Greedy method.

	LogReg	Forest	Tree	Bagging	Boosting	NN
LogReg	1.0	0.5629	0.2418	**0.0264**	**0.0164**	**0.0003**
Forest	0.5629	1.0	0.1919	**0.0347**	**0.0247**	**0.0021**
Tree	0.2418	0.1919	1.0	**0.0173**	**0.0042**	**0.0214**
Bagging	**0.0264**	**0.0347**	**0.0173**	1.0	**0.0164**	**0.0002**
Boosting	**0.0164**	**0.0247**	**0.0042**	**0.0164**	1.0	**0.0002**
NN	**0.0003**	**0.0021**	**0.0215**	**0.0002**	**0.0002**	1.0

Tab. A.3.: P-values of the Wilcoxon signed-ranks tests comparing the 6 classifiers for the Massaging method.

	LogReg	Forest	Tree	Bagging	Boosting	NN
LogReg	1.0	**0.0002**	0.0736	0.0578	0.1829	**0.0018**
Forest	**0.0002**	1.0	**0.0007**	**0.0014**	**0.0004**	**0.0001**
Tree	0.0736	**0.0007**	1.0	0.6186	0.0450	**0.0010**
Bagging	0.0578	**0.0014**	0.6186	1.0	**0.0370**	**0.0006**
Boosting	0.1829	**0.0004**	**0.0450**	**0.0370**	1.0	**0.0032**
NN	**0.0011**	**0.0001**	**0.0010**	**0.0006**	**0.0032**	1.0

Tab. A.4.: P-values of the Wilcoxon signed-ranks tests comparing the 6 classifiers for the ROC method.

Abstract :

The popularity of machine learning derives from its application to decision-making in all fields. Classification is fundamental, as it allows data to be categorized on the basis of characteristics, with supervised classification learning from labeled data to predict outcomes. The question of fairness is essential to avoid biased results and promote equal treatment for all in many areas such as clinical testing or criminal justice. There are a number of different fairness measures, each more appropriate in certain situations. Promoting fairness in machine learning is crucial for responsible and fair decision-making.

This thesis develops a new fair post-processing method, Optimal Swapping, and its greedy version, for binary score predictions in supervised classification. It aims to mitigate unfairness while maintaining overall accuracy. Comparison analyses with other post-processing techniques, using statistical tests such as Friedman, Nemenyi, and Wilcoxon signed-rank, evaluate the method's efficiency in promoting fairness. By contributing to robust and effective fairness techniques, this work advances fairness machine learning applications.

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
Louvain School of Management
Place des Doyens, 1 bte L2.01.01, 1348 Louvain-la-Neuve
Boulevard Emile Devreux 6, 6000 Charleroi, Belgique
Chaussée de Binche 151, 7000 Mons, Belgique
www.uclouvain.be/lsm