TERRITORI APERTI

FONDO TERRITORI LAVORO E CONOSCENZA CGIL, CISL, UIL

Deliverable

Analysis of algorithmic debias and fairness

http://territoriaperti.univaq.it

UNIVERSITÀ DEGLI STUDI DELL'AQUILA

Project Title	:	Territori Aperti

Deliverable Number	:	
Title of Deliverable	:	Analysis of algorithmic debias and fairness
Nature of Deliverable	:	Report, Other
Dissemination level	:	Public
Licence	:	–
Version	:	1.0
Contractual Delivery Date	:	
Actual Delivery Date	:	
Contributing WP	:	WP 1.1 WP 3.2
Editor(s)	:	Giordano d'Aloisio (UNIVAQ), Antinisca Di Marco (UNIVAQ), Giovanni Stilo (UNIVAQ)
Author(s)	:	Giordano d'Aloisio (UNIVAQ), Antinisca Di Marco (UNIVAQ), Giovanni Stilo (UNIVAQ)
Reviewer(s)	:	Giordano d'Aloisio (UNIVAQ), Antinisca Di Marco (UNIVAQ), Giovanni Stilo (UNIVAQ)

Abstract

Data Science workflows are nowadays one of the most used tools by researchers and data scientists. Usually, their quality is compared to the quality of the classifier and measured with metrics like Precision, Recall, Accuracy and others. In this deliverable, we describe another requirement that could be used to measure the quality of a workflow, *Bias and Fairness*. We start by first making a survey of the many definitions of bias and fairness existing in litterature. Then, we present an analysis of some of the most popular Debias and Fairness methods and finally propose a modification of one of these algorithms.

Keyword List

Data Science, Bias, Fairness, Machine Learning, Artificial Intelligence

Glossary, acronyms & abbreviations

Item	Description
DS	Data Science
TPR	True Positive Rate
FPR	False Positive Rate

Table Of Contents

List Of Tables

List Of Figures

1 Introduction

Data is often heterogeneous, generated by subgroups with their own characteristics and behaviours. This heterogeneities can bias the data. A model learned on biased data may lead to unfair and inaccurate predictions [1]. This problem is reflected by many examples in history. In this deliverable, we want to make an in-depth analysis of *Bias* and *Fairness* in machine learning. We first make a survey of the many definitions of bias existing in literature and analyze the different metrics to measure fairness of a machine learning classifier. Then, we analyze some methods for mitigating bias and improve fairness. Our goal is to find the best metrics and the best methods to measure the fairness of an algorithm.

This deliverable will procede as follows:

- In chapter 2 we define bias and fairness describing several definitions and metrics existing in literature
- In chapter 3 we make an analysis of some existing methods for mitigating bias and introduce an extension of a generalization of one of them. Then, we test these methods a synthetic and some real datasets known in literature to be biased.
- Finally, chapter 4 describes some future works and concludes the deliverable

2 Definition of Bias and Fairness

In this chapter, we start by describing the concepts of bias and fairness. We first make a survey of the different definitions existing in literature. Then, we describe some existing metrics to measure dataset bias and model fairness and some algorithms used to mitigate them.

2.1. Bias definitions

A biased dataset is a dataset in which there is a statistical sample in which the probability of inclusion in the sample of individuals belonging to the population depends on the characteristics of the population under study. Bias in data can exist in many shapes and forms, leading to unfairness in different downstream learning tasks. For years, many definitions have been proposed, each trying to identify a different source of bias. By the time this thesis has been written, at least 23 different definitions of bias have been identified [1]. In the following, we will cite the most common and important of them:

1) **Historical Bias** *Historical bias is the already existing bias and socio-technical issues in the world and can seep into the data generation process even given a perfect sampling and feature selection* [2]
2) **Aggregation Bias** *Aggregation bias happens when false conclusions are drawn for a subgroup based on observing other different subgroups or generally when false assumptions about a population affect the model's outcome and definition.* [2]
3) **Temporal Bias** *Temporal bias arises from differences in populations and behaviours over time.* [3]
4) **Social Bias** *Social bias happens when other people's actions or content coming from them affect our judgment.* [4]
5) **Popularity Bias** *Items that are more popular tend to be exposed more. However, popularity metrics are subject to manipulation—for example, by fake reviews or social bots.* [5]
6) **Ranking Bias** *The idea that top-ranked results are the most relevant and important will result in attraction of more clicks than others.* [1]
7) **Evaluation Bias** *Evaluation bias happens during model evaluation. This includes the use of inappropriate and disproportionate benchmarks for evaluation of models.* [2]
8) **Emergent Bias** *Emergent bias happens as a result of use and interaction with real users. This bias arises due to change in population, cultural values, or societal knowledge, usually sometime after the completion of design.* [6]
9) **Behavioral Bias** *Behavioral bias arises from different user behaviour across platforms, contexts, or different datasets* [3]
10) **Presentation Bias** *Presentation bias is a result of how information is presented* [7]

11) **Linking Bias** *Linking bias arises when network attributes obtained from user connections, activities, or interactions differ and misrepresent the true behaviour of the users.* [3]

12) **Content Production Bias** *Content Production bias arises from structural, lexical, semantic, and syntactic differences in the contents generated by users.* [3]

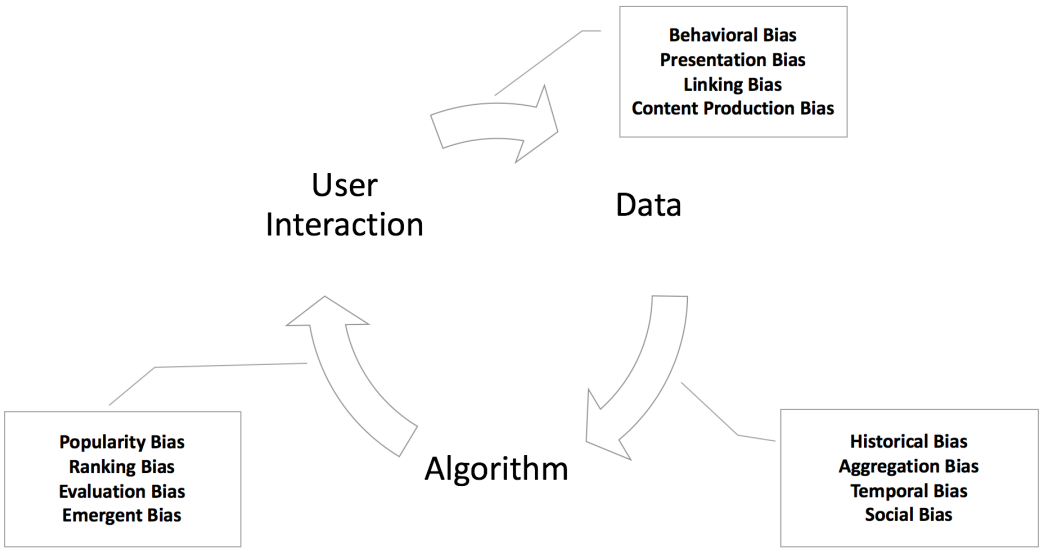


Figure 2.1: Bias Feedback Loop (from [1])

Figure 2.1 describes a possible categorization of the different definitions of bias based on the cause that generated it. Bias can be generated by the data, by the machine learning algorithm or by the user interaction with some system that again generates data. However, this categorization is not strict because bias can be propagated through the pipeline and generate another type of bias at another point of the workflow. In particular, bias in data can cause a bias in the algorithm that in turn can cause a bias in the user interaction favouring, for example, one thing instead of another. This process is called **Feedback Loop** [1] and is the reason why it is not possible to treat each definition of bias separately.

2.2. Definitions of Algorithmic Fairness and metrics

Algorithmic Fairness (that from now on we will call simply Fairness) can be defined as the absence of any prejudice or favouritism towards an individual or a group based on their intrinsic or acquired traits in the context of decision-making [8]. Usually, discrimination occurs with respect to some sensitive groups, identified by some *sensitive* (or *protected*) variables in the dataset . In particular, we define a privileged (unprivileged) group as a group (often defined by one or more sensitive variables) that are disproportionately (less) more likely to be positively classified [9]. Protected variables define the aspects of data that are socioculturally precarious for the application of ML. Common examples are gender, ethnicity, and age (as well as their synonyms). However, the notion of a protected variable can encompass any feature of the data that involves or concerns people [10].

The different definitions of fairness are strictly related to the metrics we use to measure the fairness of an algorithm. In the following, we list some fairness definitions and the related metric formulation:

1) **Statistical/Demographic Parity**

   One of the earliest definitions of fairness, this metric defines fairness as an equal probability of being classified with the positive label [11]:

$$Pr(\hat{y} = 1|A = 0) = Pr(\hat{y} = 1|A = 1) \tag{2.1}$$

2) **Disparate Impact**

Similar to statistical parity, this definition looks at the probability of being classified with the positive label. In contrast to parity, Disparate Impact considers the ratio between unprivileged and privileged groups. Its origins are in legal fairness considerations for selection procedures which sometimes use an 80% rule to define if a process has disparate impact (ratio smaller than 0.8) or not [12]:

$$\frac{Pr(\hat{y} = 1|A = 0)}{Pr(\hat{y} = 1|A = 1)} \tag{2.2}$$

3) **Equal Opportunity**

An algorithm is considered to be fair under equal opportunity if its TPR is the same across different groups. This means that the probability of a person in a positive class being assigned to a positive outcome should be equal for both protected and unprotected group members [13]:

$$Pr(\hat{y} = 1|A = 0, y = 1) = Pr(\hat{y} = 1|A = 1, y = 1) \tag{2.3}$$

4) **Equalized Odds (Average Opportunity)**

Similarly to equal opportunity, in addition to TPR equalized odds simultaneously considers FPR as well, i.e., the percentage of actual negatives that are predicted as positive [14]:

$$Pr(\hat{y} = 1|y = 1 \ \& \ A = 1) = Pr(\hat{y} = 1|y = 1 \ \& \ A = 0) \ \& \ Pr(\hat{y} = 1|y = 0 \ \& \ A = 1) = Pr(\hat{y} = 1|y = 0 \ \& \ A = 0) \tag{2.4}$$

5) **Generalized Entropy Index**

The Generalized Entropy Index (GEI) [15] considers differences in an individual's prediction (b_i) to the average prediction accuracy (μ). It can be adjusted based on the parameter α, where $b_i = \hat{y}_i - y_i + 1$ and $\mu = \frac{\sum_i b_i}{n}$:

$$GEI = \frac{1}{n\alpha(\alpha - 1)} \sum_{i}^{n} = 1[(\frac{b_i}{\mu})^{\alpha} - 1] \tag{2.5}$$

6) **Theil Index**

Theil Index is a special case of GEI for $\alpha = 1$. In this case, the formula simplifies to:

$$Theil = \frac{1}{n} \sum_{i=1}^{n} (\frac{b_i}{\mu}) \log(\frac{b_i}{\mu}) \tag{2.6}$$

Fairness definitions fall under two different groups [11, 16]:

- **Individual Fairness:** give similar predictions to similar individuals
- **Group Fairness:** treat different groups equally

Table 2.1 shows the categorization of the definitions described above.

Definition	Group	Individual
Statistical Parity	X	
Disparate Impact	X	
Equal Opportunity	X	
Equalized Odds	X	
GEI Index		X
Theil Index		X

Table 2.1: Categorization of fairness definitions

3 Analysis of Debias and Fairness Algorithms

Over the years, many approaches have been proposed to mitigate bias and improve the fairness of machine learning algorithms [1, 9]. These approaches can be categorized into three groups [16]:

- **Pre-processing**

  Pre-processing techniques try to transform the data so that the underlying discrimination is removed.

- **In-processing**

  In-processing techniques try to modify and change state-of-the-art learning algorithms in order to remove discrimination during the model training process.

- **Post-processing**

  Post-processing is performed after training by accessing a holdout set that was not involved during the training of the model.

The choice among algorithm categories can partially be made based on the system's ability to intervene at different parts of a machine learning pipeline. If the system is allowed to modify the training data, then pre-processing can be used. If the system is allowed to change the learning algorithm, then in-processing can be used. If the system can only treat the learned model as a black box without any ability to modify the training data or learning algorithm, then only post-processing can be used. However, it is recommended to apply the earliest category of methods possible in order to have the most flexibility and opportunity to correct bias [17].

In this chapter, we selected and analyzed the performances of three pre-processing algorithms. We have focused our analysis only on pre-processing methods because, as said above, they are the preferred choice if pre-processing is applicable. In section 3.1 we describe them briefly, while in section 3.2 we show the implemented analysis and describe the selected datasets.

3.1. Algorithms description

For our analysis, we have selected three of the most used pre-processing bias mitigation algorithms and compared their performances using the metrics described in chapter 2. For the sake of simplicity, we have limited our analysis to algorithms related to classification problems. In the following, we describe the selected methods.

3.1.1. Reweighing

Reweighing [18] is an algorithm that applies weights to the dataset's items according to the sensitive group they belong to and the label values. For example, items of the unprivileged group with a positive label will get higher weights than those with a negative label. Weights are calculated as the ratio be-

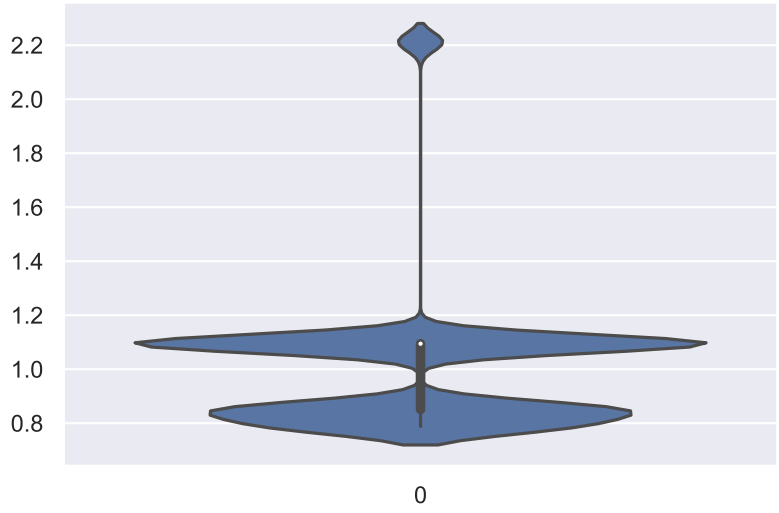


Figure 3.1: Distribution of weights

tween the expected probability of an item of a group to have a certain label and the observed probability of an item of a group to have a certain label:

$$W(X) = \frac{P_{exp}(S = X(S) \wedge Class = X(Class))}{P_{obs}(S = X(S) \wedge Class = X(Class))} \tag{3.1}$$

where:

$$P_{exp}(S = s \wedge Class = c) = \frac{|\{X \in D | X(S) = s\}|}{|D|} \cdot \frac{|\{X \in D | X(Class) = c\}|}{|D|} \tag{3.2}$$

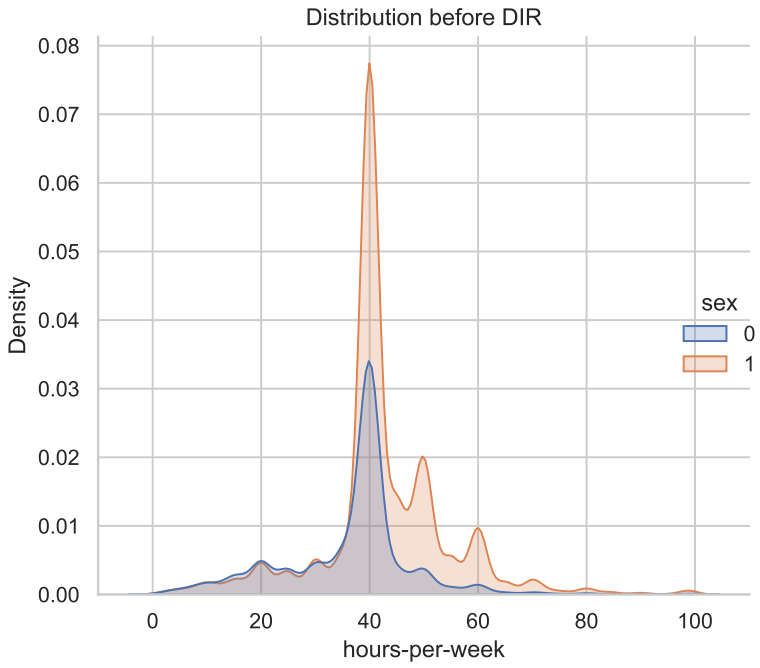
$$P_{obs}(S = s \wedge Class = c) = \frac{|\{X \in D | X(S) = s \wedge X(Class) = c\}|}{|D|} \tag{3.3}$$

Figure 3.1 shows an example distribution of weights for a biased dataset with a single sensitive variable. As we can see, there are two clusters with low weights and one small with a higher weight corresponding to items that differ from the expected observations.

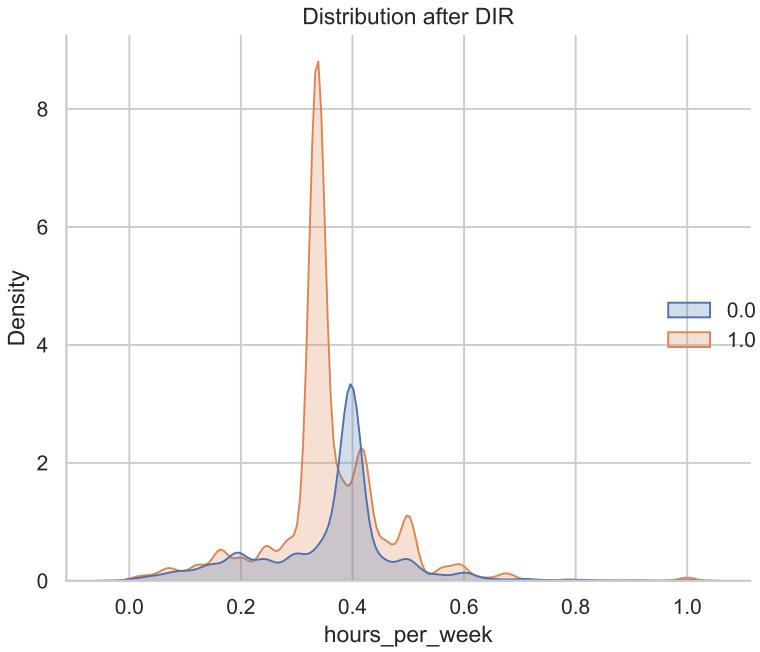
3.1.2. Disparate Impact Remover

Disparate Impact Remover (DIR) [12] is a pre-processing bias mitigation algorithm that changes the unprotected features of the dataset to remove the correlation between the protected attributes and them. The idea behind this algorithm is that even if we remove sensitive attributes from the dataset, a classifier can always trace this information from the value of other variables related to them. Given a protected attribute X and a single numerical unprotected attribute Y, let $Y_x = Pr(Y|X = x)$ be the marginal distribution of Y conditioned on $X = x$. The *rank* of $y \in Y_x$ is then defined as the cumulative distributions $F_x : Y_x \rightarrow [0, 1]$ for values $y \in Y$. In order to preserve the ability to predict the label correctly, this algorithm preserves the *rank* of the items inside each group. Formally, let $\bar{Y}$ be the repaired version of Y in the repaired dataset $\bar{D}$. We say that $\bar{D}$ *strongly preserves rank* if for any $y \in Y_x$ and $x \in X$, its repaired version $\bar{y} \in \bar{Y}_x$ has $F_x(y) = F_x(\bar{y})$.

Figure 3.2 shows the distribution of a nonsensitive variable (`hours-per-week`) before and after the application of the *DIR* algorithm. In this case, `sex` was the sensitive variable, and we can see in figure



(a) Distribution of unprotected variable before DIR



(b) Distribution of unprotected variable after DIR

Figure 3.2: Application of DIR to numerical variable

3.2(a) how the unprotected variable was related to it. In particular, a model could infer that if an item has a value of `hours-per-week` greater than 40, then it is very likely to have a value of `sex` equal to one, and so apply discrimination based on this information. Instead, after the application of *DIR*, we can see in figure 3.2(b) how the distribution of `hours-per-week` is more balanced. In this case, it is more difficult for a model to infer the membership group of an item looking only at the unprotected variable. It is also worth noting that the shape of the distributions is preserved, meaning that the rank of items inside the groups is preserved.

However, in order to work, this algorithm requires that most of the dataset variables are continuous. For example, figure 3.3 shows an application of the *DIR* algorithm to a dummy variable. In this case, we can see that the algorithm did not affect it since the variable has only two values, and so it is not possible to repair it, preserving the ranking. For this reason, if the dataset is mostly made of categorical, not orderable variables, this algorithm is not able to mitigate bias.

3.1.3. Sampling

Sampling [18] is a modified version of the *Reweighing* algorithm, in which weights are used to balance the dataset to remove discrimination. Sampling overcomes the limit of Reweighing ie that not all the classifiers consider weights during the learning process. This methods starts by partitioning the dataset in four groups: DP (*Deprived* group with *Positive* labels), DN (*Deprived* group with *Positive* labels), FP (*Favored* group with *Positive* labels) and FN (*Favored* group with *Negative* labels):

$$DP = \{X \in D | X(S) = b \wedge X(Class) = +\} \tag{3.4}$$
$$DN = \{X \in D | X(S) = b \wedge X(Class) = -\} \tag{3.5}$$
$$FP = \{X \in D | X(S) = w \wedge X(Class) = +\} \tag{3.6}$$
$$FN = \{X \in D | X(S) = w \wedge X(Class) = -\} \tag{3.7}$$

As for Reweighing, for each group, this algorithm computes the expected weight (W_{exp}) and the observed weight (W_{obs}). The ratio between these two values will be used to balance each group until the expected weight is reached. In particular, in the case of a biased dataset, *DN* and *FP* will have an observed weight higher than the expected weight, while *DP* and *FN* will be the vice-versa. In this case, the algorithm will randomly remove items from *DN* and *FP* and randomly duplicate items from *DP* and *FN* until the expected size is reached.

Figure 3.4 shows how the group disparity ($\frac{W_{exp}}{W_{obs}}$) of each group converges to one during the iterations of the algorithm. In particular, in about 1400 iterations the algorithm is able to balance the groups.

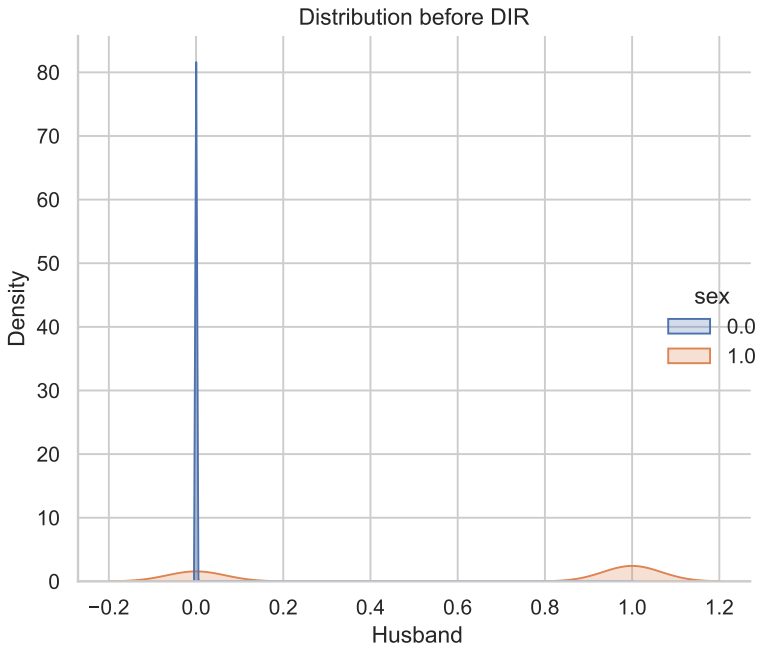
3.1.4. Sampling for multiple sensitive variables

Sampling was originally proposed in [18] only for datasets with a single protected variable, in which it was possible to identify the groups DP, DN, FP and FN. As a novel contribution, we have extended this algorithm to a multiple sensitive variable case. In particular, we consider each group formed by the combination of each value of sensitive variable (which, by the way, need always to be binary) and each value of the label. Formally, for each $s_1, \ldots, s_n \in S_1, \ldots, S_n$, each group will be defined as:

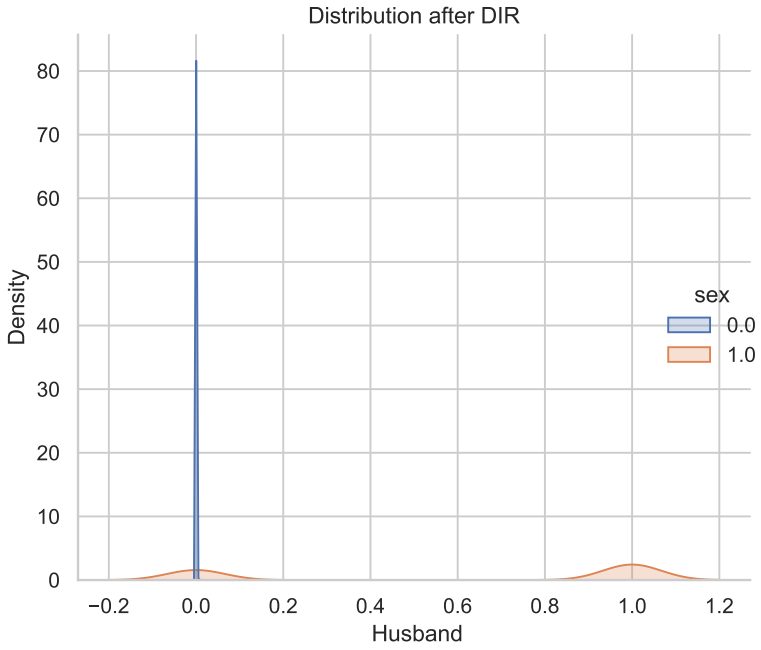
$$GP = \{X \in D | X(S_1) = s_1 \wedge X(S_2) = s_2 \wedge \cdots \wedge X(S_n) = s_n \wedge X(Class) = +\} \tag{3.8}$$
$$GN = \{X \in D | X(S_1) = s_1 \wedge X(S_2) = s_2 \wedge \cdots \wedge X(S_n) = s_n \wedge X(Class) = -\} \tag{3.9}$$

The algorithm, then iteratively adds or removes items from each group until the expected size is reached.

Figure 3.5 shows an example of sampling with multiple sensitive attributes. In particular, in this case there were two sensitive attributes and we can see that eight groups has been created. These groups



(a) Distribution of categorical variable before DIR



(b) Distribution of categorical variable after DIR

Figure 3.3: Application of DIR to categorical variable

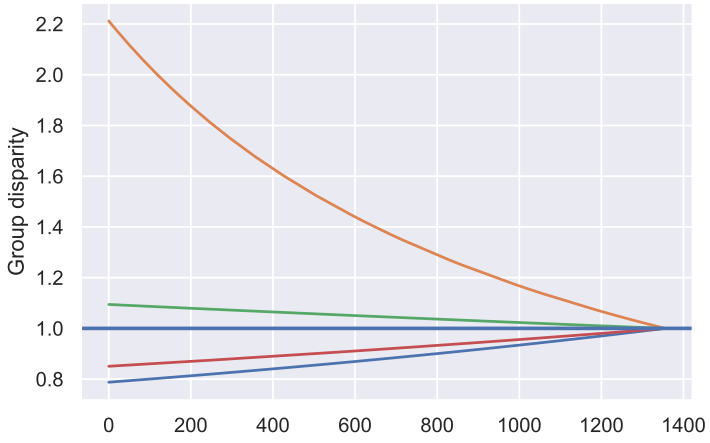


Figure 3.4: Sampling algorithm

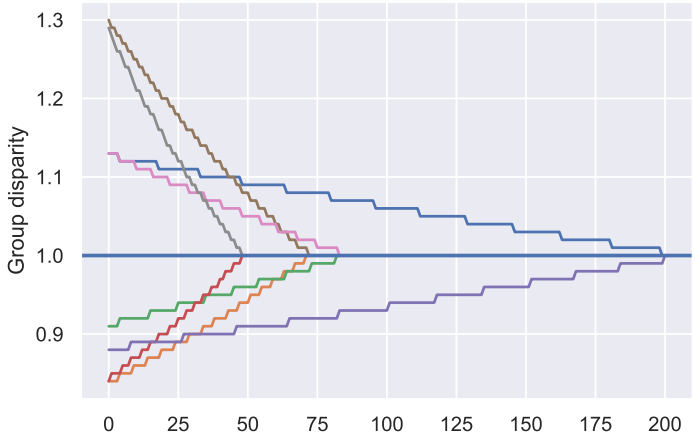


Figure 3.5: Sampling on multiple variable

are generated by the combination of the values of the two variables and the values of the label. The different shape of the curve compared to figure 3.4 is caused by the rounding applied to the weights in order to converge to one.

3.2. Analysis Description

This section describes the analysis we have done on the methods using the metrics described above. For this purpose, we have used the aif360 [1] implementation of the metrics, the Reweighing and DIR, while Sampling has been implemented from scratch. All the analysis has been done in Python.

3.2.1. Synthetic dataset

The first analysis has been done on a synthetic dataset created from scratch. This dataset is made of 10000 samples for 12 attributes, in which the attribute `10` is the label to predict and the attribute `s` is the sensitive variable.

[1] https://github.com/Trusted-AI/AIF360

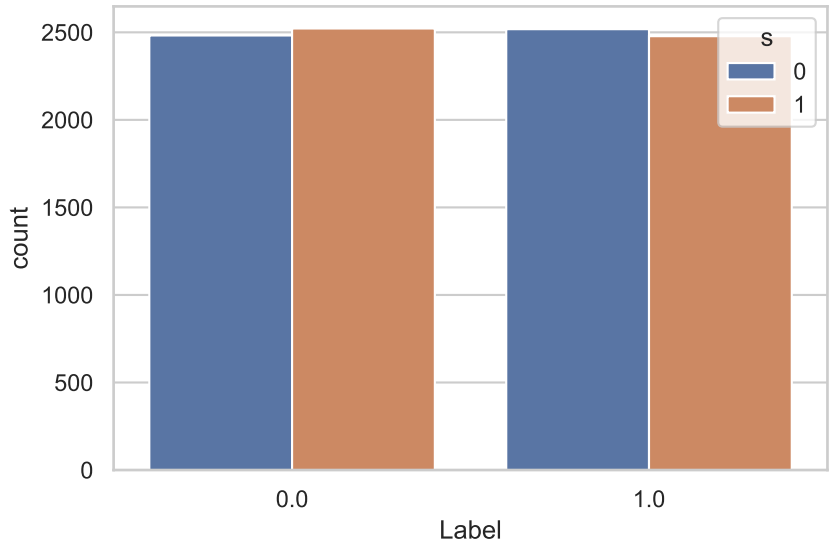


Figure 3.6: Distribution of labels for the unbias dataset

First of all, we applied our methods to an unbiased version of the dataset, in which the distribution of the label is homogeneous for both groups. Figure 3.6 shows the distribution of the labels for both groups. In this case, we can see from figure 3.7 that all methods preserve the fairness of the classifier and the accuracy of his predictions.

Then, we have introduced a bias in the dataset, unbalancing it. The distribution of the label for the entire dataset and the sensitive groups can be seen in the figure 3.8.

In this case, we can see that the DIR algorithm performs better in removing the bias. However, this improvement can be caused by removing the sensitive variable, which is not correlated to the others. Therefore, all the methods preserve the accuracy of the classifier.

Finally, we have combined the Reweighing and DIR algorithms with seeing if they can improve each other. As we can see from figure 3.10, combining these two algorithms improves the fairness of the classifier but tends to reverse the bias, favouring the previously unprivileged group and vice-versa.

From this analysis, we have seen that all these methods can improve the fairness of the classifier and the chosen metrics, except the Theil Index.

3.2.2. Datasets with single protected attribute

A second analysis has been done using real datasets known in the literature to be biased [1].

A first dataset analyzed is the **Adult Income** dataset [2]. This dataset is made of 30940 items for 15 features. The goal is to predict if a person has an income higher than 50k a year. This information is represented by the `income` variable. The sensitive attribute is `sex`.

Figure 3.11 shows the distribution of the `sex` and `income` variable and the distribution of income for the two sex groups. As we can see, the distribution of income for the two groups is unbalanced. This could be a symptom of bias.

Figure 3.12 shows the performances of the algorithms for this dataset. As we can see, Sampling and Reweighing are able to improve the fairness of the classifier, while DIR performs worse. The worst performances of DIR can be explained by the fact that Adult is mostly made of categorical variables. For this reason, a second test has been made with the Adult dataset considering only **numerical variables**. The distribution of the labels is the same as before. The only thing that has changed is that we removed

[2] https://archive.ics.uci.edu/ml/datasets/Adult

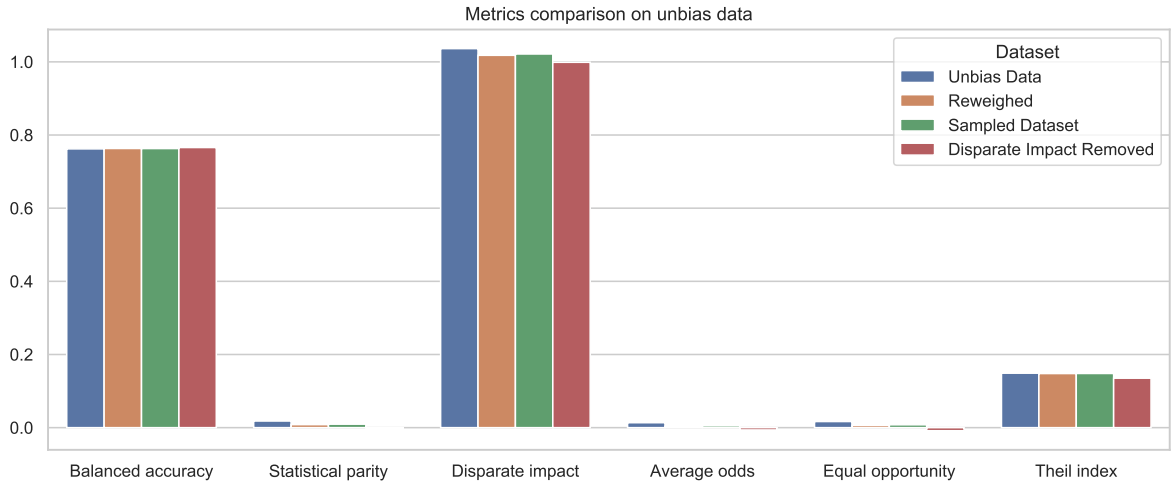


Figure 3.7: Metrics for synthetic unbias dataset

the categorical variables from the dataset. Figure 3.13 shows the performances of the algorithms in this case. As we can see, DIR performs better in this case, but, anyway, all the methods can improve the fairness of the classifier and perform better than the previous case. Finally, as for the synthetic dataset, we tried to combine Reweighing and DIR. In this analysis, we have used the numerical version of the dataset. Figure 3.14 shows the comparison of all these methods. Combining Reweighing and DIR improves a lot the fairness of the classifier, but it may cause a reverse bias.

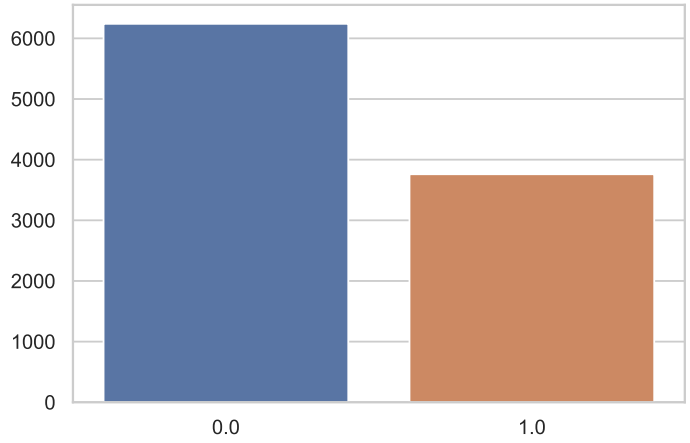
Another selected dataset with a single protected variable is the **Bank Marketing** [3] dataset. This dataset is related to direct marketing campaigns (phone calls) of a Portuguese banking institution. The classification goal is to predict if the client will subscribe to a term deposit (variable `y`). The sensitive variable is `age`, and the privileged group are people with less than 25 years. The dataset is made of 30488 samples for 21 columns. Figure 3.15 shows the sensitive variable's and the label's distributions and how the label is distributed between the two groups. From the figure, we can see that, although the privileged group is much smaller than the other, the favorable label is more present in the privileged group. This is a symptom of bias. Figure 3.16 shows the performances of the algorithms for this dataset. Since this dataset is made mostly by categorical variables, DIR is not able to improve the fairness of the classifier. Reweighing and Sampling instead can mitigate the classifier's bias, preserving the accuracy of the predictions. Combining Reweighing and DIR is of no advantage in this case since it is not able to mitigate the bias and, instead, worsen it.

Finally, the last selected dataset with a single protected variable is the **German Credit**[4] dataset. This dataset classifies people described by a set of attributes as good or bad credit risks. The dataset consists of 1000 instances and 20 features. Like the **Bank** dataset, here the sensitive variable is `age` and the privileged group are people with more than 25 years. This dataset, differently from the others, is not particularly bias, but we have selected it to see if the methods are able to detect also a small bias of the classifier. Figure 3.17 shows the metrics for this dataset. In this case, all the methods are able to mitigate the bias, since the dataset is not only made of categorical variables. As before, combining Reweighing and DIR does not give particular advantages.

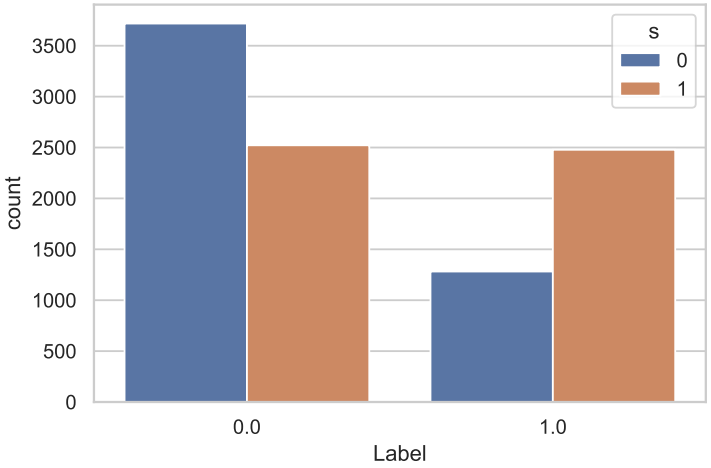
3.2.3. Datasets with multiple protected attributes

Now, we describe the analysis made on datasets with more protected variables.

[3] https://archive.ics.uci.edu/ml/datasets/Bank+Marketing
[4] https://archive.ics.uci.edu/ml/datasets/Statlog+%28German+Credit+Data%29



(a) Distribution of the label for the entire dataset



(b) Distribution of the label for the sensitive groups

Figure 3.8: Dataset bias

The first selected dataset is the **ProPublica Recidivism (COMPAS)**[5] dataset. COMPAS (Correctional Offender Management Profiling for Alternative Sanctions) is a popular commercial algorithm used by judges and parole officers for scoring criminal defendant's likelihood of reoffending (recidivism). It has been shown that the algorithm is biased in favor of white women defendants, and against black men inmates, based on a 2 year follow up study (i.e who actually committed crimes or violent crimes after 2 years) [19]. This dataset is made of 6167 samples for 398 attributes. The sensitive variables are `sex` and `race`. The goal is to predict if a person will be recidivist in the next two years. The favorable label in this case is 0 and the privileged group are caucasian women. Figure 3.18 shows the distribution of the label between the sensitive groups. As we can see, the probability of recidivism is higher for non-caucasian men. This could be a symptom of bias. Figure 3.19 shows the performances of the methods for this dataset. Before describing them, is worth noting that, in case of multiple sensitive variables, [12] suggests to apply the DIR transformation to the joint probability distribution of the sensitive variables. For this reason, only for DIR we have first computed the joint probability distribution, and then applied the fairness metrics using this variable as reference. From the picture we can see that

[5] https://github.com/propublica/compas-analysis

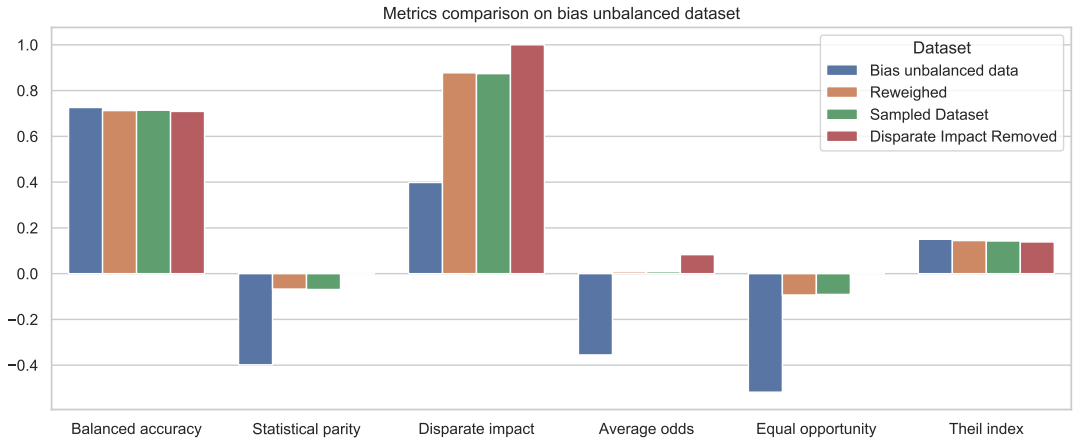


Figure 3.9: Metrics for unbalanced bias dataset

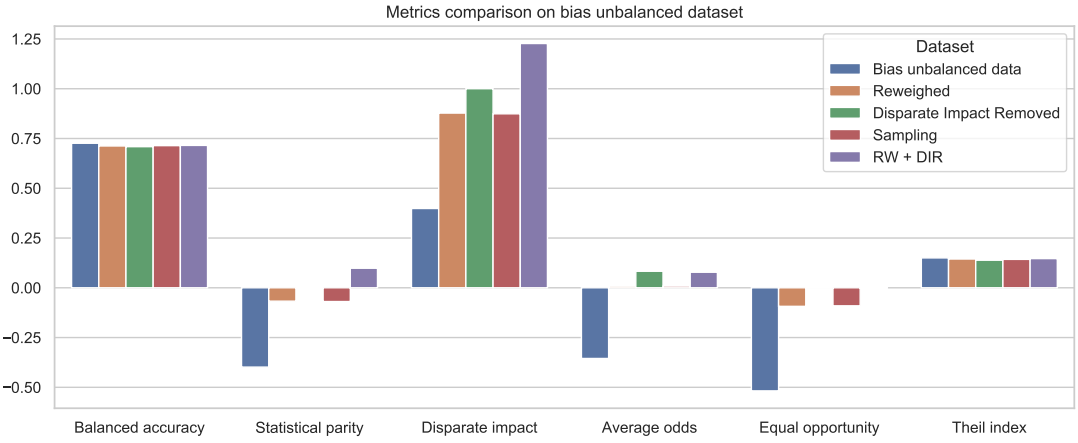


Figure 3.10: Reweighing + DIR comparison

Sampling performs very well in removing bias, while Reweighing behaves a little worse. Instead, DIR tends to reverse the bias of the classifier, and the same is for the combination of Reweighing and DIR.

A second analysis has been done using a version of the **German Credit** dataset with two sensitive variables. In this case, the privileged group are men with more than 25 years, while the unprivileged group are women with less then 25 years. Figure 3.20 shows the distribution of labels and sensitive variables. From the figure we can see that there is not an high bias in data. Figure 3.21 shows the performances of the methods. In this case, we can see that all the algorithms are able to detect and mitigate the bias. DIR is the one performing better.

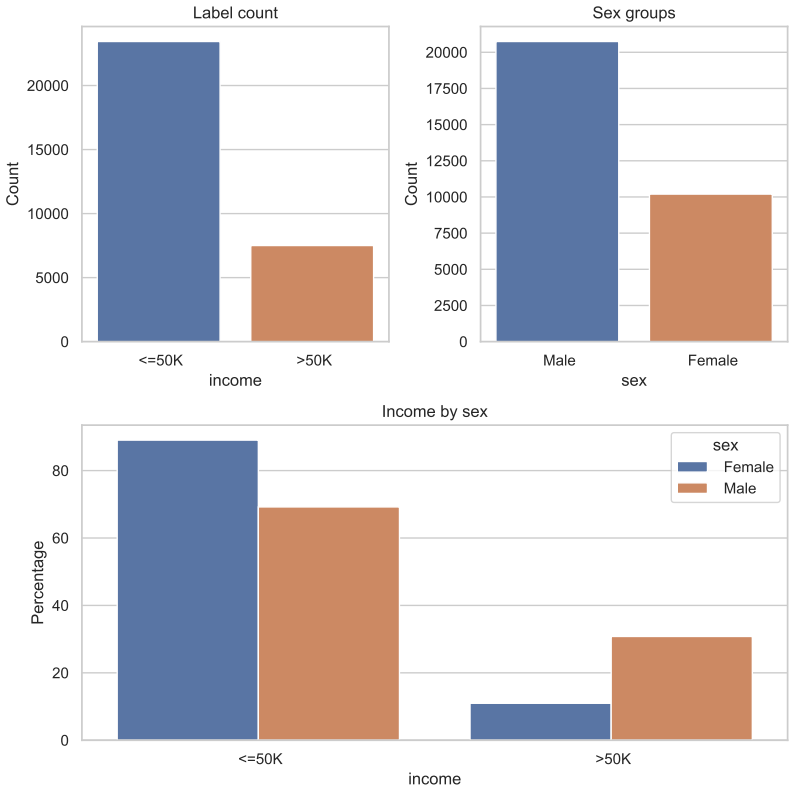


Figure 3.11: Distribution of features

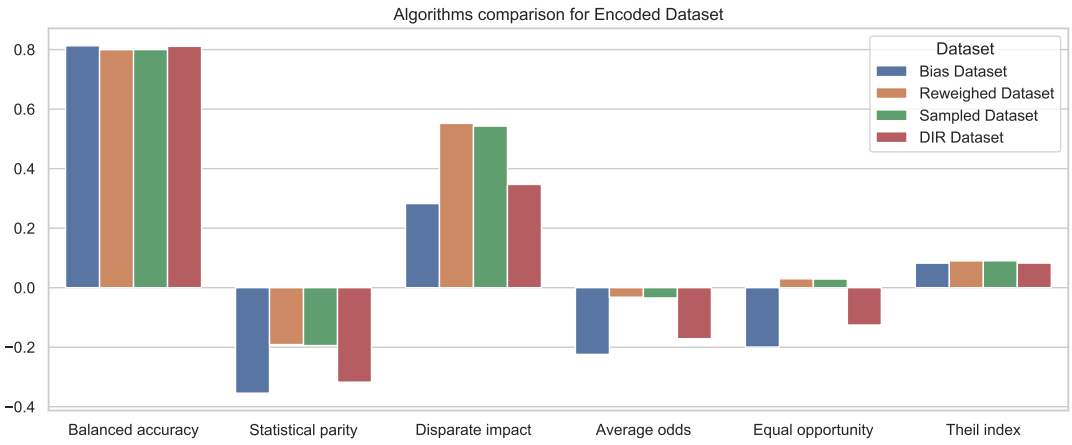


Figure 3.12: Metrics for Adult Dataset

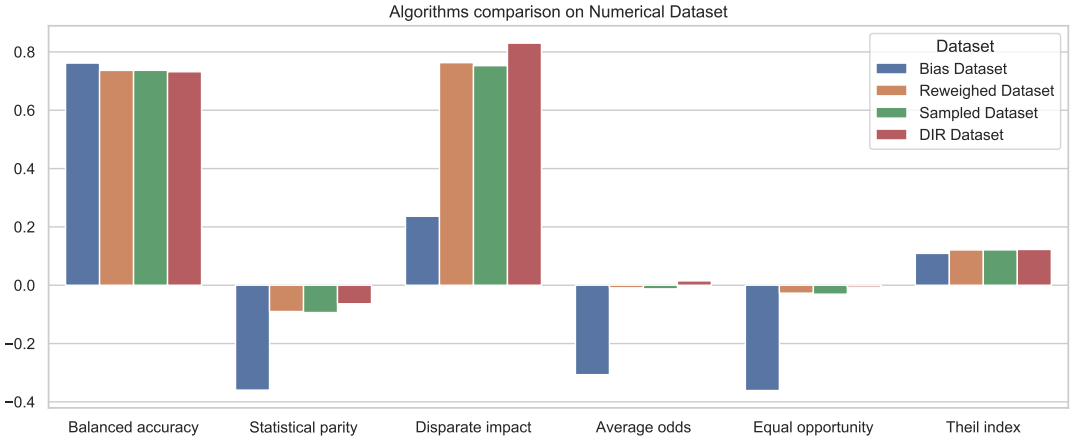


Figure 3.13: Metrics for Adult Dataset with numerical data

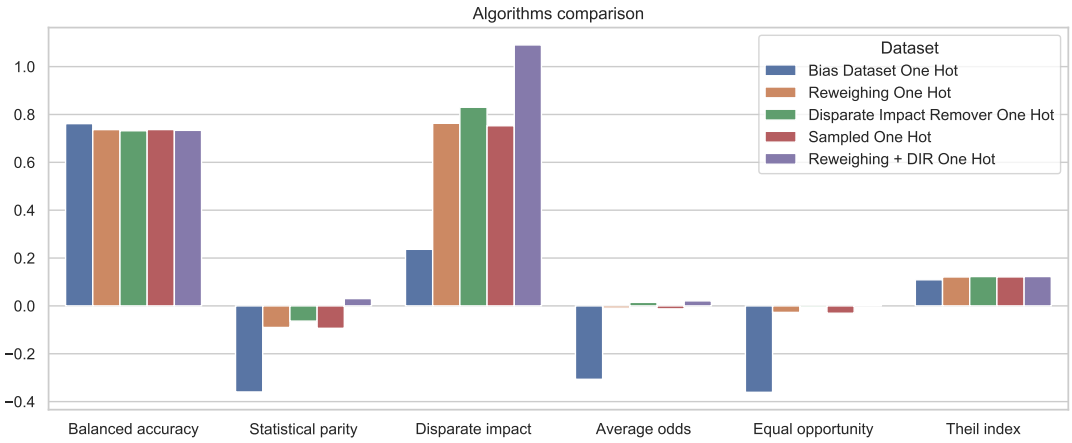


Figure 3.14: Reweighing + DIR comparison on the Adult Dataset

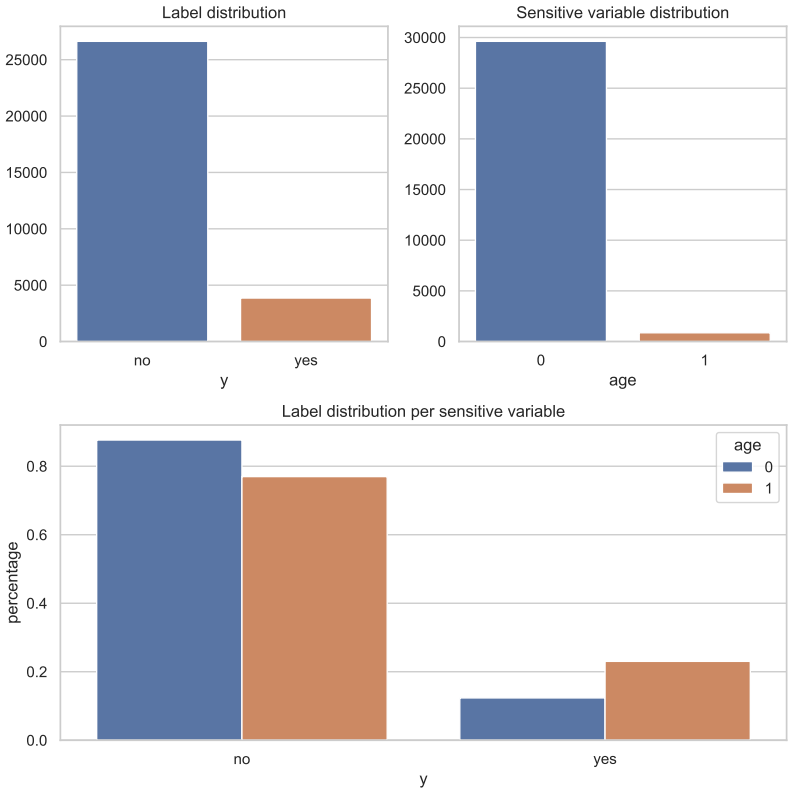


Figure 3.15: Distribution of sensitive variable and label of the Bank Dataset

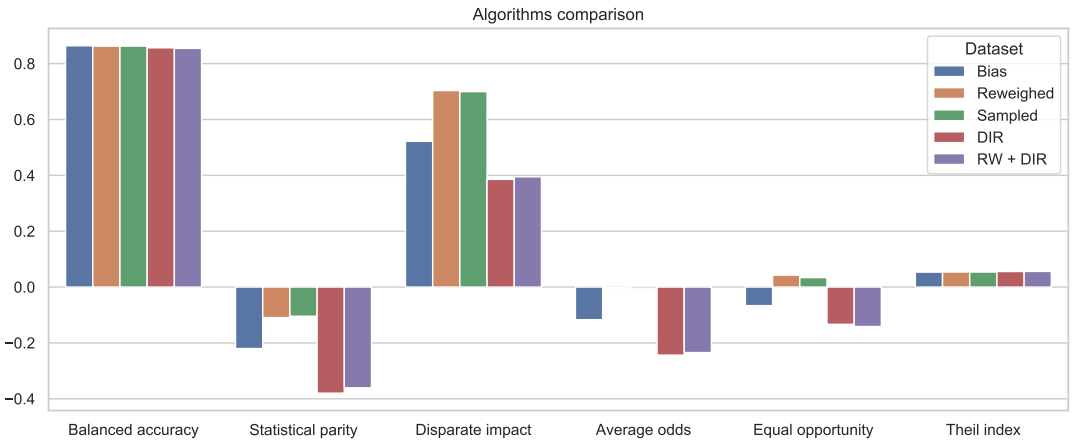


Figure 3.16: Metrics for Bank Dataset

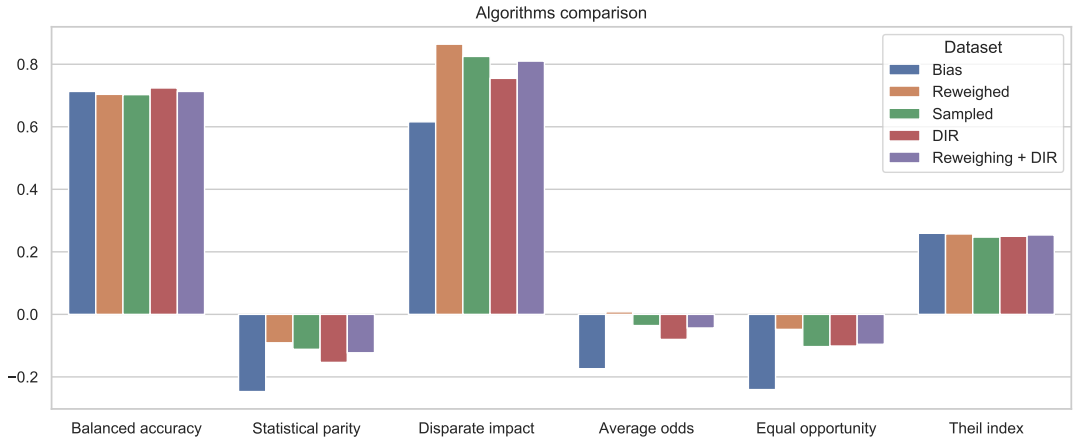


Figure 3.17: Metrics comparison for the German Dataset

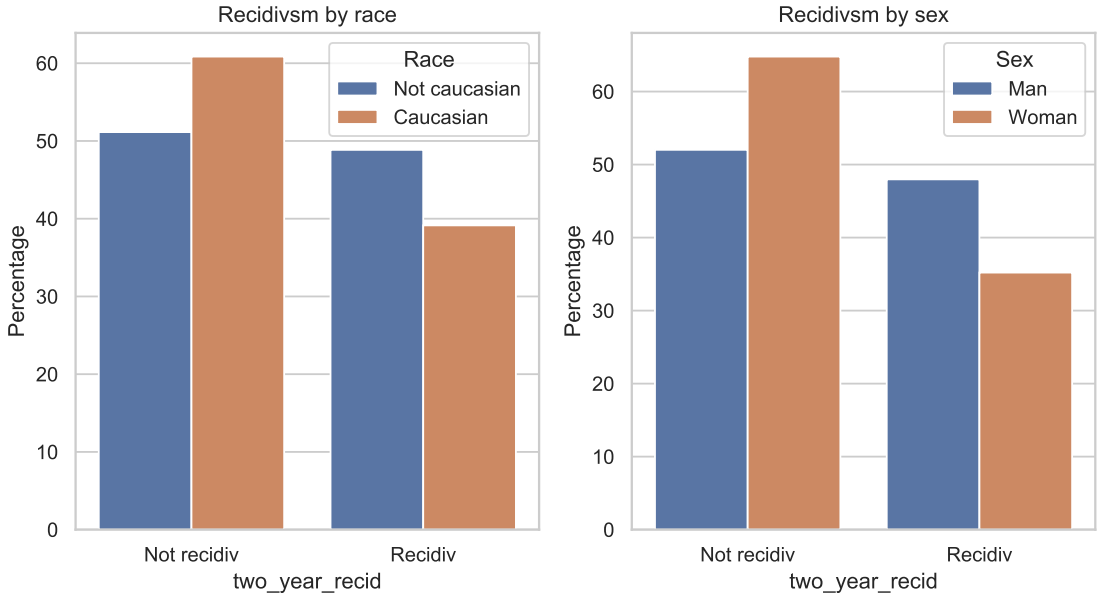


Figure 3.18: COMPAS label distributions

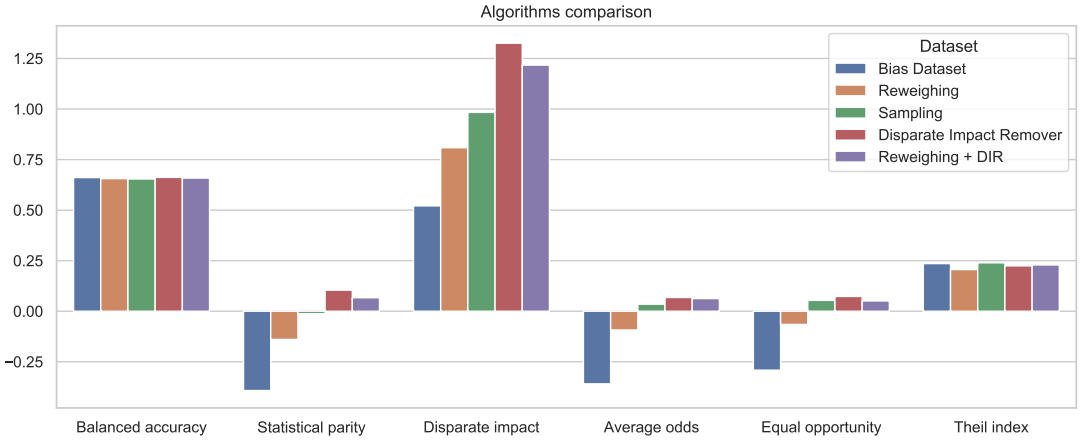
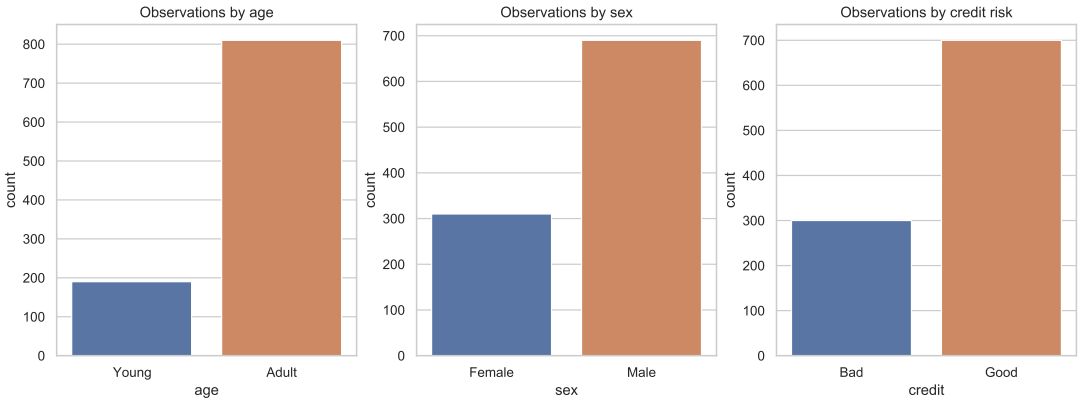
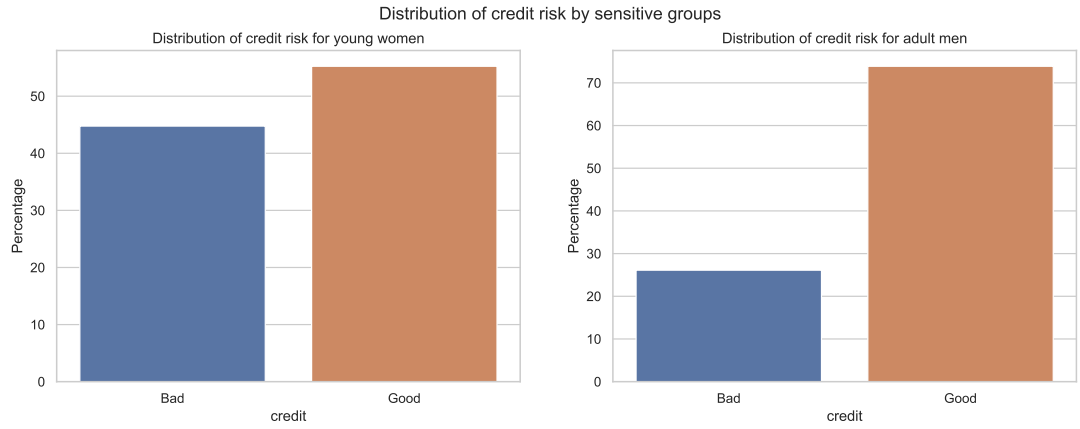


Figure 3.19: COMPAS metrics comparison



(a)



(b)

Figure 3.20: Distribution of sensitive variables and label for the German Credit

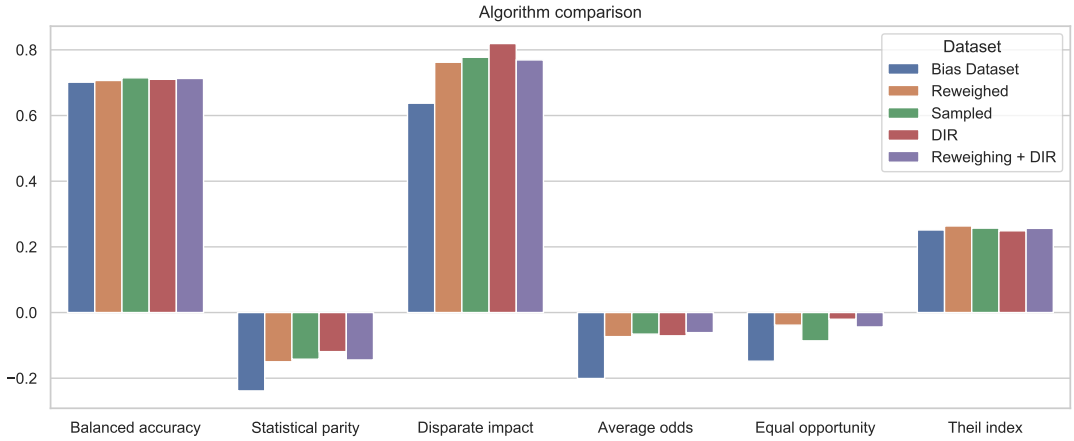


Figure 3.21: Methods performances for German Credit

4 Conclusions

This deliverable is an in-depth analysis of bias and fairness in machine learning. We have first of all made a survey of the many definitions and metrics for bias and fairness. Then, we have identified some pre-processing methods for mitigating bias and improving fairness, namely: *Reweighing*, *Disparate Impact Remover*, *Sampling* and a combination of *Reweighing* and *Sampling*, proposing also an extension of *Sampling* for multiple sensitive variables. We have tested them on several datasets. First of all a synthetic dataset, and them some real datasets with one or two sensitive variables. From our analysis we can deduce that: *Reweighing* and *Sampling* are able to manage every type of dataset, but in case of very high bias they are not able to remove it all. DIR instead is more robust, but require a dataset made by mostly numerical variables. Finally, combining *Reweighing* and *DIR* gives no evident utility.

As future works, we want to analyze how the size of the dataset and of the groups impacts on *Reweighing* and *Sampling*. We want also to analyze how dimensionality reduction techniques may impact on the performances of *DIR* for datasets with mostly categorical variables.

Bibliography

[1] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, “A survey on bias and fairness in machine learning,” *arXiv preprint arXiv:1908.09635*, 2019.

[2] H. Suresh and J. V. Guttag, “A framework for understanding unintended consequences of machine learning,” *CoRR*, vol. abs/1901.10002, 2019. [Online]. Available: http://arxiv.org/abs/1901.10002

[3] A. Olteanu, C. Castillo, F. Diaz, and E. Kıcıman, “Social data: Biases, methodological pitfalls, and ethical boundaries,” *Frontiers in Big Data*, vol. 2, p. 13, 2019.

[4] R. Baeza-Yates, “Bias on the web,” *Commun. ACM*, vol. 61, no. 6, p. 54–61, May 2018. [Online]. Available: https://doi.org/10.1145/3209581

[5] G. L. Ciampaglia, A. Nematzadeh, F. Menczer, and A. Flammini, “How algorithmic popularity bias hinders or promotes quality,” *Scientific reports*, vol. 8, no. 1, pp. 1–7, 2018.

[6] B. Friedman and H. Nissenbaum, “Bias in computer systems,” *ACM Trans. Inf. Syst.*, vol. 14, no. 3, p. 330–347, Jul. 1996. [Online]. Available: https://doi.org/10.1145/230538.230561

[7] R. Baeza-Yates, “Bias on the web,” *Communications of the ACM*, vol. 61, no. 6, pp. 54–61, 2018.

[8] N. A. Saxena, K. Huang, E. DeFilippis, G. Radanovic, D. C. Parkes, and Y. Liu, “How do fairness definitions fare? examining public attitudes towards algorithmic definitions of fairness,” in *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, 2019, pp. 99–106.

[9] S. Caton and C. Haas, “Fairness in machine learning: A survey,” 2020.

[10] S. Barocas, M. Hardt, and A. Narayanan, “Fairness in machine learning,” *Nips tutorial*, vol. 1, p. 2, 2017.

[11] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. Zemel, “Fairness through awareness,” in *Proceedings of the 3rd innovations in theoretical computer science conference*, 2012, pp. 214–226.

[12] M. Feldman, S. A. Friedler, J. Moeller, C. Scheidegger, and S. Venkatasubramanian, “Certifying and removing disparate impact,” in *proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, 2015, pp. 259–268.

[13] S. Verma and J. Rubin, “Fairness definitions explained,” in *2018 ieee/acm international workshop on software fairness (fairware)*. IEEE, 2018, pp. 1–7.

[14] R. Berk, H. Heidari, S. Jabbari, M. Kearns, and A. Roth, “Fairness in criminal justice risk assessments: The state of the art,” *Sociological Methods & Research*, p. 0049124118782533, 2018.

[15] T. Speicher, H. Heidari, N. Grgic-Hlaca, K. P. Gummadi, A. Singla, A. Weller, and M. B. Zafar, “A unified approach to quantifying algorithmic unfairness: Measuring individual &group unfairness via inequality indices,” in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 2239–2248.

[16] M. J. Kusner, J. R. Loftus, C. Russell, and R. Silva, “Counterfactual fairness,” *arXiv preprint arXiv:1703.06856*, 2017.

[17] R. K. E. Bellamy, K. Dey, M. Hind, S. C. Hoffman, S. Houde, K. Kannan, P. Lohia, J. Martino, S. Mehta, A. Mojsilovic, S. Nagar, K. N. Ramamurthy, J. T. Richards, D. Saha, P. Sattigeri, M. Singh, K. R. Varshney, and Y. Zhang, “AI fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias,” *CoRR*, vol. abs/1810.01943, 2018. [Online]. Available: http://arxiv.org/abs/1810.01943

[18] F. Kamiran and T. Calders, “Data preprocessing techniques for classification without discrimination,” *Knowledge and Information Systems*, vol. 33, no. 1, pp. 1–33, 2012.

[19] J. Angwin, J. Larson, S. Mattu, and L. Kirchner, “Machine bias,” *ProPublica, May*, vol. 23, no. 2016, pp. 139–159, 2016.