Performance analysis of machine learning algorithms trained on biased data

Renata Sendreti Broder[1] **, Lilian Berton**[1]

[1]Institute of Science and Technology – Federal University of Sao Paulo (UNIFESP)
São José dos Campos 12247-014 – SP – Brazil

{renata.broder,lberton}@unifesp.br

Abstract. *The use of Artificial Intelligence and Machine Learning algorithms in everyday life is common nowadays in several areas, bringing many possibilities and benefits to society. However, since there is room for learning algorithms to make decisions, the range of related ethical issues was also expanded. There are many complaints about Machine Learning applications that identify some kind of bias, disadvantaging or favoring some group, with the possibility of causing harm to a real person. The present work aims to shed light on the existence of biases, analyzing and comparing the behavior of different learning algorithms – namely Decision Tree, MLP, Naive Bayes, Random Forest, Logistic Regression and SVM – when being trained using biased data. We employed pre-processing algorithms for mitigating bias provided by IBM's framework AI Fairness 360.*

1. Introduction

Artificial Intelligence (AI) permeates various aspects of the daily life of most people living in the 21st century. It is pervasive because of the transparency to the user experience, that is, the user has only the result of processing a large mass of data in his hands, without understanding or even not being interested in how this processing is done, often having the feeling of AI impartiality through blind trust [Maybury 1990], because, if it is artificial intelligence, it does not have the biases that humans have. In this sense, so much trust is placed in the technology that search engines have become one of "one of our most trusted sources of information and, in many ways, have become arbitrators of truth" [Howard and Borenstein 2018]. Machine Learning (ML) is a subarea of AI, its algorithms are based on inductive learning, which consists of analyzing examples provided, so that pattern recognition and generalization can be carried out based on them [Bishop 2006]. Since the sample data that is used in the training of an ML model is produced through design and human action, they are not free from possible bias. Authors [Howard and Borenstein 2018] define "bias" as the act of "thinking about or treating another person differently based on perceived characteristics of the individual", and may also be unconscious, which is given the name of implicit bias.

The development of applications using AI has already generated many cases in which the bias was identified, disadvantaging or favoring some group. Cases range from problems in recognizing the faces of non-white people to better performance in recognizing the male voice at the expense of the female [Howard and Borenstein 2018]. There are also some known episodes, like the Google image search for "three white teenagers" that returned happy white teenagers while the search for "three black teenagers" in the same platform returned mugshots, or the case of COMPAS (*Correctional*

Offender Management Profiling for Alternative Sanctions), an auxiliary tool in the judicial system of some states in the United States (USA), such as Nova York and California, which aims to measure the likelihood of a defendant becoming a recidivist offender. A survey conducted by the agency *ProPublica* [Angwin et al. 2019] concluded that COMPAS tends to give a higher risk score for black people than for Caucasians with the same profile, when the reverse may be true.

In the face of so many cases of prejudice, discrimination and injustice in the form of bias in applications that use ML, it is necessary that the developers of such applications become aware of the existence of biases - both explicit and implicit - that can exist and be transferred to their creations, looking for ways to minimize them. This is because "given how much trust is placed in the technology, designers and coders carry with them significant ethical responsibilities for their creations" [Howard and Borenstein 2018]. In view of the ethical implications arising from the lack of representativeness in the data or even the introduction of prejudices in them, reflected in the ML models, the goal of this work is to analyze how various algorithms in this area behave when receiving a set of biased data as a training set. In this way, a comparison will be made between each of them through performance measures, trying to observe if there is any difference between them in relation to the bias - that is, if any of them would be more suitable than others to mitigate it. In addition, it is also intended to make an analysis of the algorithms after the data set has been altered by the pre-processing algorithms that have the objective of mitigating bias provided by IBM's framework *AI Fairness 360*.

The remaining of this paper is organized as follows: Section 2 presents the materials and methods employed in this work. Section 3 presents the results obtained in the classification. Section 4 presents the concluding remarks and future works.

2. Material and methods

2.1. Data sets

The data sets considered in this work contain bias and are generally used in research to address the issue of bias and justice in the ML [Mehrabi et al. 2019]. The following is a brief description of them.

2.1.1. *Adult Data Set*

It deals with the extraction from the 1994 census database. It contains attributes such as age, education, occupation, marital status, race, sex and the classification is whether a person has an annual income of up to $50,000 or if their annual income exceeds this amount. It has 48,842 instances and 14 attributes. With this data set, it is possible to observe the bias in relation to gender and race [Kohavi and Becker 1994]. This data set is also unbalanced, the information regarding the number of instances per class can be found in Table 1.

2.1.2. COMPAS

It contains two-year information on people who have committed crimes. Among the attributes, there is the name, age, gender, race, criminal record and the classification

Table 1. Quantity of instances by class for the *Adult Data set*

Group\Class	**>50K**	**<=50K**
Female, Non-Caucasian	227	2,938
Female, Caucasian	853	3,062
Male, Non-Caucasian	1,542	11,485
Male, Caucasian	9,065	19,670

is if a person was a recidivist offender in two years. This same data set was used for the development of a software called *Correctional Offender Management Profiling for Alternative Sanctions* (COMPAS) that is used by courts in some states in the USA to predict the risk of recidivism [Angwin et al. 2019]. As already mentioned, this data set is known to contain bias in relation to race and gender. This data set is also unbalanced, the information regarding the number of instances per class can be found in Table 2.

Table 2. Quantity of instances by class for the *COMPAS* Data set

Group\Class	**Non-Recurrent**	**Recurrent**
Female, Non-Caucasian	529	299
Female, Caucasian	368	199
Male, Non-Caucasian	1,946	1,986
Male, Caucasian	1,120	767

2.2. Metrics of *fairness*

The *fairness* metrics for ML are those that propose to evaluate the algorithms in relation to the bias. To present such measures, it is necessary to keep some concepts in mind: *protected attribute* is the attribute that separates pre-processing into different groups that are historically favored and disadvantaged - gender and race are examples of protected attributes that may exist in pre-processing. A *favorable label* is the label that indicates that the output of the algorithm for a given instance can mean an advantage - getting a loan and not being identified as a possible recidivist offender are examples of a favorable label [Bellamy et al. 2018, p. 2].

2.2.1. *Disparate Impact* (DI)

It is a rate calculated from dividing the probability of instances having an unfavorable value of a protected attribute X being classified with a favorable label $C = 1$ by the probability that instances with favorable value for the same protected attribute X receiving the same label $C = 1$, that is:

$$DI = \frac{Pr(C = 1|X = 0)}{Pr(C = 1|X = 1)} \tag{1}$$

It is said that there is *Disparate Impact* when the measure DI has a value ≤ 0.8, that is, if $DI > 0.8$ the measure points to a fair value [Feldman et al. 2015].

2.2.2. *Statistical Parity Difference* (SPD)

Measures the difference between the probability of instances with an unfavorable value of a protected attribute X being classified with a favorable label $C = 1$ and the probability of instances with favorable value for the same protected attribute X receiving the favorable label in $C = 1$:

$$SPD = Pr(C = 1|X = 0) - Pr(C = 1|X = 1) \tag{2}$$

The fairest value measured for SPD would be 0 and negative values would indicate a bias for the group $X = 1$.

2.3. *AI Fairness 360*

AI Fairness 360 (*aif360*) is a tool developed and maintained by IBM with the aim of generating confidence in AI, hoping to contribute to the mitigation of discriminatory bias. It consists of a library for *Python* - also available for *R* - which includes "a comprehensive set of metrics for data sets and models for testing biases", in addition to algorithms to mitigate biases [aif 2018].

The algorithms are divided into three categories, which are:

- Pre-processing: contains four bias mitigation algorithms that must be used before training the classifier;
- Processing: consists of six algorithms that contain mitigation measures during classifier training;
- Post-processing: there are three algorithms that can be used after training, working with the classifier outputs and changing them to mitigate bias.

As the aim of this work is to evaluate the performance of different algorithms in relation to the bias present in the data set, it was decided not to use the processing and post-processing techniques, so the focus remains in *aif360* pre-processing algorithms.

2.3.1. Pre-processing algorithms from *aif360*

The first one is *Reweighing* which is an approach that does not change the labels of the training data set, but instead assigns a weight to each instance in order to reduce the measure *Statistical Parity Difference* to 0 and maintain the probability of positive output. Instances that belong to the disadvantaged group of the protected attribute and have a positive label receive higher weights than those that have a negative label, and those that belong to the favored group and have a positive label receive lower weights than those with a negative label [Kamiran and Calders 2012].

This algorithm considers the expected probability (P_{exp}) - if the data set D was unbiased and the protected attribute and class were statistically independent - and the observed probability (P_{obs}) in the actual data set D, which are:

$$P_{exp}(C = 1|X = 0) = \frac{|\{d \in D|C(d) = 1\}|}{|D|} \cdot \frac{|\{d \in D|X(d) = 0\}|}{|D|} \tag{3}$$

$$P_{obs}(C = 1|X = 0) = \frac{|\{d \in D|C(d) = 1 \wedge X(d) = 0\}|}{|D|} \quad (4)$$

Keeping the probabilities in mind, the assignment of weights is done from the following equation:

$$W(X) = \frac{P_{exp}(C = 1|X = 0)}{P_{obs}(C = 1|X = 0)} \quad (5)$$

The next is the *Disparate Impact Remover* (DIR) algorithm, which works with the measure *Disparate Impact*, as the name suggests, aiming to mitigate it, trying to approximate it to 1. To achieve this goal, the training data set has the values of its unprotected attributes changed, keeping both the protected and the labels with their original values [Feldman et al. 2015]. The main idea of the algorithm is that the bias can be embedded in the attributes of the data set even if you remove the protected attribute, that is, you can infer the group of the protected attribute of an instance through its other attributes, so they need to be changed.

To make this change, it is necessary to look at the Y_x distribution of each Y unprotected attribute in relation to the X_i groups of the protected attribute and find a new A_y distribution such that $\sum x \in Xd(A_y, Y_x)$ is minimal, that is, the sum of the distances between the A_y distribution and each of the Y_x distributions is minimal.

2.4. Learning

The learning step consists of performing all the experiments using the pre-processed data sets. This work focused on the use of two of the pre-processing algorithms with the objective of mitigating bias offered by the *aif360* library, namely *Reweighing* and *Disparate Impact Remover* (DIR). For each data set used, several experiments were performed, as indicated in Table 3.

Table 3. Experiments performed by data set - disadvantaged groups

Dataset	Experiment1	Experiment2	Experiment3
Adult	Gender (female)	Race (non-Caucasian)	Gender (female) and Race (non-Caucasian)
COMPAS	Gender (male)	Race (non-Caucasian)	Gender (male) and Race (non-Caucasian)

Each of the experiments consists of three steps, the first of which is the training carried out without bias mitigation pre-processing algorithm, as a control experiment so that the results obtained in subsequent training could be observed; the second stage consists of training with the pre-processed data set using the *Reweighing* algorithm and the last stage, using the *Disparate Impact Remover* algorithm.

For the development of training, validation and test scripts, both the *aif360* library documentation and the notebooks made available by the team that developed it were used.

2.4.1. Data set pre-processing

Adult Data Set has three attributes - *workclass*, *occupation* and *native-country* - with many instances that have the value '?'. The value '?' was exchanged for NaN, and then all instances that had any NaN value were removed. The attribute *age*, which has several continuous values, was transformed and the ages were grouped into different age groups. This procedure was done for all age groups, from 10 to 69 years old, and all instances with 70 years old or more were grouped into an age group ≥ 70. Thereby, seven age groups were obtained. The attribute *education-num*, in which all possible values for education are ordered according to the level of education, has also been transformed by grouping, all those containing a value < 6 have been grouped, and all that had a value > 12 were also grouped, while the intermediate values remained the same. Thus, nine educational groups were created. The protected attribute *race* had the values $White$, $AsianPacIslander$, $AmerIndianEskimo$, $Black$ and $Other$, it was transformed into binary by assigning the value 1 for $White$, representing the favorable group, and the value 0 for all other, representing the unfavorable group. The attributes *capital-gain*, *capital-loss*, *hours-per-week* from the original data set have a lot of noise and *outliers*, so they have been removed. The attribute *fnlwgt*, which consists of a weight created by the Census Bureau, does not make sense for this analysis and, for this reason, has also been removed. At the end, the attributes used for analysis were those that contained the information of the original attributes *age*, *education*, *race*, *sex* and *income*.

For COMPAS data set, pre-processing starts by deleting attributes that are not relevant for the analysis, such as *name* and *surname*, *date of birth* and *case number*. The attributes kept were: *age*, *c_charge_degree*, *race*, *age_cat*, *score_text*, *sex*, *priors_count*, *days_b_screening_arrest*, *decile_score*, *is_recid*, *two_year_recid*, *c_jail_in*, *c_jail_out*. Then, only the instances that have the *days_b_screening_arrest* attribute value between -30 and 30 are selected - the rest of the instances are excluded - and a new attribute called *length of stay* which considers the difference between the date of release from prison (*c_jail_out*) and the entry date (*c_jail_in*) is created. The amount of data for *Asian* and *Native American* is very low in relation to the other races, while the quantities for *African American* and *Caucasian* are much larger. For this reason, only the last two are maintained. Finally, the attributes *days_b_screening_arrest*, *c_jail_out* and *c_jail_in* are removed and the attributes *priors_count*, *length_of_stay*, *score_text* and *age_cat* are categorized into different ranges. The protected attributes of this data set are race and gender, and the disadvantaged groups are the *African American* and the male gender, respectively.

2.4.2. Training without bias mitigation algorithm

This step starts with dividing the data set into a training set (70%), validation, and testing (both with 15%). For training, the separate data set for this purpose is scaled - using the StandardScaler method or MinMaxScaler from *sklearn.preprocessing*, depending on the type of classifier used - and then the *.fit()* method is trained using the set of training data, and a set of weights with all instances with a value equal to 1.

Then, the validation step begins by calling the *.predict_proba()* method, passing as

a parameter the separate data set for validation, returning to each instance the probability that it will be classified in each class. At this point, a vector is generated with 100 different class thresholds - from 0.01 to 0.99 - and for each of them, the balanced accuracy is calculated for this validation set. The threshold that obtains the highest balanced accuracy is chosen as the best class threshold, which will guide the test stage. This methodology alone is already a methodology that aims to reduce the discrimination that can come from an unbalanced data set, for when it is presented to train a classifier, it may tend to get it right more for the class containing the most examples and less for the others [Brodersen et al. 2010]. Balanced accuracy (BA) can be defined as follows:

$$BA = \frac{1}{2} \cdot \left(\frac{VP}{VP + FP} + \frac{VN}{VN + FN} \right) \tag{6}$$

Therefore, when BA is used in training, the aim is to find the best balance threshold for classifying an unbalanced set. Once trained and with the best threshold calculated, the next step is the test, which consists of calling the function *predict_proba()* passing the test set, obtaining the probabilities for each instance to be classified in each class. Using the best class threshold calculated in the previous step, the instances are classified and, comparing the original and predicted labels of the test set, the metrics that will be presented are calculated. In this step, there is also a direct prediction using the function *.predict()* which, instead of returning the probabilities, already returns the labels predicted by the classifier, without validation.

All this described processing of training, validation and testing, is performed thirty times with thirty different *seeds* - that is, random number generation "seeds" to ensure the reproducibility of the experiments - generated from an initial *seed*, of arbitrary value 13. The metrics presented in this work consist of the average of the metrics obtained in these experiments.

2.4.3. Training using the *Reweighing* algorithm

This second step is performed using the *Reweighing* method from the *aif360* library. It starts by dividing the data set into a training set (70%) and a test set (using the remaining 30%). The validation for the best class threshold is not done, since the best class thresholds are used for each *seed*, calculated in the previous step.

An instance of *Reweighing* is initialized, passing the parameters of privileged and non-privileged groups. The weights are then assigned to the instances, using the *Reweighing.fit()* method, and then the set is scaled, using MinMaxScaler or StandardScaler depending on the algorithm used.

The classifier is then trained, passing the set of scaled training data, the respective labels and the weights per instance calculated by *Reweighing*. In the same way as in the previous step, the best class threshold calculated is used to make the prediction of the labels of the test set and the metrics are calculated, using the original and predicted labels; the experiments are also carried out using the thirty different *seeds*.

2.4.4. Training using the *Disparate Impact Remover* algorithm

This last step is very similar to the previous one, with the difference that the tool *Disparate Impact Remover* (*DIR*) does not calculate weights, but makes changes to the attributes of the data set. Therefore, the set is divided into a training and test set in the same proportion; the training set is scaled and changed by the *DIR* tool and passed as a parameter to the algorithm that develops the classifier, along with the original class labels. Once the classifier is trained, the test is performed using the best class threshold calculated in the first stage. In the same way as in the two previous steps, the experiments metrics are calculated with thirty different configurations of the sets.

2.5. Algorithm settings

As the objective of this work is to focus mainly on the results obtained through the modifications made by the pre-processing algorithms provided by *aif360*, the predefined parameters of the learning algorithms for *sklearn*'s classifiers were used - *Logistic Regression*, *DecisionTreeClassifier*, *Bernoulli NB*, *Random Forest Classifier*. For SVM, the *Linear SVC* class was used instead of *SVC*, since it is similar to the first one when the kernel is set to 'linear' and is faster for large amounts of samples. The dual parameter was changed to 'False', as the documentation suggests that this is the setting that should be used when the number of samples is greater than the number of attributes, which is the case. A calibrator was also used for this classifier - *Calibrated Classifier CV* - as well as for *MLP Classifier*, in order to obtain the prediction probabilities for each class.

3. Results

The following algorithms are used in the experiments: DT (*Decision Tree*), MLP (*Multi Layer Perceptron*), NB (*Naive Bayes*), RF (*Random Forest*), LR (*Logistic Regression*), SVM (*Support Vector Machine*). The bias mitigation algorithms used are DIR (*Disparate Impact Remover*) and *Reweighing*. The evaluation metrics: Acc (*Accuracy*), SPD (*Statistical Parity Difference*) and DI (*Disparate Impact*).

3.1. Adult Data Set

The analysis of the adult data set corroborates a very important point, which concerns the *trade-off* Accuracy-Justice. Authors in [Kamiran and Calders 2012, p. 2] state that "on the one hand, the more discrimination we allow for, the higher accuracy we can obtain while on the other hand, in general, we can trade the accuracy in order to reduce the discrimination". For all experiments carried out with the original data set, without applying algorithmic measures to mitigate bias - also called *debiasing* - a relatively high accuracy was obtained, with the lowest being 73% and the highest 80.3%, as can be seen in Table 4.

In contrast, regarding the *fairness* DI metric, which corresponds to the relationship between the probability that instances belonging to disadvantaged groups (women and non-Caucasians) will be predicted as the favorable class - in this case, earning more than $50K$ annually - and the probability that the favored groups (men and Caucasians) will be predicted as the favorable class, there is a difference. As can be seen in Figure 1, in (a), which corresponds to the experiment that obtained the highest accuracy metrics, the

Table 4. *Adult Data Set* - Accuracy obtained in the experiments

Experiment	DT	MLP	NB	RF	LR	SVM
Without *debiasing*/validation	**0.803**	**0.803**	0.796	**0.803**	0.73	**0.803**
Without *debiasing*/with validation	0.734	0.734	0.738	0.734	0.737	0.728
Reweighing (Gender)	0.722	0.717	0.745	0.725	0.725	0.729
Reweighing (Race)	0.733	0.733	0.74	0.731	0.735	0.726
Reweighing (Gender and Race)	0.758	0.751	0.766	0.758	0.762	0.763
DIR Gender	0.718	0.712	0.72	0.717	0.72	0.729
DIR Race	0.732	0.731	0.74	0.731	0.734	0.726
DIR Gender and Race	0.718	0.711	0.735	0.718	0.723	0.734

values are lower than the results in (b), which brings the metrics to the experiment made with modification of the data set through the DIR algorithm.

Figure 2 brings the same SPD metric for the classifiers without bias mitigation algorithm than in 1 (a) and the classifiers produced after the *Reweighing* (b). It is possible to notice little variability between the metrics for all algorithms in each experiment, only RL had high improvement, which can also be observed for the DIR experiment.

Figure 1. *Adult Data Set - SPD*: Gender and race without debiasing (a) and with DIR (b)

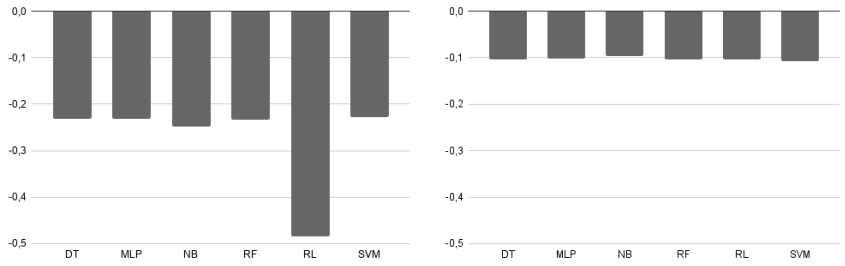
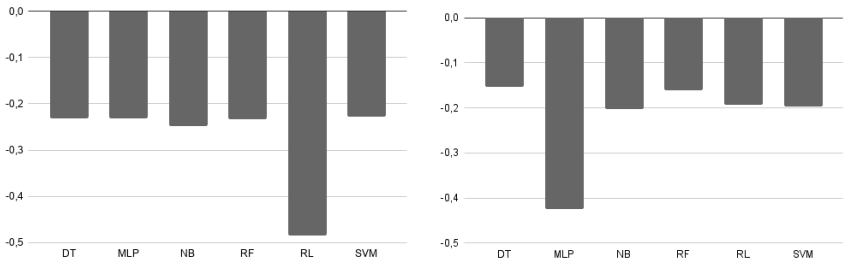


Figure 2. *Adult Data Set* - SPD: Gender and race without debiasing (a) and with Reweighing (b)



3.2. COMPAS

For the COMPAS data set, it can be observed in Table 5 that the accuracy obtained was very similar for all stages of all experiments performed, the lowest accuracy obtained was 64.7% and the highest was 66.3%. None of these results is high enough and for this reason, it is not possible to discuss the *trade off* Accuracy - Justice, as was possible with the previous data set, although it is possible to observe that there was some bias mitigation in some of the experiments.

Table 5. COMPAS - Accuracy obtained in experiments

Experiment	DT	MLP	NB	RF	LR	SVM
Without *debiasing*/validation	0.658	0.66	0.649	0.658	0.654	0.66
Without *debiasing*/with validation	0.655	0.656	0.648	0.653	0.658	0.658
Reweighing (Gender)	0.652	0.661	0.647	0.652	0.656	0.657
Reweighing (Race)	0.649	0.66	0.647	0.649	0.652	0.653
Reweighing (Gender and Race)	0.656	0.659	0.651	0.654	0.657	0.656
DIR Gender	0.658	0.659	0.648	0.656	0.661	0.662
DIR Race	0.66	0.66	0.647	0.659	**0.663**	**0.663**
DIR Gender and Race	0.661	0.66	0.647	0.659	0.661	0.661

Regarding the *fairness* DI metric, in this case, the difference between the algorithms without *debiasing* and with DIR is smaller, although DIR shows some improvement, as shown in Figure 3. It is possible to see this *debiasing* algorithm uniformized the DI among the classifiers.

Figure 4 shows the SPD without bias mitigation algorithm and the classifiers trained after *Reweighing*. The bias is quite visible through the metric, which indicates that the more negative, the greater the difference in the probability of favored groups being classified with favorable labels than disadvantaged groups. After *Reweighing*, all algorithms have some improvement in the bias, especially DT and RF.

Figure 3. COMPAS - DI: Gender and race without *debiasing* (a) and with *DIR* (b)

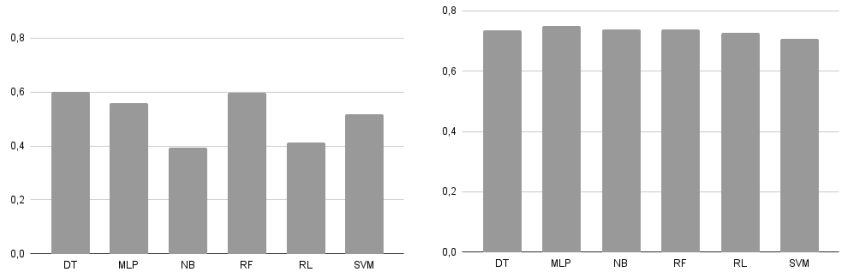
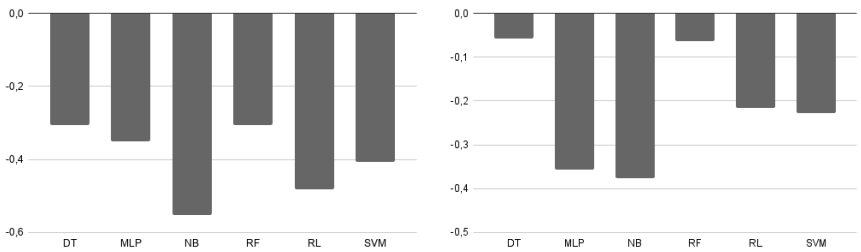


Figure 4. COMPAS - SPD: Gender and race without *debiasing* (a) and with *Reweighing* (b)



4. Conclusion

The analysis presented in this work reinforces ML does not offer universal rules that can be applied to all cases. Some applications that use ML have the potential to harm some minority groups, so it is necessary to be very careful in the development stage. It was

possible to observe no algorithm excels at reducing bias by itself. One of our experiment show an indication that the accuracy and justice (or absence of bias) may be inversely proportional and, therefore, to mitigate the bias of a classifier, it will be necessary to give up a higher accuracy. In this case, the situation of exchanging accuracy for justice would not represent a loss, since the existing data sets have a high probability of reflecting society's prejudices, not representing a fair world. Nevertheless, the authors understand that it is necessary to carry out such analysis on different and diverse other data sets to deepen the discussion on the so-called accuracy-justice trade-off.

As future work, more algorithms from *aif360* could be tested, encompassing pre-processing, processing and post-processing. But mostly, the goal is to expand the analysis of algorithm performance to other different data sets and, with that, to promote the debate about the bias issue. If the role of ML and AI professionals is to carry out their work ethically, one should not ignore the existence of data bias and conveniently contribute to the perpetuation of inequalities, but, fight it.

References

(2018). Ai fairness 360 - resources. Acessed in Out. 3rd 2020.

Angwin, J., Larson, J., Mattu, S., and Kirchner, L. (2019). Machine bias: There's software used across the country to predict future criminals and it's biased against blacks. 2016. *URL https://www. propublica. org/article/machine-bias-risk-assessments-in-criminal-sentencing*.

Bellamy, R. K., Dey, K., Hind, M., Hoffman, S. C., Houde, S., Kannan, K., Lohia, P., Martino, J., Mehta, S., Mojsilovic, A., et al. (2018). Ai fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias. *arXiv preprint arXiv:1810.01943*.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. springer.

Brodersen, K. H., Ong, C. S., Stephan, K. E., and Buhmann, J. M. (2010). The balanced accuracy and its posterior distribution. In *2010 20th international conference on pattern recognition*, pages 3121–3124. IEEE.

Feldman, M., Friedler, S. A., Moeller, J., Scheidegger, C., and Venkatasubramanian, S. (2015). Certifying and removing disparate impact. In *proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 259–268.

Howard, A. and Borenstein, J. (2018). The ugly truth about ourselves and our robot creations: the problem of bias and social inequity. *Science and engineering ethics*, 24(5):1521–1536.

Kamiran, F. and Calders, T. (2012). Data preprocessing techniques for classification without discrimination. *Knowledge and Information Systems*, 33(1):1–33.

Kohavi, R. and Becker, B. (1994). UCI machine learning repository.

Maybury, M. T. (1990). The mind matters: artificial intelligence and its societal implications. *IEEE Technology and Society Magazine*, 9(2):7–15.

Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., and Galstyan, A. (2019). A survey on bias and fairness in machine learning. *arXiv preprint arXiv:1908.09635*.