Improving Gender Fairness for Random Forest Model through a Post-processing Method

Yuhao Ding
New York Univeristy Shanghai
yd1158@nyu.edu

Abstract

Prediction related to classification problems, such as predicting incomes, may lead to bias against certain groups of people. Metrics on fairness is thus invented to prevent these kinds of bias. Previous work on this subject used a various of methods, such as pre-processing the data, changing the loss function, or altering the model structure. In this paper, we focus on Adult Income Dataset by UCL, and use Random Forest Model to classify income levels ($\geq$50K per year). To neutralize gender bias, we invented a post-processing method which we called corrector network. We will then shown the results compared with several baseline models to show its efficiency. In the rest of paper we will share our understandings on why and how the model works.

1 Introduction

Nowadays, an increasing number of decisions have been made by automated artificial intelligence models(Pessach and Shmueli, 2020). Though these algorithms have increased accuracy in many predictions, it turns out that AI algorithms are not always as objective. They may cause a bias against gender, race or age (Pessach and Shmueli, 2020). Therefore, in many decision making models, fairness has gained an increasing attention. In prediction such as household income, credit, or criminal justice, machine learning models may yield bias because historical human decisions are used as training data (Williams et al., 2018).

To improve fairness of artificial intelligence models, some general legal definitions are introduced, such as disparate impact, demographic parity, equalized odds, etc., different measures will have different advantages and disadvantages (Pessach and Shmueli, 2020). Many researches have shown that different fairness measures cannot be satisfied simultaneously. Thus, our research only focus on improving one metric of evaluating fairness. Specifically, we chose to satisfy Equalized Odds metric.

In our experiments, we choose to use Adult Income Dataset, and choose Gender to be the sensitive attribute. Our goal is to ensure that the model will not only be accurate, but also be fair on different genders. In previous studies, many fairness-enhancing mechanisms are introduced and they are classified as pre-process, in-process and post-process mechanisms. Pre-process mechanisms change the training data before feeding it into a machine learning algorithm; In-process mechanisms modifies the training algorithms; and post-processing mechanisms directly change the output score to improve fairness.

We devised a new post-processing model can could both improve accuracy and fairness. Our method is novel in many aspects. First, previous papers does not focus on post-processing random forest model, however, this research is meaningful as random forest model is very effective in predicting results for our dataset. Second, the way to construct our post-processing model is new. Third, the data we feed into the model is selected based on a new approach. Specifically, we first trained a random forest model to predict outcomes for our dataset, we fine-tuned the hyper-parameters to achieve good fairness and accuracy. Then, we trained another random forest model, which we called corrector RF, to change the output scores of the previous random forest model. Finally, we demonstrate that after post-processing, both accuracy and fairness can be improved compared to several baseline methods.

The remainder of the paper is organized as follows. First, we introduce the results of fine-tuning the first Random Forest model. Then, we explain our post-processing model in detail, and show its effectiveness empirical using several baseline models. Lastly, we tried to explain why and how this

model works in a theoretical perspective.

2 Definitions and Dataset

2.1 Fairness Definition

There are different definitions of fairness when coming to fairness studies. Generally, a fair model involve avoiding discrimination on a particular kind of group membership, such as gender or race (McNamara, 2019). Different metrics include Equalized Odds, Demographic Parity, Disparate Impact, etc. (Pessach and Shmueli, 2020). In this work we will use *Equalized Odds* as our metric.

Equalized Odds is designed by Hardt et. al, which computes the difference between the false positive rates (FPR), and the difference between the true positive rates (TPR) of the two groups (Hardt et al., 2016). In our work we calculate the sum of the two differences to be the final fairness indicator.

Suppose the protected attribute is S. The prediction is $\hat{Y}$, and the true label is Y. Our metric can be denoted as:

$$\begin{aligned}\Delta EO = &\sum_{y=0}^{1}[|P(\hat{y}=1|S=1,Y=y)\\ &-P(\hat{y}=1|S=0,Y=y)|]\\ =&P(\hat{y}=1|S=1,Y=1)\\ &-P(\hat{y}=1|S=0,Y=1)\\ &+P(\hat{y}=1|S=1,Y=0)\\ &-P(\hat{y}=1|S=0,Y=0)\end{aligned}$$

To facilitate our calculation, we use the open source tool kit Aequitas to calculate fairness (Saleiro et al., 2018).

2.2 Dataset

The dataset we used is Adult Income Dataset [1] from the UCI repository. It has a total of 45, 222 data instances, each with 14 features such as gender, marital status, educational level, number of work hours per week. Similar to some previous studies such as (Quadrianto et al., 2019), we consider gender as a binary protected attribute. We use 80% of the instances to be the training data, and the remaining 20% as test data.

Our pre-processing methods including removing or inputting missing data, using one-hot encoding for each categorical variable. For each random testing, we re-select training and testing data to avoid results due to randomness.

[1] https://archive.ics.uci.edu/ml/datasets/adult

3 Our Model

3.1 Model Overview

Our model involves first training a Random Forest model (Main RF) for perdiction, and then train another Random Forest model (Corrector RF) to change the output scores of the previous Random Forest Model (Main RF). On the training side, we will first train the Main RF, then train the Corrector RF. During prediction, we will first obtain the output scores of the Main RF, then we will use an algorithm to correct the output scores using results from the Corrector RF.

Before introducing the details of training the model, we include a graph that generally shows how the model works in *Figure 1*.

3.2 Training the model

As mentioned above, there are two models to be trained, *Main Random Forest* and *Corrector Random Forest*.

Figure 2 shows a detailed sequence of how we trained the two models.

3.2.1 Main Random Forest

As our methods are a post-processing method, training the *Main Random Forest* does not have difference compared to a normal training process. After the 80%-20% train-test-split, the training data is put into the Main Random Forest to be trained, the output result will be a binary classification of a household's income.

3.2.2 Corrector Random Forest

The intuition of the *Corrector Random Forest* is that it can tell whether the predictions in the *Main Random Forest* is wrong. Therefore, the output of the *Corrector Random Forest* will be a binary score indicating whether the output score in the *Main Random Forest* needs to be changed. Specifically, Output 0 indicates that the output score in the *Main Random Forest* does not need to be changed, and 1 indicates that the score need to be changed. *Table 1* shows all the possible output scores of the two Random Forests and the final results of each case.

With the output defined above, training the Corrector Random Forest requires a set of newly built data. Specifically, after the Main Random Forest has been trained, we can extract all data in the training set that are predicted to be either False Negative or False Positive (i.e wrongly predicted). We will

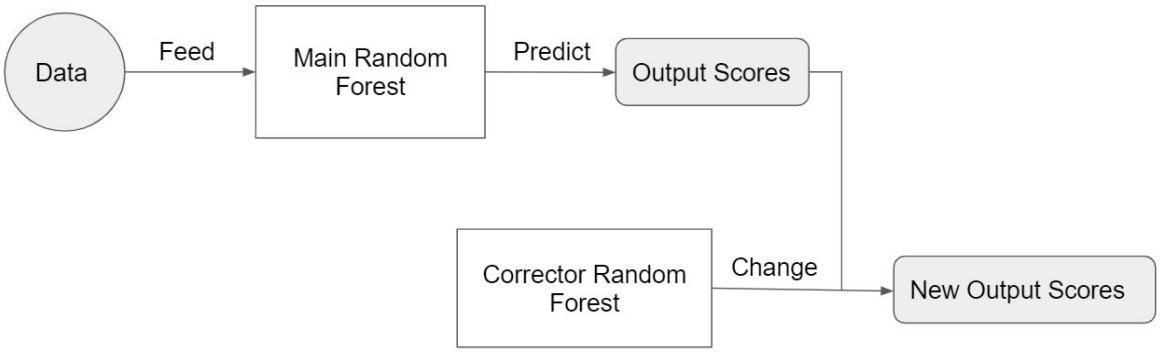

Figure 1: After the main Random Forest makes the prediction, the corrector Random Forest directly changes the output scores to generate new output scores that satisfies the fairness constraint.

Main RF Output	Corrector RF Output	Final Output
0	0	0
0	1	1
1	0	1
1	1	0

Table 1: If *Corrector Random Forest* has a output score of *1*, the output score of the *Main Random Forest* will be reversed.

give these data a new predictive label *1*, indicating that they need to be corrected. We will then combine them with a set of data containing True Negative and True Positive training samples, and give them label *0*, indicating that they didn't need to be changed.

With this new training set, the *Corrector Random Forest* can be trained to meet out intuition.

3.3 Testing the model

After the two random forests has been trained, the final results will be generated as follows.

The data will first be fed into the *Main Random Forest* and generate a preliminary output scores. Then, the data combined with this preliminary output scores will be fed into the *Corrector Random Forest*. The *Corrector Random Forest* will identify a list of data which their output scores need to be reversed. Suppose inside there are x male samples and y female samples.

After we have this list of data, we then use a simple algorithm to reverse the output scores. We will first obtain the fairness results in the training samples. As the false positive rate of male samples are always bigger than female samples, we first calculate their difference.

$$\text{diff} = \text{false positive rate}_{\text{female}} - \text{false positive rate}_{\text{male}}$$

Using the result from equation (1), we will random sample $(100\% - \text{diff})$ of male data samples and reverse their outputs. For female samples, we simply reverse all outputs identified by the *Corrector Random Forest*. This is based on the in tuition that the fairness pattern of the training set is similar to the fairness pattern in the testing set.

A graphic explanation is given in *Figure 3*.

4 Experimental Evaluation

4.1 Fine-tuning the Main Random Forest

Our first experiment involves finding the best setting for the *Main Random Forest* model. Our results show that tuning the hyper-parameters of the model could affect the fairness and accuracy result. *Figure 4* shows a detailed variation in fairness with respect to race and gender, and accuracy when using different number of samples. In this graph, the higher accuracy and fairness the better. We can see as accuracy drops when number of samples increase, fairness tend to slightly increase, showing that fairness and accuracy does have a trade-off.

However, the fine-tuning of the *Main Random Forest* model doesn't much change the final results of our model, as our post-processing method mainly do the work.

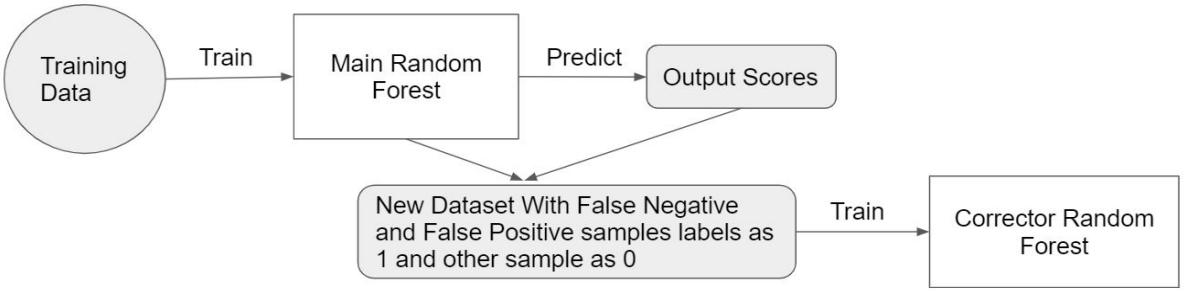

Figure 2: Training the two Random Forests

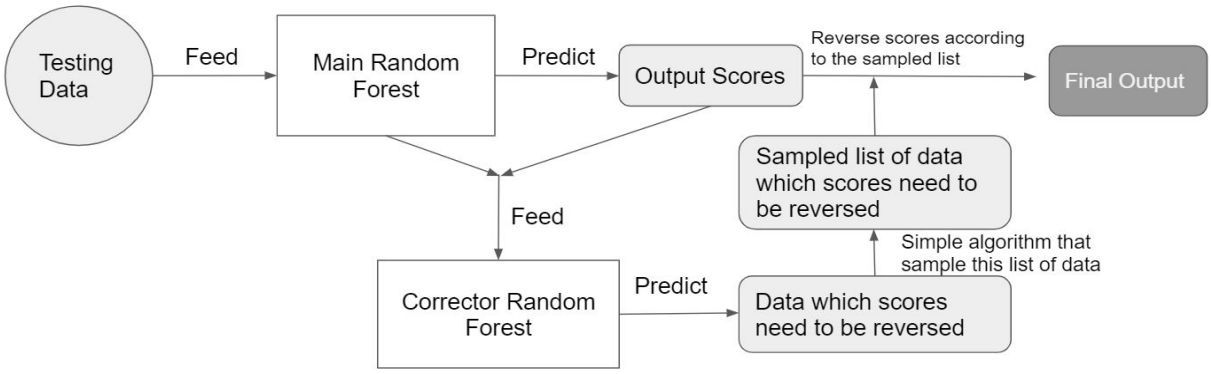

Figure 3: Predicting the final outcomes using the two Random Forests

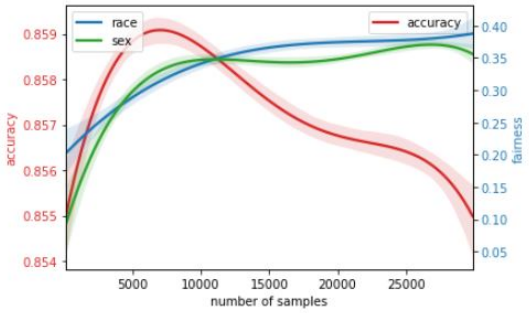

Figure 4: Fine-tuning with number of samples

4.2 Post-processing method's Improvement Compared to not Using the Method

Figure 5 and *Figure 6* respectively shows the accuracy improvement and fairness improvement of using our post-processing models, comparing to not using our models (i.e plainly using the *Main Random Forest*). The results are based on 10 random testing, with each time re-split the training and testing data.

We can see that there is a steady increase in accuracy, indicating that our *Corrector Random Forest* is relatively accurate and stable while reversing the outcome labels. The fairness improvement quite volatile, but still an improvement is seen. *Figure 7* shows the improvement of accuracy and fairness.

This comparison indicates that our *Corrector Random Forest* is stable and accurate when choosing False Negative or False Positive samples to be corrected. In our experiments, we find that the *Corrector Random Forest*'s precision could be more than 95%, in some cases 100%. And the recall, on average, is 92%. The relatively high accuracy for the *Corrector Random Forest* to identify the False Negative samples is the foundation and main reason of stability of our models.

However, the *Corrector Random Forest* is not all the post-processing method we used here. After the *Corrector Random Forest* identified the data that need to be reversed, we sampled these data points according to a certain ratio between male and female based on the False Negative rates of the training set. To prove its efficiency and necessity, we introduced two other baseline models for comparison.

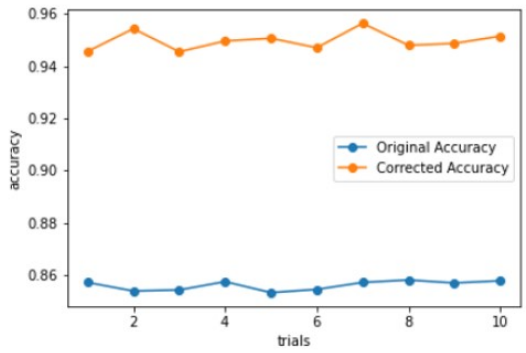

Figure 5: Improvement in Accuracy Compared to not using the Model

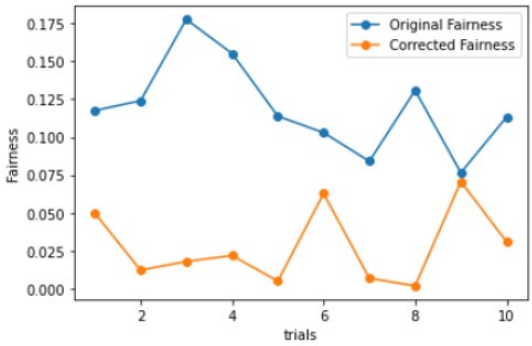

Figure 6: Improvement in Fairness Compared to not using the Model

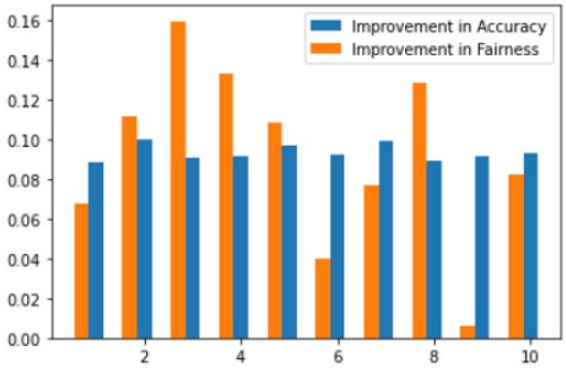

Figure 7: Improvement in Accuracy and Fairness in bar plot

4.3 Two Other Baseline Models

We designed two Baseline models to evaluate the efficiency of our Simple Algorithm. The baseline models are *Silly Baseline* and *Smart Baseline*.

4.3.1 Baseline Setups

Suppose that before sampled by our simple algorithm, there are *count_male_pre* and *count_female_pre* number of male and female outcomes identified by *count_male* and *count_female* respectively. After sampled by our simple algorithm, there are *count_male* and *count_female* number of male and female outcomes respectively.

The *Silly Baseline* model random flips *count_male* and *count_female* outcomes without using any results from *Corrector Random Forest*. This is used to show the efficiency of the *Corrector Random Forest*.

The *Smart Baseline* model flip all outcomes identified by the *Corrector Random Forest*, without using the simple algorithm for sampling. This can show whether the Simple Algorithm can improve the result.

4.3.2 Results

The fairness results based on the average of 50 random testing is shown on *Figure 8*. In the graph, the lower fairness the better, an fairness score of 0 indicates perfect fairness. Original fairness refers to not using the model at all. The *Silly Baseline* has extremely bad performance in fairness, further proving that our *Corrector Network* is necessary and useful in predicting the False samples. *Smart Baseline* shows an improvement in fairness, however, our model shows a better performance with regards to fairness. This shows that our Simple Algorithm that samples the list of data identified by the *Corrector Network* is useful in improving the fairness.

However, there is a fairness-accuracy trade-off when we use the Simple Algorithm. *Figure 9* shows the accuracy of the baseline models and our model. We observe that though previous our model has a better performance with regards to fairness when compared to *Smart Baseline*. For accuracy, our model has a worse performance than *Smart Baseline*. This is also expected, because our *Corrector Random Forest* is highly accurate in finding the False Negative samples. Following by reversing all the data it identified to be False will result in an increase in accuracy. However, as we also want to ensure fairness, our Simple Algorithm steps in

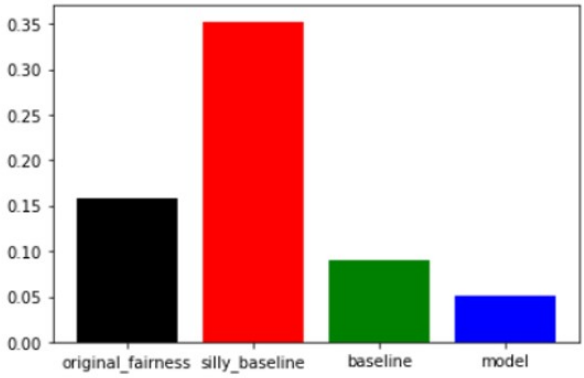

Figure 8: Comparison of Fairness between the baseline models and our model

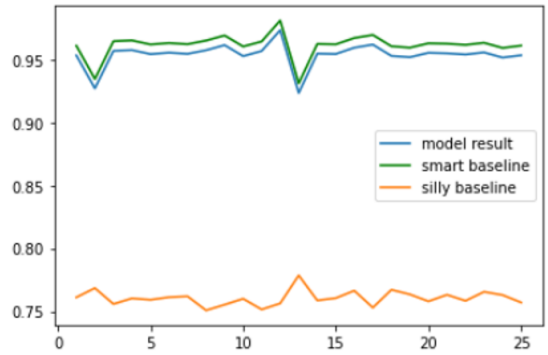

Figure 9: Comparison of Accuracy between the baseline models and our model

and correct fairness by sampling the data according to a ratio of male and female counts. This will foreseeable hurt accuracy.

These two baseline models respectively gives us these conclusions.

- ORIGINAL FAIRNESS Without using our model, the fairness performance is worse. This could justify that our model could improve fairness.

- SILLY BASELINE The Silly Baseline reverses outcomes without using our *Corrector Random Forest*, but only using the count numbers we have using the Simple Algorithm. The result is a tremendous decrease in both fairness and accuracy. This shows that our *Corrector Random Forest* is necessary in this model. The Simple Algorithm's sampling method is not the only force that balance the fairness.

- SMART BASELINE The Smart Baseline reverses outcomes using the *Corrector Random Forest*, but choose not to sample using the Simple Algorithm and blindly follow the *Corrector Random Forest*. The result is a slight increase in accuracy and a decrease in fairness. This makes us we believe that the Simple Algorithm enables a trade-off between fairness and accuracy. As the Simple Algorithm stops certain labels to be reversed, it will increase fairness, but inevitably hurt the high potential of improving accuracy by flipping those outcomes.

5 Conclusion

In this work, we proposes a post-processing method for Random Forest Model that could improve both fairness and accuracy for the Adult Income dataset. The post-processing model consists of a *Corrector Random Forest* and a Simple Algorithm that takes the fairness pattern in the training set into account.

By using several baseline models, we show that each steps is useful and necessary in our model. Specifically, *Silly Baseline* indicates the necessity of *Corrector Random Forest*, and *Smart Baseline* validates the usefulness of the Simple Algorithm. The empirical results shows that our resulting model gives an improvement in fairness and accuracy.

However, there is not much theoretical basis we could use to explain why and how this model worked. Our intuition is that, the two Random Forest models could be understood as a ensemble algorithm that could correct itself during training. And our Simple Algorithm, by sampling the data outcomes to be reversed according to a male-female ratio, enables a trade-off between fairness and accuracy. Understanding this model from a more theoretical requires a deeper understanding in ensembles algorithms.

Our research also leaves spaces of future improvement. As we mainly tested on Adult Income Dataset, it is possible to apply the same algorithms to other datasets and see its performance. There are also some stability issues when coming to this model. In 5% of the cases, the model will show an extreme result (i.e fairness suddenly deteriorates). Our understanding is that due to randomness, the *Main Random Forest* could provide a unusually fair model, leaving our model nothing to do but hurts the fairness. However, more experiments could be provided to evaluate this randomness.

6 Acknowledgements

We would thanks to Professor Guangyu Xia at New York University Shanghai for help guiding this project. Also, we express gratitude for NYU Shanghai's Dean Undergraduate Research Grant for funding this project.

References

Moritz Hardt, Eric Price, and Nati Srebro. 2016. Equality of opportunity in supervised learning. In *Advances in neural information processing systems*, pages 3315–3323.

Daniel McNamara. 2019. Equalized odds implies partially equalized outcomes under realistic assumptions. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 313–320.

Dana Pessach and Erez Shmueli. 2020. Algorithmic fairness. *arXiv preprint arXiv:2001.09784*.

Novi Quadrianto, Viktoriia Sharmanska, and Oliver Thomas. 2019. Discovering fair representations in the data domain. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8227–8236.

Pedro Saleiro, Benedict Kuester, Abby Stevens, Ari Anisfeld, Loren Hinkson, Jesse London, and Rayid Ghani. 2018. Aequitas: A bias and fairness audit toolkit. *arXiv preprint arXiv:1811.05577*.

Betsy Anne Williams, Catherine F Brooks, and Yotam Shmargad. 2018. How algorithms discriminate based on data they lack: Challenges, solutions, and policy implications. *Journal of Information Policy*, 8:78–115.