Hey, ML engineer! Is your model fair?

Jakub Wiśniewski[1], Przemysław Biecek[1,2]

[1]Faculty of Mathematics and Information Science, Warsaw University of Technology

[2]Faculty of Mathematics, Informatics, and Mechanics, University of Warsaw

Introduction

We live in a world that is getting more unequal each day. In some parts of the world, the differences and inequalities between races, ethnicities, and sometimes sexes are aggravating. The data we use for modeling is in the major part a reflection of the world it derives from. And the world can be biased, so data will likely reflect that and the model will become biased. To address this issue we made tools available in **both R and Python**, that enable ML engineers and data scientists to easily check if their classification model is biased and visualize that bias from different perspectives. They are a great aid in complex analysis of model discrimination against certain groups of people. They do not only work with binary sensitive attributes but also with non-binary ones and their intersections. We made our tools with belief that the process of bias detection should be effortless and the results of a such process clear and easy to interpret and using them is a step for more responsible ML.

Checking fairness

We introduce the **R fairmodels package** [1] that uses **DALEX** [2] package and fairness module in **Python dalex package** [3]. Packages enable model agnostic approach to bias detection.

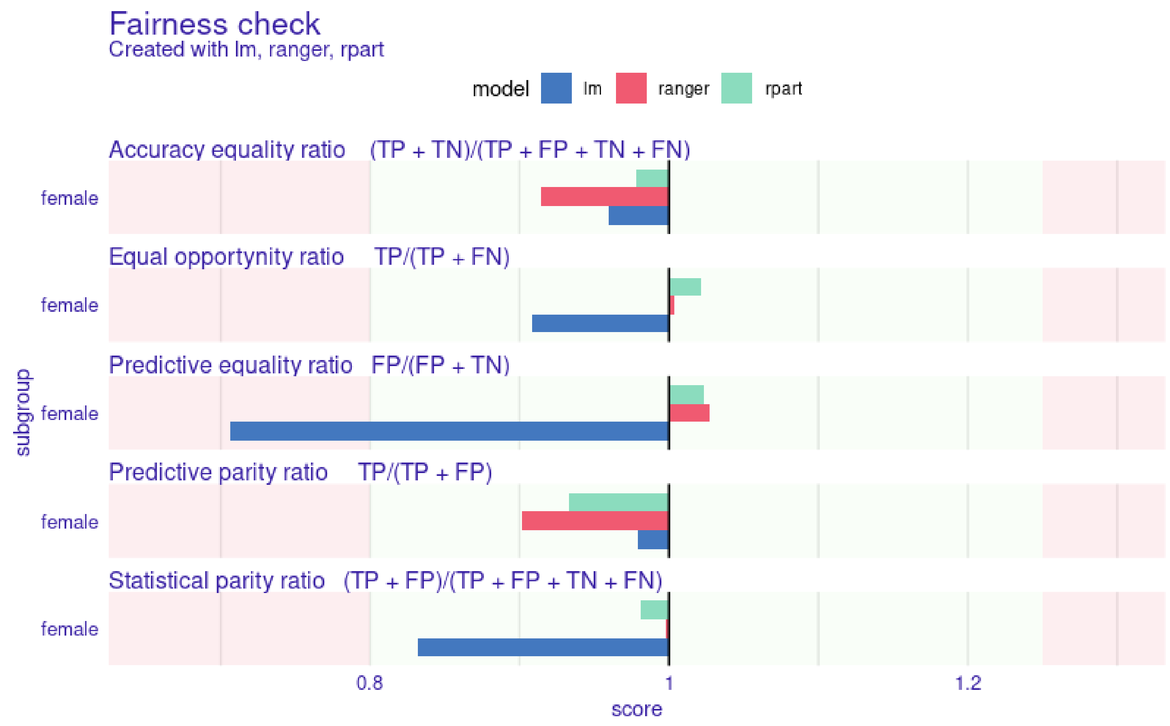

Figure 1: Output of *fairness_check() function in* ***R fairmodels package*** [1]. If a bar reaches red field that means that there is bias in particular metric (according to four-fifths rule[4]. The values of metrics are divided by values of privileged subgroup - here male. The closer the bars to 1, the better.

The main functionality of both packages is function (or method) *fairness_check()* that computes popular fairness metrics [5]. The plots are based on german credit data and models from popular R and Python packages. To aid in the visual diagnosis of discrimination we introduce *metric scores* plot which shows real metric values where.

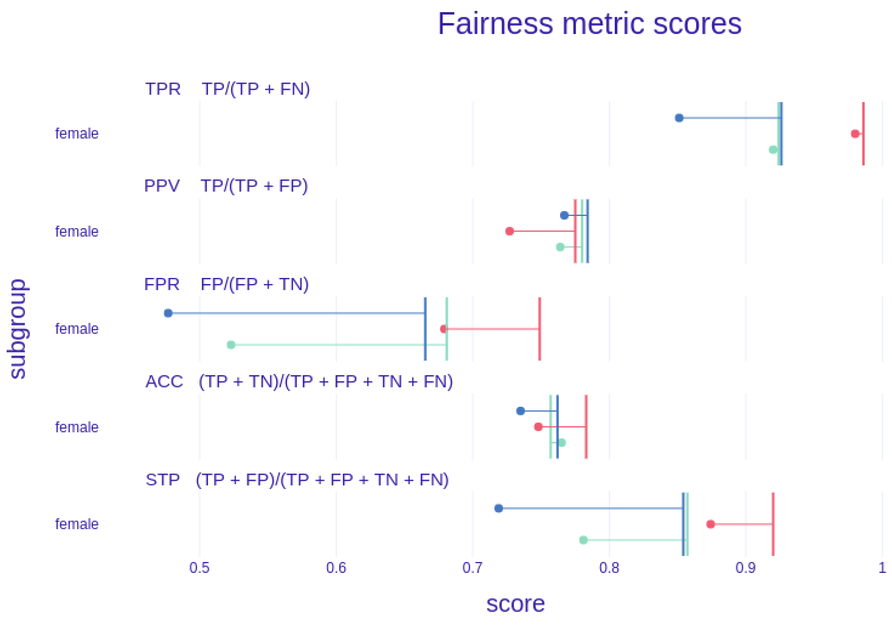

Figure 2: Metric scores plot in **Python dalex package**. The vertical lines denote the value of the metric for a privileged subgroup. The closer the dots are to lines the better.

Methodology

The user along with a classification model provides two things:

- protected vector which is an array that denotes association with a sensitive attribute like gender, nationality, disability, or intersections of those (called subgroups).
- privileged subgroup, which is an element of protected vector suspected of biggest privilege.

The metrics are computed from a confusion matrix for each subgroup (a, b, ...) where one of them is privileged. The fairness boundaries in Fig 1 are computed using epsilon value which by default is 0.8 to adhere to the four-fifths rule [4] . The model is considered to be fair in terms of metric M if

$$\forall_{i \in \{a,b,\ldots\}} \quad \varepsilon < \frac{M_i}{M_{privileged}} < \frac{1}{\varepsilon} \tag{1}$$

Other visualizations

Another way to tackle the bias visualization problem is the usage of *parity loss* which is a way to summarize metrics across subgroups

$$M_{parity\_loss} = \sum_{i \in \{a,b,\ldots\}} \left| \ln\left(\frac{M_i}{M_{privileged}}\right) \right| \tag{2}$$

It enables us to have only one value for metric and to look at bias from different perspectives. Of course the higher the metric the more biased the model is.

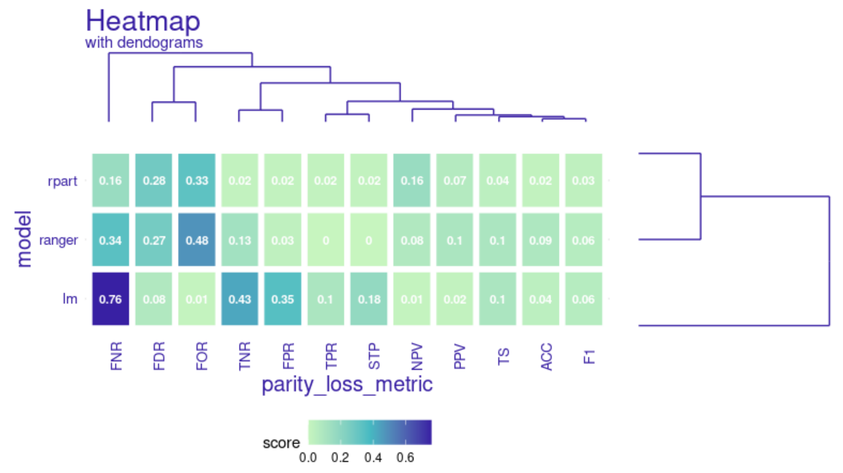

Figure 3: Fairness heatmap plot in **R fairmodels package** [1]

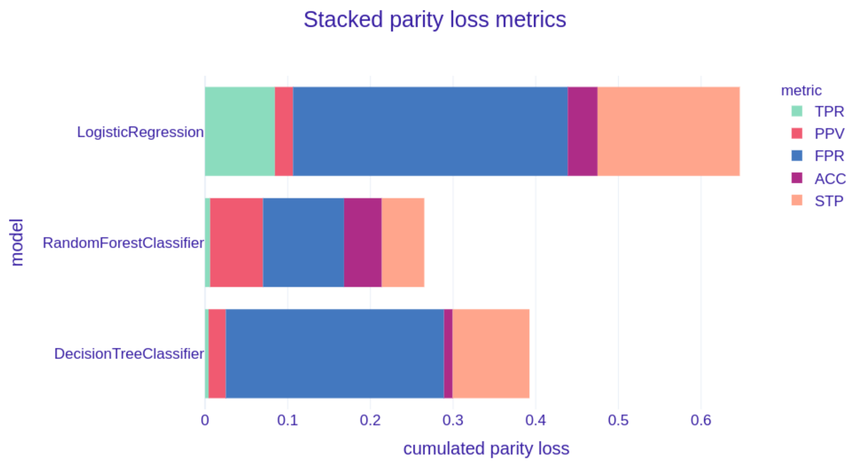

Figure 4: Stacked parity loss metrics plot in **Python dalex package** [3]

References

[1] Jakub Wiśniewski and Przemysław Biecek. fairmodels. URL https://modeloriented.github.io/fairmodels/.

[2] Przemyslaw Biecek. Dalex: Explainers for complex predictive models in r. *Journal of Machine Learning Research*, 19(84):1–5, 2018. URL https://jmlr.org/papers/v19/18-416.html.

[3] Hubert Baniecki, Wojciech Kretowicz, Piotr Piątyszek, Jakub Wiśniewski, and Przemysław Biecek. dalex: Responsible machine learning in python, 2020. URL https://dalex.drwhy.ai/.

[4] Code of Federal Regulations. Section 4d, uniform guidelines on employee selection procedures (1978). URL https://bit.ly/2IeOgde.

[5] Sahil Verma and Julia Rubin. Fairness definitions explained. FairWare '18, page 1–7, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450357463. URL https://doi.org/10.1145/3194770.3194776.

Acknowledgements

This work was financially supported by the National Centre for Research and Development in Poland, Grant POIR.01.01.01-00-0328/17

- ✉ jakwisn@gmail.com