Explanatory package for "Search-based Automatic Repair for Fairness and Accuracy in Decision-making Software",

This on-line appendix is supplementary to the paper entitled "Search-based Automatic Repair for Fairness and Accuracy in Decision-making Software", which has been submitted for review. It contains the raw results, Python code for the proposed approach, and scripts to replicate our experiments.

This `README` file describes the structure of the provided files (source code and results). as wel information on the content of this package.

For more detailed instructions on how to install datasets and required libraries, refer to the `INSTALL.md` file.

Content

"Scripts" directory

This folder contains files used to run bias mitigation methods and obtain their performace (accuracy, fairness). "utility.py" provides additional functionality to obtain datasets from AIF360, and writing results to files. The two folders "LR" and "DT" contain implementations of the three existing bias mitigation methods used for the comparison in RQ2. The files "approach_lr.py" and "approach_dt.py" provide implementations of our post-processing approach for Logistic Regression and Decision Trees respectively. Default-COMPLEX.ipynb and Default-NN.ipynb include experiments with complex classification models.

"Results" directory

This directory contains all the experimental results to answer the research questions in our paper.
The result files are organized according to the RQ they answer.

1. RQ1.xlsx: Contains the fairness and accuracy achieved by our approach on the four datasets for LR and DT. Additionally, the wilcoxon test results (p-values) and effect sizes, in comparison the default classification model are provided.
2. RQ2-1.xlsx: Summarizes the performanc of the existing post-processing bias mitigation methods and compares it to our method and the default classification model.
3. RQ2-2.xlsx: Summarizes the performanc of the existing pre- and in-processing bias mitigation methods and compares it to our method and the default classification model.
4. RQ3.xlsx: Wilcoxon test results that are used to generate Table 4. Colours represent win (green), tie (yellow), red (loss).
5. LR: This directory contains results for the performance of our post-processing approach on Logistic Regression. The performance is evaluated for each dataset and protected attribute and repeated for 50 different datasplits and 30 runs for each datasplit. The file is structured as follows: A line indicating "Trial x" to determine the datasplit, "Run y" to determine one of the 30 runs.

Following a "Run y" statement, three line follow that show the performance of our method (Line 1: Train set, Line 2: Test set, Line 3: Validation set). Each line contains four numbers: Accuracy, SPD, AOD, EOD. Filenames follow this pattern: "classifier_dataset_attribute_metric". Whereas "metric" states the fairness metric that has been used during optimization.

6. DT: Just as the "LR" directiory, this directory contains results for the performance of our post-processing approach on Decision Trees. The file structure is identical, with one difference: each "Run y" line is followed by a line "Data a b", where "a" shows the DT size before optimization, and "b" the size after optimization.
7. Benchmarking: Results for existing methods from the AIF360 framework.
8. Complex Models: Performance achieved by Neural Networks, Gradient Boosting and Random Forests.
9. LR-Parameters: Detailed results for different modification operators for LR under different weights.
10. RQ2-Boxplots: Complete collection of boxplot images as reported in RQ2.

We note that reported results are either in the format of four numbers per line (Accuracy, SPD, AOD, EOD) or seven numbers per line (Accuracy, SPD, AOD, EOD, Precision, Recall, AUC). We are in the process of unifying the format.

"Example.ipynb"

This jupyter notebook provides an example on how our post-processing approach can be implemented and shows how accuracy and fairness metrics are improved.

This purpose of Example.ipynb is to show how our approach could be used by practicioners.

"Code" directory

This directory contains the source code used in "Example.ipynb". It contains the following files:

1. utility.py: Additional functions to write content to files and obtain datasets from AIF360.
2. optimize.py: Functions that can be used to optimize LR and DT models. "optimize_lr" is used for LR models, "optimize_dt" for DT models.
3. Benchmarking: This directory contains scripts to run existing bias mitigation methods from the AIF360 framework. The classification model used is determined by passing a parameter.