Project report: German credit score dataset

Hanna Foerster
Department of Computer Science
Durham University
Durham, UK
hanna.foerster@durham.ac.uk

***Abstract*— As AI becomes increasingly involved in decision processes, it will be important to deal with Bias in AI. This is an investigation of one method to debias a dataset.**

I. PROJECT PROPOSAL

The project I intend to implement is a data debiasing algorithm proposed by Feldman [1]. This project will particularly suit the submodule bias in artificial intelligence because minimising bias in a dataset in the pre-processing stage will lead to more fairness in the machine learning models trained with this data.

First, I will give an overview as to why I want to do this project. Machine Learning (ML) is a very helpful tool that in general helps us make faster and better decisions. The idea of ML is to train a system by giving it lots of data, so it will learn the rules from this data instead of humans having to feed the system a set of rules which should handle every possible input [2]. Specific areas in which ML is used is in decision processes such as employment or admission, but also prediction of the development of diseases or weather. However, issues arise if we believe that ML driven decision processes are bias-free, because compared to subjective human decision processes, ML models seem more objective. Barocas and Selbst [3] in fact state that the concept of ML aided decision processes itself is a discrimination. It is a method to rationally distinguish people from each other and group into subgroups of people that seem statistically most similar. So, unintentionally problematic discrimination will happen when these subgroups are based on legally protected classes, such as gender, race and age.

Biases can be created in different steps of the process, such as in data collection, data processing, training and evaluation. The fallacy in general is to believe that a dataset can represent the world. The 'dataset bias' as Tommasi et al. [4] describe in their research comes from a dataset's inability to capture the real world. Hence, a 'sample selection bias' arises, where in classification problems for example, the distribution of classification labels conditional on values of features are different in each distinct dataset. Moreover, a bias could also already be present in a dataset because of historical discriminations against a group of people. In the processing stage data can be falsely aggregated to represent a very diverse group of instances. Biases can also arise by under- or overrepresentation of instances in a dataset and thus training a model with this biased data. And even in the last step after an ML model is deployed for use, biases can arise by misinterpretation of the results.

Minimising bias then will play a key role for the future use of ML. The debiasing can be done either before training an ML (pre-processing), while training the ML (in-processing) or after an ML model has been trained (post-processing). As we touched upon in the lectures, AIF360 recommends correcting the bias at the earliest stage possible, since this gives the most flexibility to correct the bias as much as possible [5]. Therefore, I am choosing to implement a pre-processing method. The particular method I will concern myself with is the Disparate Impact Remover (DIR) by Feldman [1] and it is a method to reduce demographic parity on minority groups by changing feature values of a dataset. Other pre-processing methods exist, such as reweighting, in which sample weights are recalculated until the ML model is bias-free, or resampling, which is based on the expected probabilities to meet demographic parity proposed by Kamiran et al.[6]. I have chosen to implement Feldman's method because the transformation process seemed simple enough to be transparently laid out even to the general public. If not everyone understands these transformation processes, they will not be able to judge whether the resulting ML model is fair and thus will not be able to trust in it.

I will be working with the German credit score dataset. This is a human-centric dataset and is important to be bias-free because getting a good rating directly affects the likelihood of getting a loan and reducing financial hardship or ensuring long-time financial stability. Feldman for example has found age discrimination in this dataset [1]. I will clean and pre-process the dataset first, to then be able to analyse it in terms of demographic parity. The programming language I am thinking of using is python and the sklearn package will be helpful for data pre-processing and implementing a machine learning model. As suggested in task 2, summary statistics about each subgroup in each feature can then be made to observe significant biases between subgroups. Without handling these biases, I will then train an ML model on a subset of the data and observe the performance in terms of accuracy, balanced accuracy and bias measures suggested by Feldman. I propose to use the same ML algorithm as in Feldman's research paper, which is the SVM algorithm. Finally, I will implement the DIR and test its performance by stratifying the dataset into a majority and a minority group, changing the dataset with the debiasing algorithm, and then applying the same ML algorithm. The same measures as in the unchanged dataset will be observed and then compared with the results of the original dataset. For better comparison purposes it will be good if graphs can be created in this step with the help of the seaborn and matplotlib packages. Lastly, I will compare my results with Feldman's results and judge the overall performance of this debiasing method.

II. PROJECT PROGRESS REPORT

A. Data Analysis

In this paragraph I will describe the data and a bias that I observe when splitting the dataset into demographic subgroups. The dataset is taken from UC Irvine's ML repository [7] and consists of 20 columns. 7 of these are of the numeric data type, while the rest is categorical data. The numerical data was converted from object type to integer type as a first step. Protected classes by the US federal anti-discrimination law [8] that are included in this dataset are the following: 'Personal status/sex', 'age' and 'foreign worker'. I binned the values of the feature 'age' into 'young' if the value was 25 or smaller and as 'old' for the rest of the values. The

classification of age values into these subgroups was proposed by Kamiran and Calders [9] to test the demographic parity in the German credit score dataset. The following features were further dropped from the dataset: 'purpose', 'other debtors', 'other installment plans', 'telephone' and 'foreign worker'. These features were excluded in the analysis because they are neither of numeric nor ordinal data type and cannot be dealt with the DIR.

I will present some of the results obtained when comparing the subgroups 'young' and 'old' as defined above with each other. For more detailed figures please refer to the python program, which prints out all numbers that were suggested to be compared between the two subsets in Task 2. More people in the age group 'old' had a credit history with priors of delayed payback or a critical account, with 42 % of bad credit history for people classified as 'old' in comparison to only 21% of young people. However, the mean credit amount for young people was 3003 DM compared with 3334 DM in the old age group. 84% of older people had no other installment plans compared to 67% of younger people. A difference could also be seen in terms of housing where 75% of old people lived on a self-owned property, but only 56% of the young age group. There were also more unskilled people in the younger subgroup and significantly less highly qualified employees. Lastly, only 110 of the subgroup of 190 younger people (58%) compared to 590 of the 810 people in the older group (73%) were classed in class 1, meaning that they have 'good' credit status.

Overall, this information suggests that there is a bias in the dataset with more people of the older age group receiving a 'good' credit status. I believe that this has happened, because the older people observed in this dataset had relatively more financial stability. For example, as described above, in average an older person in this dataset was less likely to have another installment plan, likelier to live on self-owned property and likelier to have a highly qualified job.

B. Conventional implementation

I chose to implement the Linear Support Vector Classification model from the sklearn library. Support Vector Machines (SVM), a model of supervised learning, relies on the notion that each instance in the dataset lies on a point on a n-dimensional hyperplane based on the feature values. For a binary classification task, such as the one at hand, the SVM learns from the classification of the training instances and generates a maximal margin which separates the classes. Then each test instance lying on the hyperplane will lie on one side of the margin and will be classified like the training instances on that side of the margin. The linear SVM especially only chooses a linear function as the function describing the margin. I have chosen this model, because it works well on small datasets and the linear kernel with its hyperparameter settings is especially advantageous to prevent overfitting. The computation cost is also relatively small compared to other algorithms and it is relatively easy to interpret the results. Initially, I tried using non-linear kernels on the data, however it was very difficult to fit the data such that it was not overfitting and after debiasing with different strengths (see Fair machine learning implementation section), some kernels did not work well anymore and using a grid of parameters, were different kernels and regularization parameters could be chosen, lead to incomparability between the models for datasets repaired at different strengths.

Before applying the ML model, I first checked for missing data in each column. Then after seeing that there was no missing data, I applied a label encoder on each categorical column of the dataset and split the data into training and test set in a ratio of 2/3 and 1/3 respectively. Then, both the training and testing set were standardized before the SVM was applied. To fit the SVM properly, it was important to choose the right hyperparameters in the training process. I used the 'l2' penalty and 'hinge' as the loss function as this was suggested by Feldman [1]. The parameter classweight was additionally set to 'balanced' because the dataset was very imbalanced in the number of instances classed with good and bad credit rating. Moreover, the best regularization parameter was chosen from a grid with the GridSearchCV function from the sklearn library.

The results I obtained were measured by four different methods. The first measure was the model accuracy which is the rate of true positives and true negatives. The second measure I used was the balanced accuracy score (a.k.a. utility) it is calculated by taking the mean of the true positive rate and the true negative rate. The third measure I used is called the Zemel Fairness [10] and it is calculated by the rate of positive outcomes in the privileged group subtracted by the rate of positive outcomes in the minority group. I have flipped this measure to be 1-Zemel Fairness, so that 1 is obtained when the model is fair and 0 is obtained when the model is very biased. The last measure is called the Disparate Impact Score. Feldman defines it as the rate of positive outcomes for the majority group divided by the rate of positive outcomes for the minority group. Here again 1 is obtained when the model is fair and 0 is obtained when the model is very biased.

In the conventional implementation the model accuracy was at about 0.73 and the balanced accuracy rate at about 0.76. The Zemel Fairness and Disparate Impact Score were calculated between the young and old age group, where the old age group is the more privileged age group. They were 0.72 and 0.55, respectively. This shows that the bias in the ML model is very high. Next, I subsampled the dataset in a way to ensure that each age group and gender was represented the same amount. The smallest group was the group of young males of which there were only 85. Hence for all other subgroups 85 samples were randomly picked and then the subgroups were put together as a new dataset on which the ML algorithm was trained. The SVC model lost accuracy when training with this new dataset, the model accuracy and the balanced accuracy were at about 0.67. However, the Zemel Fairness and Disparate Impact scores rose to 0.78 and 0.66, respectively. So, this model trained with the new resampled dataset is less biased towards giving the older age group a good credit rating and hence a fairer ML model. This fits with my first assumption that a fairer dataset will produce a fairer ML model.

C. Fair machine learning implementation

I have chosen to implement the Disparate Impact Remover (DIR) by Feldman [1]. The analysis of the dataset in Task 2 has suggested that a bias between the younger and older age group is present in the dataset. The conventional implementation in Task 3 has shown that a model trained on such a biased dataset will result in a biased ML model. Resampling the dataset has shown that a fairer ML model can be produced by training on a less biased dataset. The data debiasing method from Feldman will determine if a bias-free dataset will lead to a bias-free ML model.

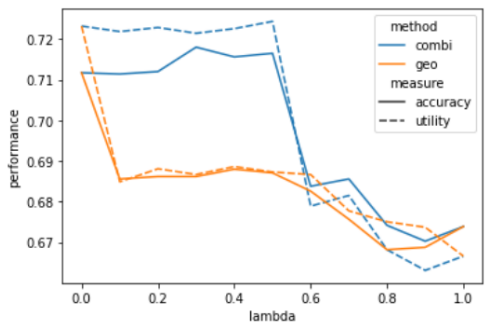

Fig. 1. Graph of performance-repair amount correlation

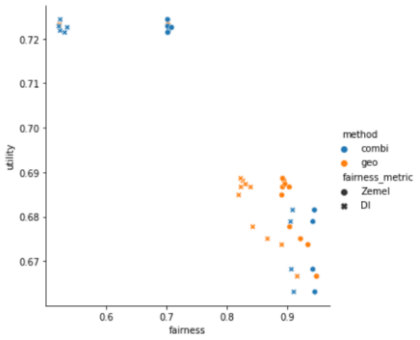

Fig. 2. Graph of utility-fairness correlation

The DIR is a debiasing method applied at the pre-processing stage which changes the values in the dataset while maintaining the rank-ordering within each column and group. All columns which are not protected attributes are independently repaired by the algorithm. The goal of the algorithm is that the distribution of each feature is the same for all subgroups identified from a protected column. First, for each column all unique values were put into a list and sorted. For later access, the sorted list for each column was saved into a hashmap sorted_lists with the column names as keys. Then another hashmap was created with the unique values as key and the position in the sorted list as value. This hashmap called index_lookup by Feldman [1] was stored in the hashmap index_lookups with the column names again as keys. In the German credit dataset, the bias that I wanted to remove was the age bias. Hence, I stratified the dataset into the two age groups 'young' and 'old'. Then the size of the smaller group ('young') was taken as the number of quantiles, so that the number of quantiles was maximised while ensuring that each quantile had at least one instance from each group. To have equal distributions in each column for both subgroups, the following process was applied to each column individually. The median value in every quantile was taken from each subgroup and then the median of the medians was taken as the overall median value for that quantile. For even-length lists the lower median value was taken. Now, each instance in the quantile was corrected based on this median value by either the 'Combinatorial' repair method or the 'Geometric' repair method. The 'Combinatorial' repair method uses the index g of the original value of the instance and the index t of the median target value in sorted_lists[column] and a 'repair index' is calculated with the following formula:

$$repair_{index}\ ri = round\left(g + \left(\lambda * (t - g)\right)\right) \text{ [1]}$$

The value at the repair index in the sorted_list of the column is then taken as the new value for the instance. The 'Geometric' method directly calculates the repair value from the original value and the target value with the following formula:

$$repair_{value}\ rv = \left((1 - \lambda) * orig_{val}\right) + (\lambda * targ_{val}) \text{ [1]}$$

The λ in both repair methods is the repair amount and lies between 0 and 1, where 1 indicates a full-repair and 0 is no-repair.

22 new datasets were created with the DIR for the two repair methods with λ values from 0 to 1 in steps of 0.1. The Linear SVM classifier described in Part 3 was retrained for each new dataset on a training set obtained from a train-test set split of ratio 2/3, 1/3 as in Part 3. To ensure randomness for each new dataset the train-test set split and following training of the Linear SVM was repeated with a different pseudo-random number 10 times and the mean of the measures from each experiment was taken. This procedure and the above algorithm are from Feldman's paper "Computational Fairness: Preventing Machine-Learned Discrimination" [1].

After implementing the above algorithm, I did not get the results that Feldman described in his paper. Feldman describes that with a full repair, each column in the new dataset contained the same value for the German credit score dataset. This was not the case in my implementation and the fairness was only slightly raised at first. After adding the 'class' to be repaired along the other columns, I obtained the graph shown in Figure 4, which shows that the probability of being classed as 'good credit' attributed by the SVM model to instances at each percentile rank was nearly equal for both subgroups. This was the goal of the repair and a full repair also leads to a very good Zemel Fairness and Disparate Impact Score, which in the average of 10 train-test splits were 0.948 and 0.916, respectively. The scores heavily depend on how the dataset is split into train and test set, because the dataset is not very large, so the graph [Fig. 4] is a better indicator of how well the dataset has been repaired. Figure 1 shows that the accuracy goes down as λ gets nearer to 1, in exchange to that however the fairness goes up [Fig. 2]. In Figure 2 it can also be observed that there is a spike in accuracy at λ=0.4, 0.5 for the 'Combinatorial' repair method. The data shows that this is only the case when both Fairness Measures perform badly, hence it seems as though there is a negative correlation between fairness and accuracy. Several aspects could have contributed to the difference in results to Feldman. First, he could have used a different method of choosing the number of instances for each quantile, since this is not clear from the pseudocode. Moreover, differences could have occurred by using slightly different versions of the Linear SVM classifier, since even changing the regularization constant C changes the predictive model significantly.

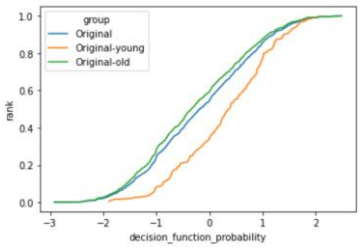

Fig. 3. Graph of percentile rank – probability of being classed 1 correlation given the original dataset

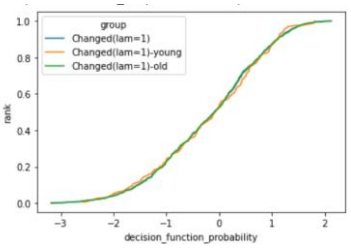

Fig. 4. Graph of percentile rank – probability of being classed 1 correlation given the fully repaired dataset

In conclusion, debiasing the dataset with the DIR has led to a fair ML model in respect to the age groups, but this seems to correlate with a loss of accuracy and utility.

REFERENCES

[1] M. Feldman, "Computational Fairness: Preventing Machine-Learned Discrimination", Haverford College. Department of Computer Science, 2015.

[2] M. Jordan and T. Mitchell, "Machine learning: Trends, perspectives, and prospects", Science, vol. 349, no. 6245, pp. 255-260, 2015. Available: 10.1126/science.aaa8415.

[3] S. Barocas and A. Selbst, "Big Data's Disparate Impact", California Law Review, vol. 104, no. 3, pp. 671-732, 2016. Available: 10.2139/ssrn.2477899.

[4] T. Tommasi, N. Patricia, B. Caputo and T. Tuytelaars, "A Deeper Look at Dataset Bias", CoRR, vol. 150501257, 2015. Available: http://arxiv.org/abs/1505.01257. [Accessed 2 May 2021].

[5] "AI Fairness 360", Aif360.mybluemix.net, 2021. [Online]. Available: https://aif360.mybluemix.net/. [Accessed: 02- May- 2021].

[6] F. Kamiran and T. Calders, "Data preprocessing techniques for classification without discrimination", Knowledge and Information Systems, vol. 33, no. 1, pp. 1-33, 2011. Available: 10.1007/s10115-011-0463-8.

[7] "UCI Machine Learning Repository: Statlog (German Credit Data) Data Set", Archive.ics.uci.edu, 2021. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data). [Accessed: 02- May- 2021].

[8] "Protections Against Discrimination and Other Prohibited Practices", Federal Trade Commission, 2021. [Online]. Available: https://www.ftc.gov/site-information/no-fear-act/protections-against-discrimination. [Accessed: 02- May- 2021].

[9] F. Kamiran and T. Calders, "Classifying without Discriminating", 2009 2nd International Conference on Computer, Control and Communication, pp. pp. 1-6, 2009. Available: 10.1109/IC4.2009.4909197 [Accessed 2 May 2021].

[10] R. Zemel, Y. Wu, K. Swervsky, T. Pitassi and C. Dwork, "Learning Fair Representations", Proceedings of the 30th International Conference on Machine Learning, vol. 28, no. 3, pp. 325-333, 2013. Available: http://proceedings.mlr.press/v28/zemel13.html},. [Accessed 2 May 2021].