ECS189G: Fairness in Machine Learning – Explicit Deweighted Features Approach

Student: Aditya Mittal
University of California, Davis
adimittal@ucdavis.edu

Professor: Norman Matloff
Course: ECS_189G
Department of Computer Science
University of California, Davis
matloff@cs.ucdavis.edu

June 2023

1 Abstract

Many different methods have been developed in order to address the growing concern surrounding fairness/bias in practical machine learning algorithms. In particular, several measures of fairness have been introduced, though many of them are extremely complex and quite specific to their respective algorithms. Moreover, the current literature surrounding fairness algorithms may require significant industry knowledge for each individual case. In this paper, we introduce the concept of Explicit Deweighted Feature (EDF): an approach to reducing the impact of proxies that are related to our sensitive variables in our algorithms. We will use the K-nearest neighbor algorithm; though, this approach can be extended to a number of different ML algorithms. We get empirical results using the German Credit dataset, and compare our findings with a published paper discussing fairness in ML with the same dataset.

2 Introduction

The growth of machine learning and its applications have developed rapidly in recent years, ranging from various different sectors in business to healthcare. The use of machine learning algorithms are more frequently being employed to guide critical decisions – decisions that can have potentially negative consequences on everyday consumers. As such, with the increasing use of these

algorithms, a new concern arises: the issue of fairness in machine learning. In particular, fairness in machine learning refers to the process of correcting and preventing algorithmic bias from protected attributes (race, gender, disability, etc.) in decisions made by the models. [2]

Within the finance industry, the use of machine learning algorithms have become increasingly common to help mitigate risks and optimize their decision making process. [4] Consequently, new restrictions have been imposed to help account for increased fairness in the decision making process of these algorithms. For instance, Basel Committee on Banking Supervision (BCBS) and the EU data protection regulations have led to an increase in research from switching to black-box machine learning to means of explainable machine learning [3]. They hope to ensure employing explainable machine learning can better help understand model accuracy, fairness, transparency, and bias of outcomes [5].

To ensure fairness in our machine learning algorithms, we need to ensure our algorithms are not biased towards any protected variables. These protected features commonly include race, religion, gender, marital status, etc. Disparate impact occurs when a particular protected class is disproportionately favored unintentionally by variables tied to protected attributes. As such, a naive approach employing fair machine learning would be to simply remove the sensitive variables from the dataset. Though, this approach raises the issue of "Fairness through unawareness." This notion assumes that if "we are unaware of protected attributes while making decisions, our decisions will be fair." [7]. However, protected features could be correlated with proxy variables – proxies refer to features or variables that are used to indirectly infer sensitive attributes and are used in the machine learning algorithm, ultimately resulting in potential bias.

With this, we have an inherent Fairness-Utility tradoff: an inherent tradeoff between fairness and utility. This states that as fairness is enforced in an algorithm, then accuracy tends to suffer. This, of course, depends on the metrics of fairness used and each individual case.

The primary objective of this paper is to present the methods practiced during ECS 189G: Fairness in Machine Learning to detect and mitigate potential bias. We compare our results to a research paper submitted to Tilburg University for a Master's program, submitted by Majda Lalla Kasmi [1]. Her paper is titled "Machine Learning Fairness in Finance: An Application to Credit Scoring." We compare our results using the German credit dataset; we are predicting a binary outcome, thus this is a classification problem. In this report, we use Explicitly Deweighted Features (EDF) to reduce the impact of potential proxies on our algorithmic decisions and develop a model while having the fairness-utility trade-off in mind.

The paper is organized as follows: Section 2 discusses several evaluation measures of fairness and utility; Section 3 details the proposed methodologies further; Section 4 highlights the results; Section 5 is our discussion.

3 Evaluation Measures

3.1 Evaluating Fairness and Utility

The inherent tradeoff between fairness and utility is an extremely important concept in research surrounding fairness in machine learning. Already, literature has been published discussing several measures of fairness, many of which are quite complex and require significant knowledge of the individual subject matter. In this paper, we use the following criteria for fairness:

$$\rho(\hat{Y}, S) = \text{correlation between predicted Y and S} \tag{1}$$

With this, we measure our fairness between our predicted Y and S using Kendall's Tau correlation coefficient to measure the relationship between our variables [6]. We use this measure as the correlation coefficient is measured between two continuous variables; though in many cases, it is often that Y or S can be categorical (ordinal or nominal) and we would then need a measure to compute the correlation. Thus, we use the Kendall's Tau correlation coefficient. For the purpose of measuring fairness, we want to get smaller values (closer to 0) to indicate our predicted Y is not associated with the sensitive variables.

In order to measure utility, we compute the test accuracy produced by from the holdout set from the machine learning algorithms. In particular, the qeML package provides a convenient framework to calculate the overall test accuracy which can be compared between different algorithms to compare utility.

To reduce our results sampling variability across different holdout sets from each run, we use ReplicaMeans() function across 25 runs to compute overall test accuracy and correlation coefficient measures.

3.2 Related Work in "Machine Learning Fairness in Finance: An Application to Credit Scoring"

The student paper [1] discusses several additional measures of fairness as detailed below:

1. Statistical Parity: This measure calculates the difference in probabilities between protected groups to be classified positively and the probability for the unprotected group to be classified positively.

$$\rho(\hat{Y} = 1, A = 0) - \rho(\hat{Y} = 1, A = 1) \tag{2}$$

   A positive score indicates positive discrimination, while a negative score indicates that discrimination against the protected group occurs. A score of 0 is ideal for fairness. In this case, the A = 0 refers to male group; A = 1 refers to the female group (defined as the protected group in the paper).

2. Disparate Impact: this is a ratio of positive classification between protected vs unprotected groups.

$$\frac{\rho(\hat{Y} = 1, A = 0)}{\rho(\hat{Y} = 1, A = 1)} \tag{3}$$

Score of 1 is ideal. Values below 1 indicate discrimination against the protected class. Values above 1 indicate positive discrimination. In this case, the A = 0 refers to male group; A = 1 refers to the female group.

4 Proposed Methodologies

4.1 Methods used in this paper

We make use of the functions provided by the qeML package. As we are focusing on classification, we have a variety of different options of functions available to us. For this report, we use the K-nearest neighbors (k-NN) algorithm. As noted earlier, we are trying to reduce the weight of potential proxies in our dataset (Explicit Deweighting Features) in order to avoid the issue surrounding "Fairness through unawareness". As such, we use the arguments provided in the qeKNN function in order to de-weight the proxies according to different values of D_i. We can then compare the fairness-utility tradeoff of the algorithms across different values of D_i and choose an appropriate value to de-weight the proxy with and create our algorithm. We're interested in finding two things, how predictive accuracy is affected with different measures of D_i and how our model compares to a K-NN algorithm with all the variables (including sensitive variables). Furthermore, we will compare the results from our findings with the Kasmi's paper in terms of changes in fairness and utility (using the measures detailed above).

It is important to note that in Kasim's paper, she did not account for proxies within their research. It was mentioned briefly, though none of the fairness processing methods actually take into account the impact of proxy variables in creating bias. In this report, we aim to learn the effects of these interaction terms in regards to the fairness-utility tradeoffs.

4.2 Related Work

The student paper [1] we compare our analysis to uses the logistic regression model. In order to enforce fairness, the author uses several methods as detailed below:

1. **Pre-processing**: Pre-processing refers to the processing of data before employing an algorithm to reduce the bias of sensitive features. Two methods of pre-processing are used.

a. *Suppression*: Here, the sensitive attributes are simply removed from the dataset.

b. *Massaging*: This method will relabel the observations to remove discrimination in the dataset.

2. **In-processing**: fairness constraints are employed during the algorithm in order to enforce model fairness.

3. **Post-processing**: Post-processing methods aim to achieve fairness but do so after running the algorithm. Several methods of post-processing are shown.

   a. *Equalized odds*: "$\hat{Y}$ (predictions) and protected attribute A independent conditional on Y (true labels). $P(\hat{Y} = 1|A = 0, Y = y) = P(\hat{Y} = 1|A = 1, Y = y)$. There should be equal true positive rates for both A = 0 and A = 1 as well as equal false positive rates. " [1].

   b. *Calibrated equalized odds*: In this method, calibration is taken into consideration when trying to enforce equalized odds.

   c. *Reject Option Classification*: In this method, probabilities closer to 1 are considered to be of higher degree of certainty. Based on this, a critical region is defined between 0.5 and threshold t. "Observations that fall in that region and are labeled reject. If an instance from the protected group is labeled reject, then it receives a positive classification, whereas those in the other group are negatively classified" [1].

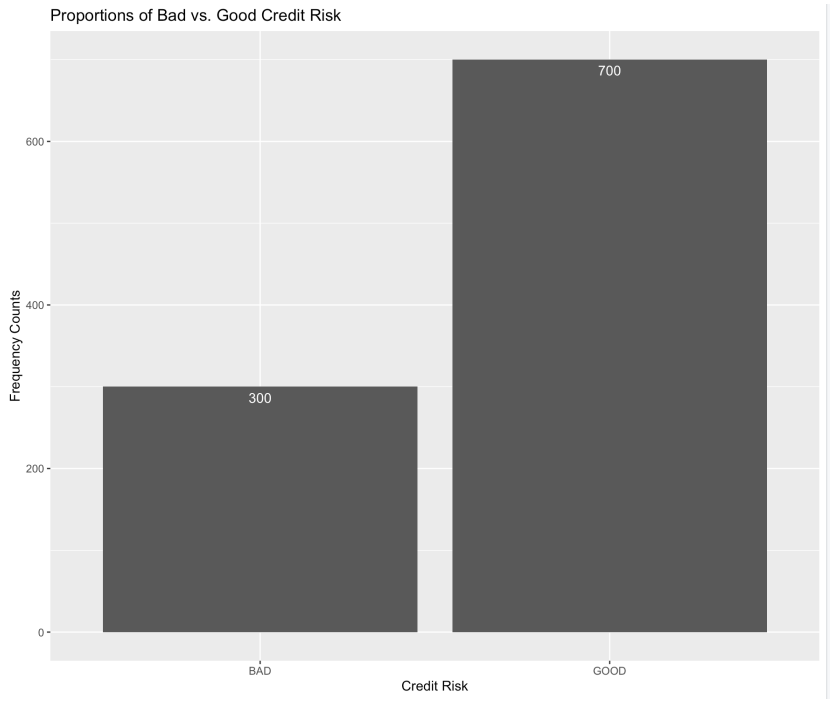
4.3 Dataset

The dataset used for this paper is the German Credit dataset acquired from UC Irvine's Machine Learning Repository and has been widely used across various different research studies. In particular, the dataset contains 1,000 observations of 21 variables – including the response variable "Credit Risk". The dataset includes several demographic variables describing the applicant's existing credit history, employment history, and credit requested. In particular, there are 3 potential sensitive variables present: Age, Gender, Foreign worker. The response variable, 'Credit risk', is a binary variable that determines whether the consumer is ultimately deemed "Good" for credit or "bad" for credit.

In our case, the dataset required very minimal pre-processing. Furthermore, since we have 20 potential predictor variables, we don't need to worry about the issue of overfitting in our model when using the training dataset. Furthermore, we drop the feature "Foreign Worker" as it was only present in 3.84% of the instances; this is the same pre-processing step taken as the student paper. Now, our dataset contains two sensitive features: Age and Gender.

From the distribution of credit risk (shown below), our response variable, we can see that the proportion of customers who were deemed "GOOD" credit is

significantly higher than those who were labeled as risky. More specifically, the proportion of customers labeled for "Good" credit accounts for 70% of our entire dataset. Due to the significant difference in the distribution of our response variable, a potential issue arises for many ML scientists: the issue of unbalanced data. In particular, there is a concern with unbalanced data such that, if there is much more data of one class (i.e. dominant class), then all of the future predictions will predict the dominant class. With this, it should make sense that our predictions from the dominant class, if the proportion of "good" credit is significantly higher and this dataset is representative of the population proportions. Thus, in this report, we choose to keep the existing dataset without balancing the distribution. In the student paper, the author uses SMOTE to create synthetic data of the minority class to balance out the data across our response variable. As such, this is a potential avenue where our results may differ right from the pre-processing stage.



5 Results

In this section, we discuss our findings from the K-Nearest Neighbors models to the logistic regression model from Kasim's paper and compare our results using different fairness measures. First, we first introduce two algorithms: KNN model with all the variables (including age and sex) and a model without age and sex (Pre-processing method: Suppression, as with the student paper). Note:

The first model may not be accepted in practical settings under the EU regulations due to the use of protected attributes in decision making, nevertheless it can serve as a good baseline to measure predictive power. In our model without sensitive variables, we have to be aware of the issue surrounding "fairness through awareness" as inclusion of proxies may still indirectly infer our sensitive attribute.

	Test Accuracy (25 runs)
Model with all variables	0.333
Model without sensitive variables 'Age' and 'Sex'	0.327

Our test accuracy across 25 runs was 0.333, indicating that our predictions were incorrect approximately 33.3% of the time. For the model without sensitive variables, our test accuracy is 0.327. Here, we can see that our test accuracy actually increased by 0.006, which was a surprising result since we actually removed two variables from our dataset. In this case, our results from the KNN model match with the logistic regression model in terms of utility – the model's predictive accuracy increased by 0.02 in data without the sensitive variables as opposed to the baseline model. Again, we have to be aware of proxies and finding them. Thus, we now create a model in which we exclude the sensitive variables, but also account for proxy detection and de-weighting to reduce their impact.

5.1 Choosing Deweighting Parameter

As mentioned previously, we are finding potential proxies in our dataset as they can be representative of S and be a potential cause of bias. To decide which features that go into the deweighting set, we need to find the variables that are good at predicting our sensitive variables. Here, domain expertise may be a factor along with employing many machine learning methods that have been developed already. For this report, we use exploratory analysis to find significant differences between our sensitive variables across different features.

Based on our findings, we find the relation between gender and present employment as follows:

	<1	1 <x <4	4 <x <7	<7	unemployed
Male	0.27741935	0.34516129	0.15161290	0.15161290	0.05652174
Female	0.12463768	0.33623188	0.18405797	0.29855072	0.07419355

It is evident that there is a substantial difference between present employment status and gender. In particular, the proportion of males is twice as likely to have under 1 year of current employment as opposed to female applicants. Alternatively, Male applicants are half as likely to have 7+ years of experience as opposed to female applicants. Moreover, the probability of an applicant being assigned as "GOOD" credit risk varies significantly by employment status, as shown below.

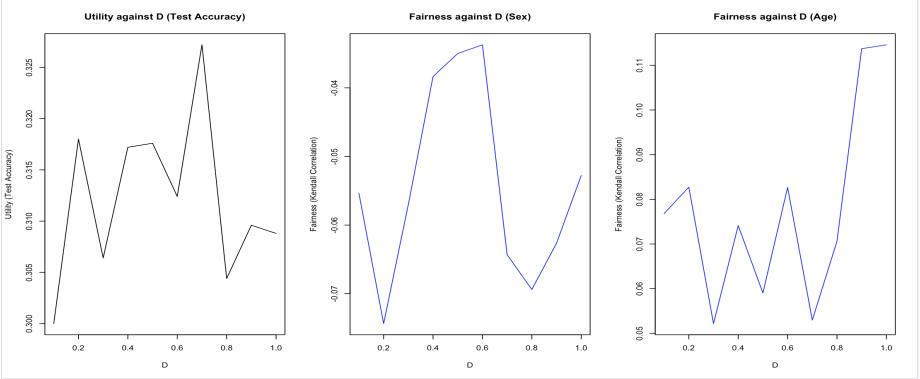
>1	1 <x <4	4 <x <7	<7	unemployed
0.5930233	0.6932153	0.7758621	0.7470356	0.6290323

Thus, we can see that present_employment_status is an excellent candidate to be considered as a proxy. We can now choose to de-weight the variable across different values of D_i, and measure the trade-off in fairness/utility. For the values of D_i, we select values between 0 and 1, by increments of 0.1. We are interested in finding an optimal balance in terms of fairness-utility trade-off with our model, then compare our empirical results with the student paper in accordance with the fairness measures provided as above.

5.2 Fairness/Utility Across D

Across 25 runs, we compute our fairness and utility values across different values of D_i. Below are the results from our simulated runs:

d	**Utility**	**Fairness_Sex**	**Fairness_Age**
0.1	0.3000	-0.05536874543	0.07681764215
0.2	0.3180	-0.07435814764	0.08273424074
0.3	0.3064	-0.05701408376	0.05218836144
0.4	0.3172	-0.03835116433	0.07412592098
0.5	0.3176	-0.03499170400	0.05904472290
0.6	0.3124	-0.03373157553	0.08264484653
0.7	0.3272	-0.06432409463	0.05297072443
0.8	0.3044	-0.06939058352	0.07062918026
0.9	0.3096	-0.06264802434	0.11371927518
1.0	0.3088	-0.05276502717	0.11457758419



We can see that, across different values of D_i, our test accuracy appears to vary around 0.3. Furthermore, we can see that the fairness (Kendall's correlation) values may also vary significantly for both Gender/Age and predicted Y across

different values of D_i. Here, we keep the fairness/utility trade-off in mind in selecting the value of D_i. Thus, in order to get an ideal combination of fairness and utility, we select the value of $D_i = 0.6$. We find that our test accuracy at $D_i = 0.6$ is 0.3124, indicating that our predictive power actually increased compared to our previous runs (we observed testAcc of 0.327 of the algorithm without Age & Gender; with full weight of Present_employment_status). Furthermore, Kendall's correlation coefficient is close to 0 for Gender and Predicted Y at value of -0.03. Similarly, the correlation value appears to be not too high for age and Predicted Y (about 0.08). Here, our simulated study indicates that Explicitly Deweighted Features can help employ new fairness measures, while also seeing a slight increase in predictive. This was an extremely surprising result, as our predictive power actually increased when deweighting our proxies across different values of D_i.

Using the Kasim's results from the suppression pre-processing methods, we can see how our results are similar with K-Nearest Neighbors and the tested logistic model. As opposed to the KNN function with all the variables, we can see that simply removing our sensitive variables actually slightly improved our predictive performance – though these improvements were not significant. This result matched with the suppression pre-processing results under the logistic model of the student paper as well. Additionally, we also compare the fairness measures as defined by statistical parity and disparate parity. In particular, we measure the scores of the algorithm of using all variables and model using EDF with our proxy 'Present employment status' de-weighted at $D_i = 0.6$:

	Statistical Parity	**Disparate Parity**
Entire Model	0.037	1.0408
Model with deweighted proxies	0.027	1.0301

Here, we can see that the statistical parity reduced in our model that used de-weighted proxies as opposed to the model with all the variables. Our statistical parity of the entire model was 0.037, while the statistical parity was 0.027 for the model with de-weighted proxy "Present Employment Status." Thus, we can see that we have actually reduced the positive discrimination (i.e. against men) in our protected groups. Similarly, our disparate parity ratio is closer to 1 as well – indicating that a reduction in positive discrimination has occurred. Overall, our results were extremely surprising, as we were able to increase the fairness in our model using the KNN algorithm while actually observing an increase in predictive accuracy.

In terms of comparison with the student paper, the thesis paper saw similar results. As the suppression pre-processing method led to an average increase of 0.005 compared to the baseline model. Overall, our results seem to match those with the student paper as both the K-NN model and the logistic regression model can help increase fairness without having a significant loss in predictive accuracy.

6 Discussion

The use of fair machine learning algorithms are becoming increasingly critical in day to day applications. As such, it is important to account for fairness through these algorithms in efforts to reduce bias against certain groups. In this report, we deweight our proxy variables in efforts to reduce the inference of the sensitive attribute and ultimately reduce the bias. Surrounding the traditional notion of fair-utility trade-off, we did not observe any significant loss in predictive power when attempting to adjust for fairness. In fact, our findings were very surprising – we saw an increase in both utility while being able to reduce bias against the protected groups. Research surrounding the de-weighting of proxy variables was a new avenue we pursued in this paper, in hopes of comparing the results to different fairness processing methods as conducted in Kasim's paper. With this, our results appeared to confirm the student paper's findings in terms of increasing predictive accuracy and fairness using the suppression pre-processing methods (we also account for proxies). With this, Kasim's paper employed many additional techniques to achieve fairness that were not discussed in this paper. For future discussion, we can employ a similar strategy to the Credit Card Defaults dataset (as done in the student research paper) from the UCI machine learning repository. Kasim's student paper also uses results from this dataset. It would be interesting to see if the trade-off in fairness vs utility is more prevalent through those cases, and to see if we can achieve similar results in a different dataset.

This report was completed individually, thus a member contribution section is not necessary.

7 References

1. Machine Learning Fairness in Finance: An Application to Credit Scoring, arno.uvt.nl/show.cgi?fid=157552. Accessed 13 June 2023.
2. "What Is Machine Learning Fairness? What You Need to Know." Coursera, www.coursera.org/articles/machine-learning-fairness. Accessed 13 June 2023.
3. Bücker, M., Szepannek, G., Gosiewska, A., and Biecek, P. (2021). Transparency, Auditability and eXplainability of Machine Learning Models in Credit Scoring. J. Oper. Res. Soc. in print. doi:10.1080/01605682.2021.1922098
4. Crook, J. N., Edelman, D. B., and Thomas, L. C. (2007). Recent Developments in Consumer Credit Risk Assessment. Eur. J. Oper. Res. 183, 1447–1465. doi:10.1016/j.ejor.2006.09.100
5. "What Is Explainable AI (XAI)?" IBM, www.ibm.com/watson/explainable-ai. Accessed 13 June 2023.

6. Stephanie. “Kendall’s Tau (Kendall Rank Correlation Coefficient).” Statistics How To, 14 Nov. 2017, www.statisticshowto.com/kendalls-tau/.

7. “Protected Attributes and ‘Fairness through Unawareness’: Exploring Fairness in Machine Learning for International Development: Supplemental Resources.” MIT OpenCourseWare,ocw.mit.edu/courses/res-ec-001-exploring-fairness-in-machine-learning-for-international-development-spring-2020/pages/module-three-framework/protected-attributes/. Accessed 13 June 2023.

8. Description of the German Credit Dataset - Axxio, www.axxio.io/wp-content/uploads/2018/05/documentation.pdf. Accessed 13 June 2023.