Fairness in AI

Tables of Content

Executive Summary

The extensive use of artificial intelligence and machine learning in almost every aspect of decision making has not only improved the efficiency and accuracy but has also led to unintended risks. One such risk is that the machine learning algorithms disproportionately hurt or benefit specific groups of people. Since such discrimination can destroy the reputation of a firm and can also lead to a potential lawsuit, therefore extensive research is being conducted to make the algorithms fair. The report aims at highlighting how algorithms can have bias, what are the source of these bias, and how they negatively impact any organization. Further, many techniques have also been explored to mitigate discrimination. These techniques can be broadly divided into three; mitigation during pre-processing, in-processing, and post-processing. Under the pre-processing techniques, the dataset is often transformed before feeding it into an algorithm. On the other hand, in-processing techniques try to optimize the algorithm subject to fairness, while post-processing techniques usually include changing the labels of the output to enhance fairness.

Based on this emerging area, the report has attempted to try reducing the gender discrimination in the Loan Dataset by applying five different approaches. Positive Predictive Value (PPV) was used to measure gender bias, where a positive PPV indicates bias against women. First and foremost, simple logistic regression was used to predict whether an applicant will receive a loan or not. After calculating the PPV value, it was observed that the rejection rate among women was higher than men and lower percentage of women received loans. In an attempt to remove bias, gender was dropped from the model and another logistic regression was fitted. However, by merely removing the sensitive variable, the PPV value did not improve because often the sensitive variables are correlated to other variables.

Based on this assumption, another model was created, but this time the model predicted gender. From the results of the logistic regression and random forest, four variables were selected which were important for predicting gender. Next, a regularization model was coded in Python, where rather than assigning the same lambda for each variable, custom lambdas could be given to each variable. By assigning

different values to the variables important for predicting gender, the model was able to reduce the PPV value but at the cost of reducing accuracy. Usual L2 regularization was also applied which worked but was unable to reduce the PPV as much as the custom lambda regularization.

Next, two approaches were adopted from Zafar et al. (2017). Under the first approach, the author tried to impose a demographic party, which means that the percentage of women for whom the prediction was to give loan should be equal to the percentage of men for whom the prediction was to give loan. To implement this, covariance between the sensitive variable and the distance between the vector of independent variables and decision boundary was added as a constraint to the loss minimization function of the logistic regression. By applying this, the model experienced an increase in the demographic parity, but the PPV value increased. However, demographic parity only considered who to give loan but did not consider whether the loan will be repaid. To account for this, equal opportunity was used which also considers the true labels of the object, i.e., whether the person received the loan or not in reality. The objective and constraints remained like the previous approach. However, this approach did not improve anything in the loan dataset but is considered to be an effective approach in many studies.

After exploring different approaches, it was found out that custom regularization, demographic parity, and equal opportunity approaches can be used to make the machine learning models fair. However, it should be noted that fairness does not come without any cost because as discrimination reduced the accuracy of the models and the profitability of the bank reduces too. Therefore, while choosing a technique to reduce bias, one must carefully select the parameters such that the model becomes fair without losing a lot of accuracy and profit.

Overview

With the advancement of technology, the private and public sectors are increasingly turning to machine learning and artificial intelligence for making day-to-day decisions. The algorithms used are often considered to be more efficient and accurate compared to humans. However, the adoption of these algorithms brings in several unintended risks. One such risk is concerned with making 'unfair' decisions, where the outcomes can disproportionately hurt or benefit particular groups of people belonging to a specific group. Needless to say, blindness to such discrimination not only brings in poor reputation for an organization but can also hurt them legally. Thus, it is highly essential to treat sensitive features carefully in any machine learning algorithm.

In this report, the concept of 'unfair' machine learning and artificial models has been explored along with highlighting the potential sources and problems associated with unfair models. The report lays out work conducted in this field, and finally provides several methods to deal with such bias and obtain a fair machine learning model.

Unfair Algorithms

What is Bias

In 2019, Apple received immense negative publicity as the Apple Credit Card was claimed to be "sexist" against women applying for credit. Shortly after the card was launched, it was observed that women were getting lower lines of credit or even rejected for Apple's credit card, while their husbands with equal financial status were readily accepted. Some women also claimed that despite having a better credit score and other factors in their favor, they were rejected for the credit cards (Gopinath, D., 2021). Since credit acceptance and rejection is widely done by algorithms, therefore, Apple's algorithms were criticized to be 'unfair'. Similarly, in 2016, the risk assessment AI software named COMPAS was found to have severe racial bias, where blackness ridiculously increased the predicted risk of the criminal (Irolla, P., 2020). Such

algorithms which have the *"presence of any prejudice or favoritism towards an individual or group based on their inherent or acquired characteristics while making decisions"* are often termed as unfair algorithms (Mehrabi, N., 2022). The decisions of unfair models are usually skewed towards a group of people belonging to a particular gender, race, religion, and so on, irrespective of these groups being similarly-situated in terms of other factors.

According to Barocas and Selbst (2016), unfairness in an algorithm exists in two forms namely *disparate treatment* and *disparate impact.* A decision-making process suffers from disparate treatment when the decisions are partly based on the subject's sensitive attribute information. In other words, disparate treatment occurs when sensitive attributes like gender or race, are used in the algorithm. On the other hand, the process has a disparate impact if the outcomes disproportionately hurt or benefit people with certain sensitive attribute values (Zafar, M.B., 2017). For instance, in the case of Apple Cards, women were hurt more as they were not readily given the credit cards.

Sources of Bias

One primary reason why such an impact exists in the machine learning algorithms is the presence of human bias in the historical dataset. The existing human bias against certain groups can reflect in the way the data is prepared and thus often reproduced in the algorithms (Lee, N.T., 2019). For a long time, women have been associated with having poor credits and having lower financial stability compared to their male counterparts, while blackness is primarily targeted for any criminal activities. Because of this, the dataset usually highlights that woman should be given less credit and blackness is high risk, which was further reflected in the algorithms of Apple and COMPAS. Furthermore, insufficient training data is another cause of algorithmic bias. If the training data represents one segment more than the other, then the predictions are bound to have worsened outcomes for the under-represented groups. The algorithms are also expected to be skewed towards the group that is over-represented (Lee, N.T., 2019).

Impact of Bias

Since algorithms are now harnessing volumes of data to make important decisions which affect humans in a wide range of tasks. Therefore, understanding the negative impact of such bias is essential to support the efforts of mitigating the bias. First and foremost, in almost every country, right for equality are present, which states that any discrimination based on “race, national or ethnic origin, religion, gender, age, or mental or physical disability” is punishable offence (Section 15, Government of Canada). This means that unfair algorithms can lead to potential lawsuits against an organization, along with heavy penalties. Furthermore, ignoring any discriminatory activity an organization can ruin their reputation, which can hamper the revenue of the company. Lastly, unfair algorithms can restrict the organization from finding potential clients, who fit all the criteria but are rejected due to their sensitive attributes. For instance, Apple was unable to capture the women who had the ability to pay back credit on time, just because their model had gender discrimination. Owing to these drawbacks, it has become essential today to find ways for mitigating the unfairness in the machine learning and artificial intelligence models.

Techniques for Mitigating Bias

Despite being a new research area, extensive research has been conducted in a short span of time. Generally, the bias can be mitigated during pre-processing, in-processing, and post-processing. While mitigating the bias in the pre-processing field, the training dataset is usually transformed so that the underlying discrimination can be removed. On the other hand, in-process techniques try to remove discrimination during the training of the model. This is normally done by changing the objective function or imposing a constraint such that fairness is imposed. Lastly, under the post-processing techniques, the bias is treated after training the model where transformations are applied to the output of the model to improve the prediction fairness (Caton, S., 2020). Literature under each category has been explored in the subsequent subsections.

Pre-Process Mechanisms

The techniques used during pre-processing usually include sampling, transformation, relabeling, and reweighing. Kamiran and Calders (2012) and Luong et al. (2011) gave the first few approaches to pre-processing the training data. According to the authors, changing the labels of some observations or reweighing them before training can make the classification process fairer. In the first technique, the labels of some objects are changed from class 0 to class 1, such that discrimination is reduced while maintain the overall class distribution. The samples which are closer to the decision boundaries are typically the ones whose labels are changed as they are most likely to be discriminated against.

In the second approach, the weights are assigned to each object such that an object with protected attribute and positive class are given higher weights while an object with protected attribute and negative class is given a lower weight (Kamiran, F., 2012). The more recent mechanism suggests modifying the features in the dataset to obtain equal distribution for both protected and unprotected groups, so that it is difficult for the model to differentiate between the two groups (Pessach, D., 2022). Kelley et al. (2021) also suggested the use of down-sampling, i.e. to remove observations randomly from the majority class until counts are equal with the majority group, and up-sampling, i.e. adding more observations for minority class, to reduce bias.

In-Process Mechanisms

Modifying the machine learning models itself to account for fairness comes under in-process mechanisms. Kamishima et al. (2012) gave the initial approaches for fair machine learning models where the objective function was modified. In their approach, the authors introduced the L2 regularization along with another regularization term which penalized the mutual information between the sensitive attribute and the classifier predictions. The regularizer is designed in such a way that it takes a higher value when a class is determined mainly based on sensitive features so that it can become less influential in final output (Kamishima, T., 2012). Bechavod and Ligett (2017) also introduced a regularization term which is not only hypothesis-dependent but also data-dependent. In this method, two penalizers were defined which captures the

difference between false positive rate and false negative rate between the protected and unprotected population. Through these terms the authors wanted to achieve an equalized odds criterion, which suggests that a classifier should predict any particular label equally well for all values of an attribute.

Similar to changing the objective function, another approach is to add constraints to the cost minimization function. Zafar et al. (2017) constrained the covariance between the sensitive variable and the distance between the vector of independent variables and decision boundary, such that it is lower than a given covariance threshold. According to the author, if the covariance is zero then the sensitive attribute is not affecting the decision, and thus the model will be free from any discrimination. However, it was also highlighted that there is a trade-off between the accuracy of the model and fairness, therefore the covariance was kept a little above zero to maintain accuracy of the model. This approach has been explained in depth in the later sections. Furthermore, some authors have also tried applying a multi-objective loss function based on logistic regression (Zemel, B., 2013).

Post-Process Mechanisms

Under the post-processing mechanisms, changes are made on the output scores of the classifier to make fair decisions. Hardt et al. (2016) proposed techniques for simply changing some decisions of the classifier to enhance equalized opportunity. Similarly, some techniques also try to select different threshold values for the two groups of the sensitive attribute (Menon A.K., 2017). Menon and William (2017) highlighted that for a classification model which has a tradeoff between accuracy and fairness, the optimal fairness-aware classifier is one which has an instance-dependent threshold of the class-probability function. Corbett-Davies et al. (2017) also used a similar approach, but the authors selected separate thresholds for each group that maximizes accuracy while minimizing discrimination.

Numerous approaches during pre-process, in-process, and post-process are available for mitigating the discrimination and move towards a fair learning. While some approaches can be applied easily, some are complex in nature. However, one must weigh the advantages and disadvantages of the respective approaches to get desirable results.

Objective

Considering the importance of fairness, the report aims to address the issue of discrimination in training of the machine learning models. A Loan eligibility status dataset was used to calculate the gender discrimination while predicting the loan eligibility status of a customer given the independent variables. The gender discrimination was calculated using defined metrics and the objective was to reduce the gender discrimination while using machine learning models to predict the loan eligibility status. First, a Logistic Regression model was used to predict the loan eligibility status, and the gender discrimination was measured on this model. Various approaches were then used in order to reduce the gender discrimination.

Dataset

Source of Dataset

The dataset used in this report was sourced from Kaggle that provided the loan eligibility status based on customer details provided while filling the online application form. These details were Gender, Marital Status, Education, Number of Dependents, Income, Loan Amount, Credit History, and others. The goal was to use a machine learning algorithm to predict the loan eligibility status given the set of independent features or details about the customer. It is critical to note that the sensitive attribute here was Gender and the objective of this project was to measure the gender bias that occurred while using machine learning algorithms and subsequently reduce this bias using various approaches.

Visualizing Gender Bias

A count plot was plotted on the Loan dataset to visualize the distribution of males and females and it was found that the proportion of males in the dataset was significantly higher than the number of females in the dataset suggesting that the females might be under-represented in the data (Figure 1).

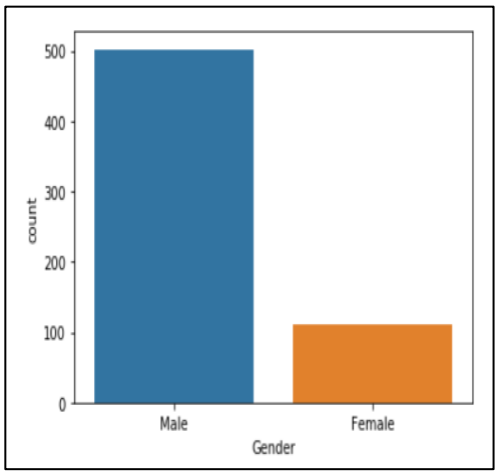

Figure 1: Count of Men and Women in Loan Dataset

Additionally, bar charts were plotted to visualize the gender wise Loan Acceptance and Loan Rejection rates. It was observed that the loan acceptance rate among men was higher compared to loan acceptance rate for women whereas the loan rejection rate for women was higher compared to loan rejection rate for men (Figure 2).

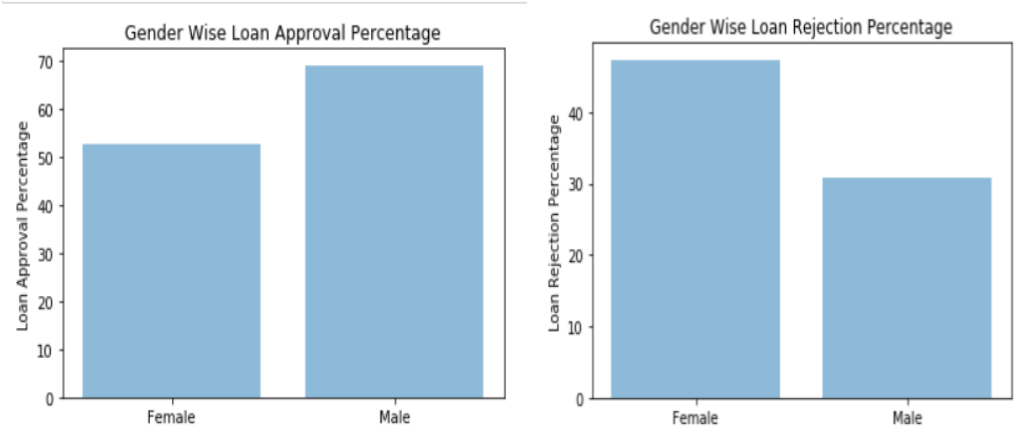

Figure 2: Loan Acceptance and Rejection Rate

Pre-processing the Dataset

The dataset was first evaluated to check for any missing values. The missing values were then imputed with the mode for the categorical variables and using median for the numeric variables. A log transformation of

loan amount was also added. The Loan Id column was removed as it didn't have any value for the machine learning models. Next, the dependent variable, Loan Status, was converted into a binary variable, where 0 signified N (loan eligibility status – Not eligible) and 1 signified Y (loan eligibility status – Yes). The other categorical variables were converted into dummy variables as well. Lastly, the dataset was split into training and testing dataset for the machine learning models. The machine learning models are trained on the training dataset and evaluated for performance on the testing dataset.

Evaluating Machine Learning Models

Confusion matrix are used normally to evaluate the classifier models which shows how well a classification model (or "classifier") performs on a set of test data for which the true values are known.

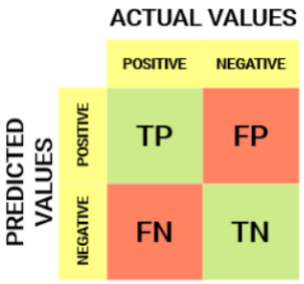

Figure 3: Confusion Matrix

TP = True Positive, i.e., correctly predicted loan eligibility status as Yes

FP = False Positive, i.e., incorrectly predicted loan eligibility status as Yes whereas actual status was Not eligible

FN = False Negative, i.e., incorrectly predicted loan eligibility status as No whereas actual status was eligible (Yes).

TN = False Negative, i.e., correctly predicted loan eligibility status as No

Evaluation Metrics

Accuracy – Accuracy is the most common method of evaluating machine learning models. Accuracy in a classification algorithm is calculated as the number of all correct predictions divided by the total number of the dataset.

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN}$$

Sensitivity - Sensitivity is calculated as the number of correct positive predictions divided by the total number of positives.

$$\text{Sensitivity} = \frac{TP}{TP + FN}$$

Specificity - Specificity (SP) is calculated as the number of correct negative predictions divided by the total number of negatives.

$$\text{Specificity} = \frac{TN}{TN + FP}$$

Profit - The corporation earns money for each consumer they lend to (a true positive prediction), but they lose money when they lend to someone who would not be able to repay the loan (a False Positive prediction). It is assumed that a consumer who is not eligible for a loan does not harm the company (a true negative prediction), and for simplicity, they suffer no opportunity cost for rejecting a customer who would have been eligible (a false negative prediction).

If a loan amount equal to 1000$ was granted to each customer, for the time duration of 4 years, at an interest of 6 % compounded annually, the company earns a revenue equal to ~1262 $ from each customer and if a person is not able to pay back the loan, the firm loses 1000$. As a result, the profit of the bank can be calculated using the below formula:

$$\text{Profit} = 1262 * TP - 1000 * FP$$

Measuring Gender Bias

In order to evaluate whether the machine learning model incurred any gender bias, a metric, Positive predictive value (Chouldechova 2017) was used. The metric illustrates the difference in the algorithm's ability to predict loan eligibility status correctly, conditional on actual loan eligibility status between the two genders.

Positive Predictive Value (PPV) represents the quality of a positive prediction made by the model, and in general is calculated using the below formula:

$$PPV = \frac{TP}{TP + FP}$$

For a classification threshold, τ, gender discrimination is measured using $PPV(\tau)$ which can be calculated using the positive predicted value for males minus the positive predicted value for females.

$$PPV\ (\tau) = \frac{TP_M(\tau)}{TP_M(\tau) + FP_M(\tau)} - \frac{TP_F(\tau)}{TP_F(\tau) + FP_F(\tau)}$$

where:

TPM = True Positives for Males

FPM = False Positives for Males

TPF = True Positives for Females

FPF = False Positives for Females

Interpretation of PPV:

A $PPV(\tau)$ greater than zero signifies bias against females, whereas a value less than zero signifies bias against male. On the other hand, $PPV(\tau)$ value equal to zero represents an ideal situation where there is no gender discrimination. Throughout the approach PPV is calculated at 0.5 cutoff and is termed as PPV instead of $PPV(\tau)$.

Unfair Logistic Regression

A logistic regression model was used to predict the loan eligibility status using all the independent variables. The model was trained using the training data and evaluated on the testing data. A train test split of 70:30 was used where 70% of training data was used for training and 30 % of data was used for testing the model. The evaluation metrics were calculated on the testing data, and gender discrimination was calculated using PPV. It was observed that the PPV value was 19.28 % which denoted that there was 19.28 % discrimination or bias against women (Table 1).

Metric	Value
Accuracy	80%
Sensitivity	98.34%
Specificity	45.31%
Profit	$115178
PPV	19.28 %

Table 1: Metrics for Logistic Regression

Approaches for Fair Models

After conducting thorough literature review and the machine learning concepts, five approaches were applied on the Loan dataset to check whether the PPV value can be reduced or not. The in-depth explanation of the approaches is done in the subsequent sub sections.

Unawareness

In the simplest form, bias might be reduced by deleting the sensitive attribute which is termed as 'unawareness'. Usually, such approach is considered to be ineffective because generally the sensitive attribute is correlated to the other variables in the dataset. Additionally, removing sensitive attribute only removes the disparate treatment but does not account for disparate impact. While predicting the loan status in Loan dataset by removing gender, the accuracy of the model remained the same, but the PPV value increased from 19.28% to 22.4%. This suggests that removing the sensitive attribute can lead to higher discrimination. This has also been illustrated by Kelley et al. (2021) in their research. Furthermore, sensitivity was 98.34%, specificity was 45.31%, and the profit was calculated to be $115178 (Table 2).

Metric	Value
Accuracy	80%
Sensitivity	98.34%
Specificity	45.31%
Profit	$ 115178
PPV	22.4%

Table 2: Metrics for Logistic Unawareness

Regularization

Another approach used to reduce bias was L2 norm regularization. Regularization is accomplished by adding a penalty term to the cost function such that the coefficients are reduced. This helps in reducing the variance in the model and hence allows for reducing the testing errors. In logistic regression, the cost function is modified by inserting a penalty term (shrinkage term) that doubles the lambda with the squared weight of each unique feature. As a result, the following is the cost function:

$$J(w) = \frac{1}{m}\sum_{i=1}^{m} Cost\left(h\left(x^{(i)}\right), y^{(i)}\right) + \frac{\lambda}{2m}\sum_{j=1}^{n} w_j^2$$

The regularization parameter is called λ. It manages the trade-off between two objectives: a good fit to the training data and keeping the parameters modest to avoid overfitting. Since the objective of regression is to minimize the loss function, therefore the gradient of the modified loss is computed to get the values of coefficients of variables:

$$\frac{\partial}{\partial w_i} J(w) = \frac{1}{m}\left[\sum_{j=1}^{m}\left(h\left(x^{(j)}\right) - y^{(j)}\right)x_i^{(j)} + \lambda w_i\right]$$

Here, large w_i will be significantly penalized due to the presence of the regularization parameter, and vice versa. This way the regularization parameter regulates the growth of the coefficients and does not allow their value to be very large. This way, the ***h(x)*** obtained with these controlled parameters will be more generalizable. This method of regularization was used on the Loan dataset while predicting the loan status. Different values of lambdas were used from which λ = 50 gives the lowest value of PPV. The value of accuracy, PPV, sensitivity, specificity, and profit were also calculated for different λ values which are presented in Table 3.

λ value	Accuracy	PPV	Sensitivity	Specificity	Profit
1	0.794595	0.210618	0.975207	0.453125	113916
5	0.794595	0.209091	0.966942	0.453125	112654
10	0.794595	0.209091	0.966942	0.453125	112654
20	0.794595	0.209091	0.966942	0.453125	112654
50	0.794595	0.192831	0.983471	0.453125	115178
100	0.794595	0.209091	0.966942	0.453125	112654
200	0.794595	0.210618	0.975207	0.453125	113916

Table 3: Metrics for different Lambda in Regularization

Custom Lambda

In the next method, the regularization method was transformed in such a way that despite giving same λ value to each variable, a python code was written to assign custom values to each variable (Appendix A). Before assigning custom values, a logistic regression was fitted which predicted gender rather than loan. This was done primarily to identify the variables which were important to predict gender. Since these important variables were enhancing the effect of gender while predicting loan status, therefore by giving them higher weights it was expected that the effect of gender can reduce and thus the model will become more gender neutral. Based on the p-values of the logistic regression (Appendix B) and importance value of random forest (Appendix C), "Applicant Income", "Married: Yes", "Property_Area_Semiurban" and "Co-applicant Income" were identified to be important for predicting gender.

To apply the custom lambda approach, the cost function of the logistic regression of modified to have unique values of λ. The equation is as follows:

$$J(w) = \frac{1}{m}\sum_{i=1}^{m} Cost\left(h\left(x^{(i)}\right), y^{(i)}\right) + \frac{1}{2m}\sum_{j=1}^{n} \lambda_j (w_j^2)$$

Similarly, the gradient of the cost function for custom lambda is computed to be:

$$\frac{\partial}{\partial w_i} J(w) = \frac{1}{m}\left[\sum_{j=1}^{m} \left(h\left(x^{(j)}\right) - y^{(j)}\right) x_i^{(j)} + \lambda_i w_i\right]$$

This algorithm was then used to give a higher regularization penalty to the important variables. In the first phase all the important variables were given the same λ value which was higher than the λ for the other variables. By doing this, the lowest PPV value was achieved by taking λ for important variables as 150 and λ for others as 50. However, it was observed that by reducing the PPV value, the accuracy, specificity, and the profit of the model dropped significantly. The accuracy, PPV value, sensitivity, specificity, and profit calculated for several combinations of λ are shown in Table 4 below.

λ for other variables	λ for important variables	Accuracy	PPV	Sensitivity	Specificity	Profit
1	100	0.794594	0.210617	0.975206	0.453125	113916
10	100	0.778378	0.200248	0.983471	0.390625	111178
20	100	0.778378	0.200248	0.983471	0.390625	111178
50	100	0.681081	0.186179	0.983471	0.109375	93178
50	150	0.691892	0.170998	0.983471	0.140625	95178
50	200	0.670270	0.199999	0.983471	0.078125	91178

Table 4: Metrics for custom Lambda Regularization

In the next phase, the λ value of important variables were assigned based on their importance value from the random forest. According to the importance value, "Applicant Income" was given the highest value, followed by "Married: Yes", "Co-applicant Income", and "Property_Area_Semiurban". All other variables were given the same value of λ. The accuracy, PPV value, sensitivity, specificity, and profit were calculated using a variety of custom lambda values, as shown in Table 5 below.

λ Applicant Income	λ Co-applicant Income	λ Married	λ Property	λ Others	Accuracy	Sensitivity	Specificity	PPV	Profit
150	100	120	75	50	68%	98%	11%	18.6%	$93178
120	75	100	55	50	68%	98%	12.5%	19.1%	$94178
120	75	100	50	20	77.8%	98.3%	39%	20%	$111175
200	120	150	100	50	68%	98%	11%	18.6%	$93178

Table 5: Metrics for custom Lambda Regularization based on their importance

Demographic Parity

The next approach has been taken from Zafar et. al (2017), where the authors have tried to reduce disparate treatment and impact and enhance the demographic parity. Demographic parity is a fairness metric which indicates that the classification should be independent of the sensitive attribute. This means that the positive rate among both the groups of the sensitive attribute should be the same. In terms of Loan dataset, the percentage of men with label '1', i.e. loan should be given, and the percentage of women with label '1' should be equal. Mathematically, it can be written as:

$$P\ (\hat{y} = 1 \mid z = 0) = P\ (\hat{y} = 1 \mid z = 1)$$

where y is the predicted labels and z is the sensitive attribute, gender (z = 1 means male).

Based on demographic parity, p% is defined as the ratio of the positive rate of one group of the sensitive attribute to the positive rate of the other group. Usually, the group who faces discrimination is the one which is taken in the denominator, therefore p% in loan dataset can be calculated by taking the ratio of positive rate of females to positive rate of males.

$$\text{p\% score} = \frac{P(\hat{y} = 1 \mid z = 0)}{P(\hat{y} = 1 \mid z = 1)}$$

In the paper (Zafar, M.B., 2017), the author separated the sensitive attribute from the dataset to create two disconnected sets where set *Z* included the sensitive variables while set *X* included all the other variables. By removing the sensitive feature from the dataset, the author tried to remove the disparate treatment. On the other hand, p% is used to reduce the discrimination against women and make the algorithm free from disparate impact. In general, 80% p-rule is suggested by the U.S. Equal Employment Opportunity Commission. Therefore, it is suggested that a constraint should be applied to the cost minimization function. However, p% is a non-convex in nature which is very difficult to implement in an optimization problem.

To avoid such a problem, the covariance between the sensitive variable and the distance between the decision boundary and the set of other variables *X*, was used as a proxy variable. The decision boundary

is typically based on the parameters θ that is achieved by minimizing the loss function. Then, given the set of feature vector *X*, the classifier predicts the label 1 or 0 based on the signed distance from the feature vector *X* to the decision boundary. Based on this signed distance, the covariance is calculated as follows:

$$Cov\left(z, d_\theta(x)\right) = \mathbb{E}[(z-\bar{z})d_\theta(x)] - \mathbb{E}\left[(z-\bar{z})\overline{d_\theta}(x)\right]$$
$$\approx \frac{1}{N}\sum_{i=1}^{N}(z_i-\bar{z})d_\theta(x_i)$$

where Z_i is the sensitive attribute and $d_\theta x_i$ is the signed distance from the user's feature vectors to the decision boundary.

After computing the covariance, it is used to constraint the loss function such that the covariance is between a given threshold. By keeping the covariance low, the p% will be higher and discrimination will be lower. This constraint is applied as follows where log *p* is the *c* is the covariance threshold:

$$\text{minimize} \quad -\sum_{i=1}^{N}\log p\ (y_i|x_i,\theta)$$
$$\text{subject to} \quad \frac{1}{N}\sum_{i=1}^{N}(z_i-\bar{z})d_\theta(x_i) \leq c$$
$$-\frac{1}{N}\sum_{i=1}^{N}(z_i-\bar{z})d_\theta(x_i) \geq -c$$

This constraint can be easily applied to the Loan dataset by using the inbuilt library **Scikit Lego** where the function **DemographicParityClassifier** can take different values of covariance threshold. The mentioned library was used on the loan dataset to check how different values of covariance threshold could change the p% score and the accuracy of the model. Figure XXX illustrates that by applying the constraint on covariance, the p% of the model increased from 85% to 92%. However, there is a trade-off between accuracy and fairness, as the accuracy of the unfair logistic regression was higher than the fair classifier. The PPV value was also calculated for the fair-classifier, however it was observed that the value increased to 21% compared to the initial unfair model with 19% PPV.

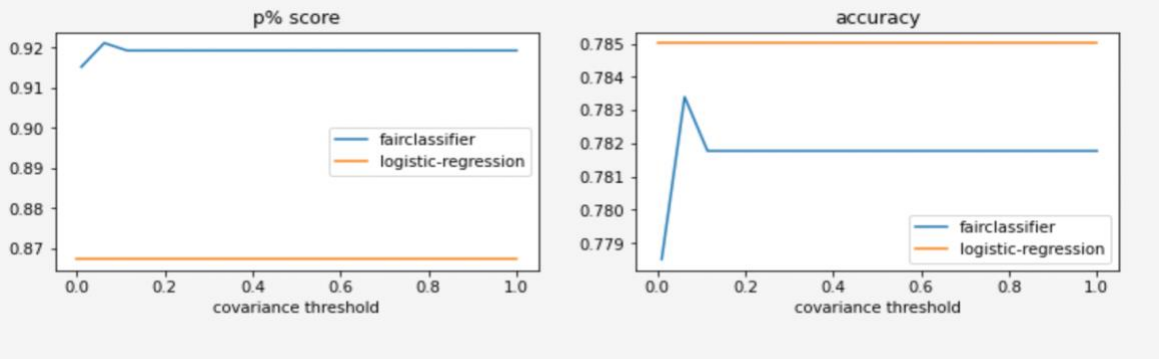

Figure 4: p% score and Accuracy for different vales of covariance threshold

The similar approach was used by the author on the Adult Dataset which also showed an improvement in the p% score but at the cost of accuracy.

Equal Opportunity

Although demographic parity helps to reduce the discrimination by making the percentage of classification of positive class equal among the two groups equal. However, the measure only looks at the loans made to the groups but fails to incorporate the rate at which the loans are to be repaid. To account for this, Equal Opportunity is used instead. As described earlier, equal opportunity also looks at the true labels of the objects, due to which the p% changes as follows:

$$\text{equality of opportunity} = \frac{P(\hat{y} = 1 \mid z = 0, y = 1)}{P(\hat{y} = 1 \mid z = 1, y = 1)}$$

where *y* is the true label of that object.

Similar to the previous approach, the covariance is constrained such that it lies within a given threshold. The only difference is that rather than taking the entire population, only the subset of population where the true label *y*, is equal to '1', which means loan is given to the individual. The updated optimization model is as follows:

$$\text{minimize} \quad -\sum_{i=1}^{N} \log p\ (y_i|x_i,\theta)$$

$$\text{subject to} \quad \frac{1}{POS}\sum_{i=1}^{POS}(z_i-\bar{z})d_\theta(x_i)\ \leq\ c$$

$$-\frac{1}{POS}\sum_{i=1}^{POS}(z_i-\bar{z})d_\theta(x_i)\ \geq\ -c$$

The **Scikit Lego** library has another function **EqualOpportunityClassier** which can be used to implement the above constraint. The function was used on the Loan dataset, however no change was observed compared to the unfair logistics regression model. This might be because the dataset in general had a high p% score due to which the fair-classier failed to reduce discrimination. The model also showed no change in the PPV value. However, the author implemented this approach on the Adult Dataset, which showed quite good results by employing a fair classifier.

Findings

A logistic regression model was used to predict the loan eligibility status and it was found that the model discriminated against women with a PPV value of 19.4 %. To mitigate gender bias, five approaches were used. The first approach involved removing the sensitive attribute, i.e., gender while using the Logistic Regression model to predict the loan eligibility status. However, it was found that on removing gender from the model, there was an increase in PPV value from 19.28 % to 22.4 % suggesting that merely removing sensitive attribute from the machine learning model did not help in reducing discrimination. The next approach used Regularization where different regularization parameter (λ) values were used and PPV was measured for these different values, and it was found that there was no improvement in the PPV value. Next, a custom logistic regression model was developed that allowed customizing the regularization parameter for individual features. Logistic regression model and Random Forest model were used to predict the gender to get the important variables predicting the gender. The variables that were significant were:

Applicant Income, Co-applicant Income, Property: Semiurban, and Married: Yes. These variables were assigned a higher regularization parameter value than the other variables (penalized more), and it was found that it helped reduce the PPV value. The lowest PPV value, 17.09 % was observed using 150 as λ value for the significant variables impacting gender and a λ value of 50 for the remaining variables. The other approaches that were used were: Demographic Parity and Equal Opportunity. Although Demographic Parity worked well in increasing p % score, there was no improvement in PPV value whereas Equal Opportunity did not work well on this dataset to improve the p % score or PPV.

The other important point to be noted here was that any method that helped in reducing gender discrimination has an associated cost in terms of reducing the accuracy of the model and the firm profitability. The custom regularization, for instance, that reduced the gender discrimination PPV value to from 19.4 % to 17.09 %, reduced model's accuracy from 80 % to 69 %, and profit from $115178 to $95178. As a result, when selecting a technique to eliminate bias, the parameters must be carefully chosen such that the model becomes fair without sacrificing a lot of accuracy and firm's profitability.

Conclusion and Recommendations

With the emerging importance of making the machine learning algorithms fair, several approaches were tried to examine which approach was effective in reducing the discrimination. A total of five approaches were used from which custom regularization worked best for the Loan dataset. Furthermore, demographic parity and equal opportunities are also used commonly to reduce discrimination but these approaches failed to show any results in the Loan dataset. On the final notes, all the approaches showed that there is always a trade off between reducing discrimination and accuracy and profit. Whenever the discrimination index reduced there was a reduction in the accuracy and profit. Therefore, it is recommended that the firms who wish to make their models fair must choose their parameters in such a way that their accuracy and profit is not reduced greatly. Similarly, from the approaches applied to the Loan dataset, it is recommended that if

the loss of reputation or loss of resources due to lawsuit is higher than the loss in profit due to the fair machine learning, then the firm should adopt the fair-machine learning.

Way Forward

The custom regularization provided the best results to reduce the gender bias. Currently, fixed values of λ were used to check the model's performance and the gender bias metrics used were PPV and p % score. The hyperparameter, λ can be optimized using automatic hyperparameter tuning methods. Additionally, constraints can be added on the custom λ values to penalize the variables according to the variable importance given by Random Forest algorithm to predict the gender. Constraints can also be added so that the accuracy or profitability should not be reduced beyond a certain level while minimizing the gender bias. The techniques other than custom regularization could also be explored to minimize the gender bias and other fairness metrics such as predictive rate parity, individual fairness, counterfactual fairness can be used to evaluate which techniques are able to reduce which fairness metric.

References

Barocas, S., Selbst, A.D., 2016, "Big Data's Disparate Impact", *104 California Law Review 671.*

Bechavod, Y., Ligett, K., 2017, "Learning fair classifiers: A regularization-inspired approach", *arXiv:1707.00044.*

Caton, S., Hass, C., 2020, "Fairness in Machine Learning: A Survey", *arXiv: 2010.04053.*

Chouldechova, A., 2017, "Fair prediction with disparate impact: A study of bias in recidivism prediction Instruments", *Big Data* 5(2):153–163.

Corbett-Davies, S., Pierson, E., Feller, A., Goel, S., Huq, A., 2017, "Algorithmic decision making and the cost of fairness", *In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, New York, NY, 797–806.

Gopinath, D., September 2021, "What does it mean to be fair? Measuring and understanding fairness", retrieved from https://towardsdatascience.com/what-does-it-mean-to-be-fair-measuring-and-understanding-fairness-4ab873245c4c

Hardt, M., Price, E., Srebro, N., 2016, "Equality of opportunity in supervised learning", *Advances in Neural Information Processing Systems*, 3315 – 3323.

Irolla, P., April 2020, "Unfair biases in Machine Learning: what, why, where and how to obliterate them" retrieved from https://medium.com/disaitek/unfair-biases-in-machine-learning-what-why-where-and-how-to-obliterate-them-1d6e682ac556

Kamiran, F., Calders, T., 2012, "Data preprocessing techniques for classification without discrimination", *Knowledge and Information Systems*, Issue 33, 1 – 33.

Kamishima, T., Akaho, S., Asoh, H., Sakuma, J., 2012, "Fairness-aware classifier with prejudice remover regularizer", *In proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases,* 35–50.

Kelley, S., Ovchinnikov, A., 2020, "Anti-discrimination Law, AI, and Gender Bias in Non-mortgage Fintech Lending".

Lee, N.T., Resnick, P., Barton, G., May 2019, "Algorithmic bias detection and mitigation: Best practices and policies to reduce consumer harms", retrieved from https://www.brookings.edu/research/algorithmic-bias-detection-and-mitigation-best-practices-and-policies-to-reduce-consumer-harms/

Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., Galstyan, A., 2022, "A Survey on Bias and Fairness in Machine Learning", *ACM Computing Survey,* Vol 6, Issue 54, 1 – 35.

Menon, A.K., Williamson, R.C., 2018, "The cost of fairness in binary classification", *In Proceedings of the Conference on Fairness, Accountability, and Transparency*, 107–118.

Pessach, D., Shmueli, E., 2022, "A Review on Fairness in Machine Learning", *ACM Computing Survey*, Vol. 3, Issue 55.

Zafar, M.B., Valera, I., Rodriguez, M. G., Gummadi, K.P., 2017, "Fairness Constraints: Mechanisms for Fair Classification", *In proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS) 2017, Fort Lauderdale, Florida, USA.*

Zemel, R., Wu, Y., Swersky, K., Pitassi, T., Dwork, C., 2013, "Learning fair representations", *In proceedings of the international conference on machine learning,* 325 – 333.

Appendix

A. Python Code for Custom λ Regularization

```
def sigmoid(x):
    return 1/(1+np.exp(-x))

def customWeight(W,L):
    totalPenalty = 0
    for i in range(len(W)):
        totalPenalty += L[i+1] * (W[i] ** 2)
    return totalPenalty

def customGradiantWeight(W,L):
    totalPenalty = []
    for i in range(len(W)):
        totalPenalty.append((L[i+1]/m) * (W[i]))
    return totalPenalty

def lrCostFunction(theta_t, X_t, y_t, lambda_t):
    m = len(y_t)
    J = (-1/m) * (y_t.T @ np.log(sigmoid(X_t @ theta_t)) + (1 - y_t.T) @ np.log(1 - sigmoid(X_t @ theta_t)))
    #reg = (lambda_t/(2*m)) * (theta_t[1:].T @ theta_t[1:])
    reg = customWeight(theta_t[1:],lambda_t)/(2*m)
    J = J + reg
    return J

def lrGradientDescent(theta, X, y, lambda_t):
    m = len(y)
    grad = np.zeros([m,1])
    grad = (1/m) * X.T @ (sigmoid(X @ theta) - y)
    grad[1:] = grad[1:] + list(i/m for i in customGradiantWeight(theta[1:],lambda_t))
    return grad
```

```
(m, n) = x_train2.shape
X3 = x_train2
y3 = np.array(y_train2)
theta = np.zeros((n,1))
#Custom_lmbda =[100,100,20,20,20,20,20,100,20,20,20,20,20,100,20]
#Custom_lmbda =[100,100,10,10,10,10,10,100,10,10,10,10,10,100,10]
#Custom_lmbda =[100,100,1,1,1,1,1,100,1,1,1,1,1,100,1]
#Custom_lmbda =[500,500,100,100,100,100,100,500,100,100,100,100,100,500,100]
#Custom_lmbda =[1000,1000,100,100,100,100,100,1000,100,100,100,100,100,1000,100]
#Custom_lmbda =[100,100,50,50,50,50,50,100,50,50,50,50,50,100,50]
#Custom_lmbda = [200,200,50,50,50,50,50,200,50,50,50,50,50,200,50]
#Custom_lmbda  = [150,150,50,50,50,50,50,150,50,50,50,50,50,150,50]
#Custom_lmbda  = [120,120,50,50,50,50,50,120,50,50,50,50,50,120,50]
Custom_lmbda  = [200,120,50,50,50,50,50,150,50,50,50,50,50,100,50]
J = lrCostFunction(theta, X3, y3, Custom_lmbda)
output_regularization = opt.fmin_tnc(func = lrCostFunction, x0 = theta.flatten(), fprime = lrGradientDescent, \
                        args = (X3, y3.flatten(), Custom_lmbda))
theta = output_regularization[0]
pred_regularization = [sigmoid(np.dot(x_val2, theta)) >= 0.5]
x_val2['Pred'] = np.array(pred_regularization).flatten()
x_val2['Actual'] = np.array(y_val2)
accuracy = np.mean(x_val2['Pred'] == x_val2['Actual'])
print('The accuracy of the model is :',accuracy)
cal_metrics(x_val2)
x_val2 = x_val2.drop(columns=['Pred', 'Actual'])
```

B. Logistic Regression Result

Logit Regression Results

Dep. Variable:	Male	No. Observations:	429
Model:	Logit	Df Residuals:	414
Method:	MLE	Df Model:	14
Date:	Sat, 26 Mar 2022	Pseudo R-squ.:	inf
Time:	20:26:48	Log-Likelihood:	-3164.8
converged:	True	LL-Null:	0.0000
Covariance Type:	nonrobust	LLR p-value:	1.000

	coef	std err	z	P>|z|	[0.025	0.975]
ApplicantIncome	0.0001	6.89e-05	2.091	0.036	9.05e-06	0.000
CoapplicantIncome	4.222e-05	4.37e-05	0.967	0.334	-4.34e-05	0.000
LoanAmount	-0.0016	0.003	-0.480	0.631	-0.008	0.005
Loan_Amount_Term	-0.0024	0.002	-0.980	0.327	-0.007	0.002
Credit_History	-0.2290	0.439	-0.521	0.602	-1.090	0.632
Loan_Status	0.7080	0.320	2.209	0.027	0.080	1.336
LoanAmount_log	0.1495	0.248	0.604	0.546	-0.336	0.635
Married_Yes	2.2422	0.332	6.757	0.000	1.592	2.893
Dependents_1	-0.5208	0.412	-1.263	0.207	-1.329	0.287
Dependents_2	0.4793	0.546	0.878	0.380	-0.591	1.550
Dependents_3+	0.2956	0.696	0.425	0.671	-1.068	1.660
Education_Not Graduate	0.4997	0.386	1.295	0.195	-0.257	1.256
Self_Employed_Yes	-0.0771	0.460	-0.168	0.867	-0.978	0.824
Property_Area_Semiurban	-0.9736	0.359	-2.710	0.007	-1.678	-0.269
Property_Area_Urban	-0.0556	0.376	-0.148	0.882	-0.792	0.681

C. Random Forest Result

Random Forest Results

```
Variable: ApplicantIncome      Importance: 0.27
Variable: Married_Yes          Importance: 0.17
Variable: CoapplicantIncome    Importance: 0.12
Variable: LoanAmount           Importance: 0.11
Variable: LoanAmount_log       Importance: 0.11
Variable: Loan_Amount_Term     Importance: 0.05
Variable: Loan_Status          Importance: 0.03
Variable: Dependents_1         Importance: 0.03
Variable: Property_Area_Semiurban Importance: 0.03
Variable: Credit_History       Importance: 0.02
Variable: Education_Not Graduate Importance: 0.02
Variable: Property_Area_Urban  Importance: 0.02
Variable: Dependents_2         Importance: 0.01
Variable: Self_Employed_Yes    Importance: 0.01
Variable: Dependents_3+        Importance: 0.0
```