CMPT-733

Model Transparency & Fairness

A Project on Detecting, Understanding and Mitigating the Bias in Machine Learning Models

|Chidambaram Allada, Manju Malateshappa, Urvi Chauhan, Vignesh Vaidyanathan|

1. Introduction

Bias is a prejudice in favor or against a person, group, or a thing that is considered to be unfair. At present times, Machine Learning and Artificial Intelligence play a major role in taking decisions on behalf of humans, whether it be powering a self-driving car, detecting cancer or predicting our interests based upon the past behavior. But as Machine Learning becomes an integral part of our lives, a key challenge is the presence of bias in the classifications and predictions of machine learning. They have consequences based upon the decisions resulting from a machine learning model. Therefore, it’s important to understand how bias is introduced into machine learning models, how to test the bias, and then how to mitigate it. By developing a fair Model, the decisions taken by the ML model would be therefore unbiased and would enable much more transparency that would benefit all.

2. Motivation & Background

Artificial intelligence (AI) and machine learning (ML) promise a smarter, more automated future for everyone. However the algorithms that underpin these technologies can be at risk of bias, a substantial threat that could undermine their entire purpose. Artificial Intelligence is good but can they be really trusted and if they can be trusted, how to check there isn’t biasness in the Model developed? There is a kind of misconception that AI is absolutely objective. AI is objective only in the sense of Learning what Human teaches. Data provided by the human can be highly bias. Machine Learning, Deep Neural Networks, they rely heavily on the data that is provided to them, on which they are trained. In the event of data being biased, the model that run on this data can show biased results, i.e the results being favorable to a particular group or individual. What will be the repercussions of it? How to detect the bias and try to mitigate it? These were the questions that motivated us to take up the project. We wanted to look into a domain of Machine Learning and Artificial Intelligence which is different from the traditional method of approaching a problem, build a model which would either do a regression or a classification task.

There have been many examples of bias in Machine Learning Models in the past. Few are mentioned as follows-

- Amazon scraps AI recruiting tool that showed bias against women.
- Google Cloud’s Computer Vision Algos are found to be biased against black people.

Unintentionally it may be, but the bias exists in the data which needs to be removed before training the data on the Machine Learning models to make sure that all the groups/races are treated equally without any prejudice. These biases are not benign. They have consequences based upon the decisions resulting from a machine learning model. Therefore, it’s important to understand how bias is introduced into machine learning models, how to test for it, and how to remove it.

2.1 Who cares about this project?

Detecting Bias in the data and removing Bias from the Prediction/Classification Model can serve as an important paradigm shift in the field of AI and Machine Learning. Opaque and potentially biased mathematical models are remaking our lives and neither the companies responsible for developing them nor the companies deploying them as well as the government is interested in addressing the problem. There is a wide area in which the project can be implemented. We have seen few examples above where the Machine bias lead to unfair treatment among certain groups.

Algorithms that may conceal hidden biases are already routinely used to make vital financial and legal decisions. Proprietary algorithms are used to decide, for instance, who gets a job interview, who gets granted parole, and who gets a loan. Hence, Mitigation of Bias can be beneficial in the Financial, Healthcare fields or any other area where a Machine assisted decision can impact particular section of the society or may lead to unfair treatment of certain group on the basis of age/race/gender.

2.2 Related Work

We took up the Compas dataset to analyze the bias and come up with the methods to mitigate the bias and make the Machine Learning Model fair. **COMPAS,** which stands for **Correctional Offender Management Profiling for Alternative Sanctions** is an algorithm used across U.S states to assess a criminal defendant’s likelihood of becoming a **recidivist** — a term used to describe criminals who re-offends. ProPublica, which is an independent investigative journalism initially came up with the article titled MACHINE BIAS and highlighted the significant bias in US judicial system that rated the offenders from Black Race at higher risk of re-offending in comparison with their White counterparts. ProPublica compared the algorithm predictions of committing the crime of the offenders with the ground truth and found that the black defendants who were rated at a higher risk of recidivism committed fewer crimes in future in comparison to the white defendants who were incorrectly flagged at low risk. In our work we analyzed the Bias in the data using the Machine Learning and Deep Learning Models and mitigated the bias using tools, details of which are explained subsequently in the report.

3. Problem Statement

We analyzed the COMPAS dataset to find the bias in the protected attributes (race/gender/age). Primary motive is to detect the bias , mitigate the bias and then check whether our Machine Learning model is able to classify (without being biased against a particular group) if a person will commit crime in two years after being released from jail. Then check the results against the fairness metrics if they have improved or not to validate our method of removing the bias from the Model prediction. One more important part of the project is the Model Transparency. Basically how clear/interpretable our model is, if the results can be clearly conveyed to the end-user. One more part of the problem statement is to find for each individual prediction, what is the supportive evidence?

3.1 Questions that we address:

In our project we address to the problem of mitigating the bias in our data which may have been inadvertently introduced in the data. We check how our model classification changes after removing the bias from the data, that the model learns. Other questions that we answer are as follows.

- How the bias data affects the output of the Machine Learning model and why it should be mitigated?
- How to ensure that the output from the model is a fair result?
- Model Transparency – How to explain our model?

3.2 Challenges

The problems mentioned above are challenging because we need to understand how these bias arises in the first place. Secondly, the standard practices in Data Science are not designed to detect it.

We had to learn new tools like the **IBM AIF360 tool** and **Google**'s **What-If Tool** and also implement it in the project work. The selection of protected attributes, favorable, unfavorable labels and how to arrange the dataset for the different tools were challenging. This topic is completely new domain in the field of Artificial Intelligence and Machine Learning and now more and more companies have started checking into the bias of the data as well as model. The dataset used has 53 features columns, therefore feature selection was one of the challenging task.

In terms of Model Evaluation we had to learn about various **Fairness Metrics** and look for the Accuracy-Fairness trade-off.

4. Data Science Pipeline

The whole data science pipeline is shown as follows.

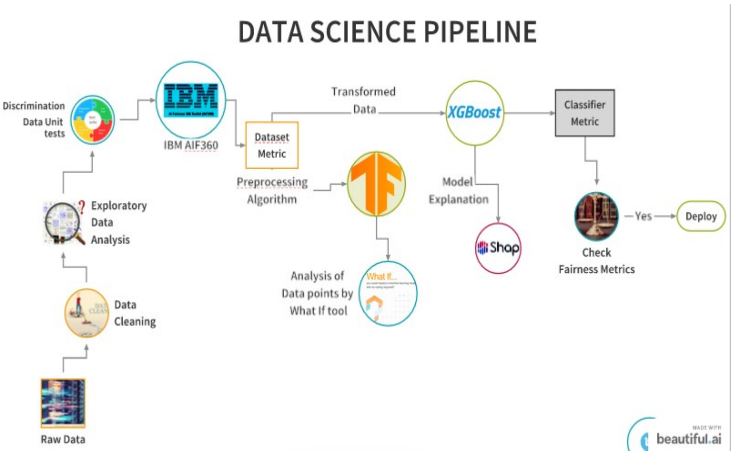


Fig.1 Data Science Pipeline

4.1 Data Collection

As stated by our mentor to choose any public dataset, we decided to web scrape the COMPAS Data from the Northpointe's website, which is the maker of the COMPAS algorithm. However they had not made it public. Propublica had acquired the data from Broward County Sheriff's Office in Florida through a public request for analyzing their algorithm. We obtained the same data containing two years worth of COMPAS scores from Propublica.

4.2 Data Cleaning

We used pandas(Python-Data Analysis Library) to read the dataset in the form of dataframe. Raw Dataset had **53 feature columns** of which many rows had missing values. There were also empty spaces that were replaced by Null values and then subsequently replaced by suitable values.

4.3 Exploratory Data Analysis

We used Dataprep, Seaborn and Matplotlib for EDA part. Through EDA, we got insights about the feature columns and the data distribution through Histograms. We also checked for the outliers through Dataprep analysis. Detected Correlations between the features using Heatmap.

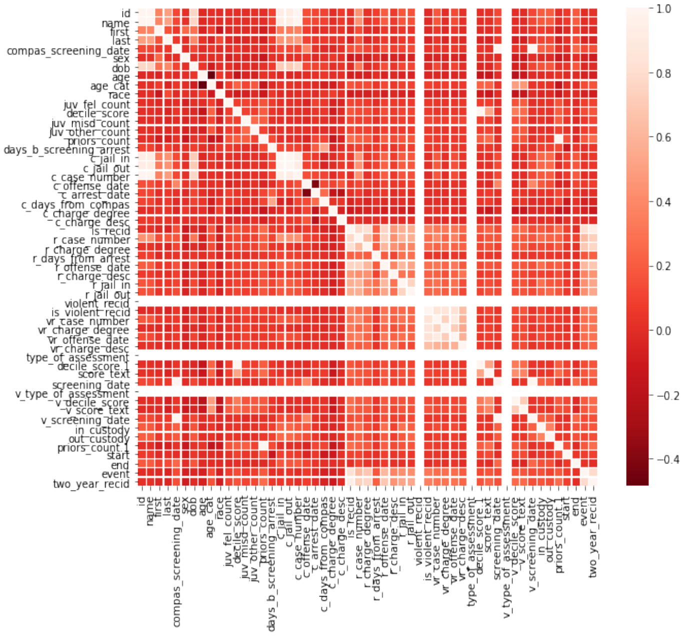


Fig.2 HeatMap showing Correlation

Plotting Initial Findings :

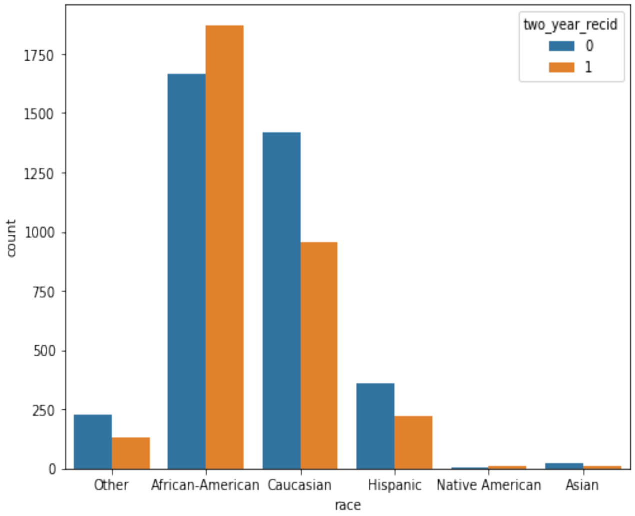


Fig.3 Distribution of Race in Data

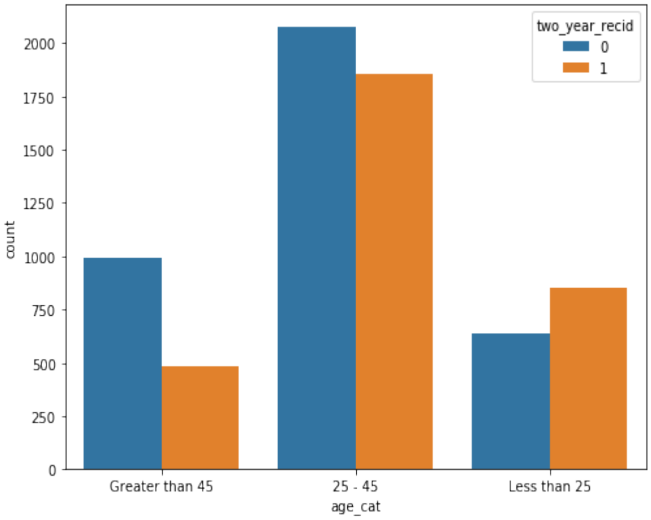


Fig.4 Age Distribution in the Data

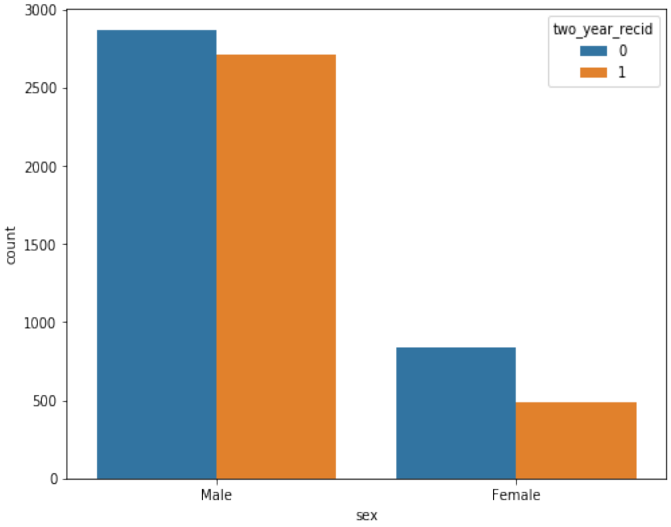


Fig.5 Gender Distribution in Data

two_year_recid – If the offender commits crime again within two years.

The above plots show the distribution of race, age and gender. There is improper data collection and distribution of races from the initial findings that may have lead to the biasness. We check for some more findings to see if there is a relation between race and their corresponding COMPAS scores (corresponding to the likelihood of re-offending in near future).

Some more plots are as follows.

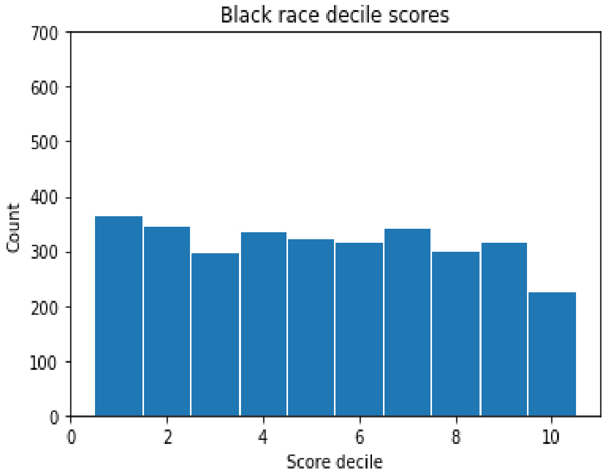


Fig.6 African-American Race Decile scores

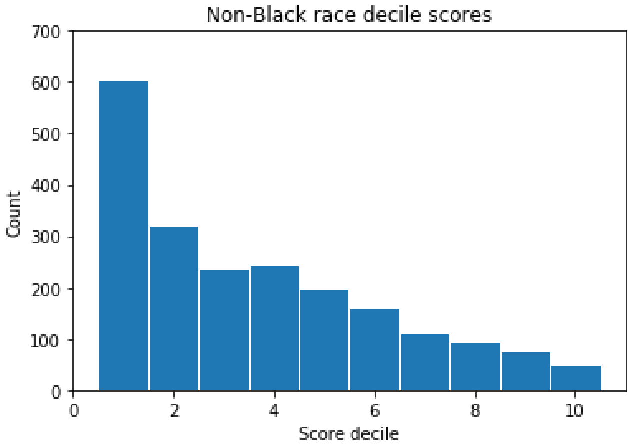
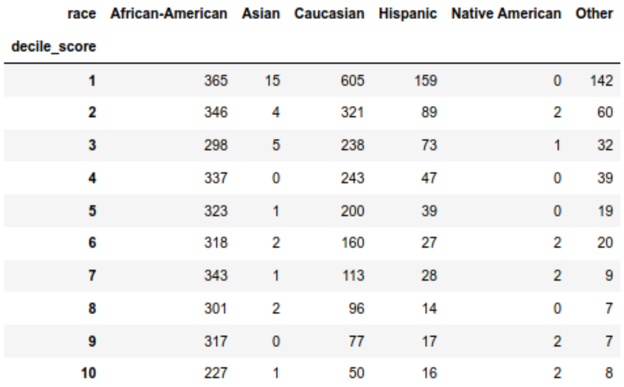


Fig.7 White Race Decile scores

decile_score - scores of offenders are arranged in rank order from lowest to highest. -the scores are divided into 10 equally sized groups or bands on the basis of severity of the crime.

From the above EDA, we can find that for African-American race the decile score remains the same more or less but for the White race, the number of offenders in the data decreases with increase in the decile scores.



race decile_score	African-American	Asian	Caucasian	Hispanic	Native American	Other
1	365	15	605	159	0	142
2	346	4	321	89	2	60
3	298	5	238	73	1	32
4	337	0	243	47	0	39
5	323	1	200	39	0	19
6	318	2	160	27	2	20
7	343	1	113	28	2	9
8	301	2	96	14	0	7
9	317	0	77	17	2	7
10	227	1	50	16	2	8

Fig.8 decile_score for individual races

We can see that there is a clear downtrend in decile scores as those scores increase for white defendants.

4.4 Feature Selection & engineering

Initially we checked for the missing values in the in individual feature columns. If the percentage of missing values in the rows was more than 10% of the total rows, we dropped that column. Manually replacing the missing values through mean/median would result in data being biased .

Through correlation heatmap, we found the feature columns are correlated between each other and selected those features to be dropped that were represented very well by some other feature columns in the dataset. This had to be done to avoid the redundancy of the data fro the Machine Learning models. There were few columns like prior_count and prior_count1(prior_count is the number of offenses prior to the current offense) which had similar values. Some columns had the same value for all rows , for eg. type_of_assessment which had the single string data as ‘Risk of Recidivism’. Columnns like this were also further dropped as they do not add much importance to the prediction of two_year_recidivism as the row values are same throughout.

Some columns like 'charge_description' had many unique values and using it as the feature column would result in several other columns if one hot encoding is done. Since the charge_description column had correlation with the score_text (low/high/medium), we decided to drop the column.

On few columns, we did **Feature engineering.** Like to check if the duration of jail tenure relates with the target output of re-offending the crime within two years. Changed the datetype to total days n time duration. However the correlation was less with the two_year_recidivism and decile_score.

correlation between length of stay and two_year_recid is: 0.109

correlation between length of stay and decile_score: 0.207

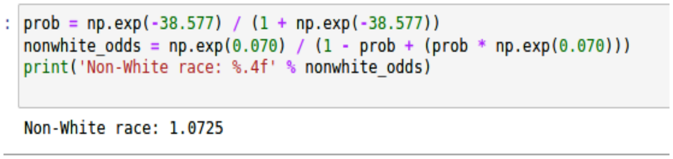
We also checked for the 'age' column. There were 65 unique values and therefore changed it to age category for better representation of data and later for the model explanation part. After all the attributes selection and data manipulation part we are left with 6172 rows and 13 columns. After doing one-hot encoding for the categorical columns we had 6172 rows and 24 columns.

4.5 Fairness tests

In any data, bias occurs in any of the *protected attributes* like age,race and gender. *Protected attributes* are attribute that partition a population into groups that have difference in terms of parity received. In order to check for the bias in the data we implemented few methods using scikit learn ML library and also use **IBM AIF360** and **Google's What-If tool**.

In scikit learn we used the RandomForest feature importance and Logistic Regression log-odds. From the RandomForest feature importance table found that the feature importance of Caucasian race is much lower than the African-American race. This shows the importance of features when the fit is performed.

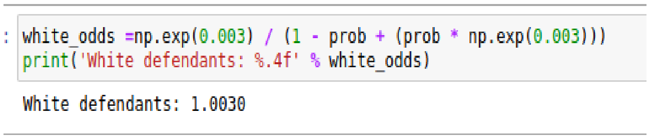
In Logistic regression we used the log-odds and the observations were as follows.



```
prob = np.exp(-38.577) / (1 + np.exp(-38.577))
nonwhite_odds = np.exp(0.070) / (1 - prob + (prob * np.exp(0.070)))
print('Non-White race: %.4f' % nonwhite_odds)
```

```
Non-White race: 1.0725
```

Fig.9 Log-Odds for African-American Race



```
white_odds =np.exp(0.003) / (1 - prob + (prob * np.exp(0.003)))
print('White defendants: %.4f' % white_odds)
```

```
White defendants: 1.0030
```

Fig.10 Log-Odds for Caucasian Race

We took the compas score(score_text) as the target ouput and found the log odds for the individual races. From the above results we find that to get the Higher score_text the odds are more in favor of African-American race and less in Caucasian race around close to 40%.

Further analysis of bias was done using IBM AIF360 tool. We checked for the bias in the gender and race column. Changed the datatypes alternately between *privileged* and *unprivileged* group. *Privileged* value of a protected attribute indicates a group that has historically been at systematic advantage. We checked for the fairness metric- *Disparate Impact* using **AIF360** MetricTest explainers. The metric lied between 0.8-1.2 in all other combinations but for African-American race as the unprivileged and

Caucasian as the privileged group, the **Disparate Impact** was found to be **0.7**(less than 0.8) Value less than 1 indicates higher benefit for the privilege group. In addition to that we also implemented *TensorFlow* Network on the *Google Colab* along with **What-If tool** to see the comparison between a data point and the next-closest data point where the model predicts a different result.

4.6 Data Preparation

Data was split into training and test data using train-test split in scikit learn. In order to mitigate the bias AIF360 Binary Label Dataset was used to transform the data using favorable/unfavorable labels and privileged/unprivileged groups.

4.7 Machine Learning Models

RandomForest and *XGBoost* model were used from scikit learn and *TensorFlow Sequential Model* was used in Google Colab.

4.8 Explanation of Evaluation methods

We evaluate the model classification on the basis of the improvement in the fairness metrics when the model was trained on the Transformed Dataset.

Fairness metrics considered are as follows :

- **Disparate Impact** - This is the ratio of probability of favorable outcomes between the unprivileged and privileged groups.
- **Average odds difference** - This is the average of difference in false positive rates and true positive rates between unprivileged and privileged groups. A value of 0 implies both groups have equal benefit.
- **Equal opportunity difference** - This is the difference in true positive rates between unprivileged and privileged groups. A value of 0 implies both groups have equal benefit.

Confusion Matrix and classification report have been taken into consideration while evaluating the model. Our aim is to reduce the bias for the African- American race. Earlier many offenders of Black race were incorrectly identified as the recidivists. But now after running the model on the transformed dataset, the *False Positive Rate* should decrease indicating the fairness in the model.

4.9 Model Transparency

Model Transparency is also one of the main aspect of our project where we try to explain the Model predictions and which all feature contribute towards the final prediction and how the individual data points behave. For this purpose we use SHAP and What-If tool with better visualization and explanation.

5. Methodology

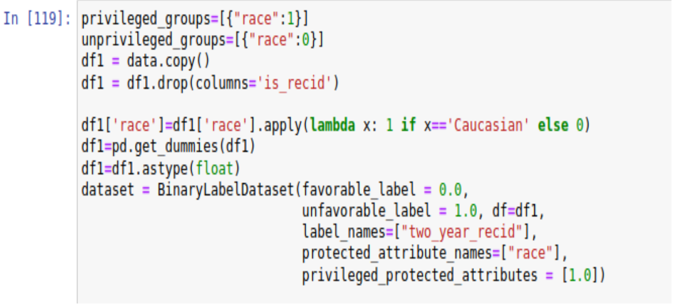
5.1 Tools Used

- IBM AIF360
- Google What-If
- Shap

5.2 Bias Mitigation

As discussed above we used AIF360 explainers and What-If tool along with the EDA to find the bias in the dataset. Through insights from What-If and applying Disparate Impact fairness metric, we found out that the bias arises in the African-American race and that is why the offenders from the African-American race are placed at higher risk of recidivism in comparison to their White counterparts.

First of all the Machine Learning model was trained on original data. For this the dataset was transformed into Binary Label Dataset which is required by the IBM AIF360 tool. We selected the *protected attributes* in which the bias was there as race and choose the *privileged* as well as unprivileged *groups*. We denoted the Caucasian race as the privileged group and labeled it as 1 in the dataset. Rest all data belonging to race were labeled as 0. Label was used as two_year_recid.



```
In [119]: privileged_groups=[{"race":1}]
          unprivileged_groups=[{"race":0}]
          df1 = data.copy()
          df1 = df1.drop(columns='is_recid')

          df1['race']=df1['race'].apply(lambda x: 1 if x=='Caucasian' else 0)
          df1=pd.get_dummies(df1)
          df1=df1.astype(float)
          dataset = BinaryLabelDataset(favorable_label = 0.0,
                                       unfavorable_label = 1.0, df=df1,
                                       label_names=["two_year_recid"],
                                       protected_attribute_names=["race"],
                                       privileged_protected_attributes = [1.0])
```

Fig.11 Using BinaryLabelDataset in AIF360

the fairness of the model was evaluated through the standard fairness metrics. We also evaluated the confusion matrix to get the False positive rate (FPR). Then the original data was transformed using pre-processing algorithm in AIF360 which is called ***Reweighing***. The Reweighted data was then passed through the models (RandomForest and XGBoost) for training and testing part and once again the fairness metrics as well as the FPR was measured. We implemented the Reweighing method because instead of changing/editing the feature values (as in case of Disparate Impact remover pre-processing algorithm), we wanted to generate weights for the training in each (group, label) combination differently to ensure fairness before classification and also explain the data points through the explainer tools.

We even implemented the same methodology for mitigating the bias through Reweighing in TensorFlow sequential model in Google Colab. However the Disparate Impact rate and FPR obtained was not satisfactory. This may be because neural networks require large dataset for training. Our transformed dataset had 6172 rows. May be for that reason the neural network model didn't perform well.

Idea behind using IBM AIF360 tool was it's easy integration with the Scikit's learn fit/predict paradigm. Designed to be used easily in python or by neural networks. We were able to detect the bias as well as mitigate the bias, thereby improving the fairness of the model. In AIF360, there are various bias mitigation algorithms available like the pre-processing, in-processing and post-processing algorithm. The inbuilt Classification Metrics library enable to get the results efficiently.

Next, for explaining the Model Transparency we used the What-If tool and SHAP. The visualizations for them are explained in the Evaluation part below.

What-If tool was used as it is a user-friendly interface for expanding understanding of the black-box classification and regression models. We could perform inference on a large set of examples and immediately visualize the results in variety of ways. The *most beneficial part of using the tool was that the plugin also provides a tab for investigating model performance and fairness over subsets of Data.*

SHAP was used to give a Global Interpretability as well as local interpretability for the data points. Variable importance plot was used to show the positive or negative relationship for each feature feature variable with the target. Since we used Tree-Based models, SHAP could be implemented very well for them. The Variable dependence plot shows which feature has maximum interaction with the other feature while predicting an output.

6. Evaluation

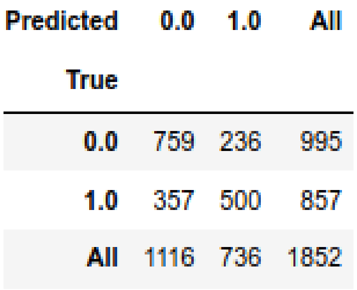
As mentioned earlier we used the fairness metrics before and after the mitigation of bias to evaluate the fairness of the model. The results are as follows.

RandomForest:

	Original Data	Transformed Data
Disparate Impact	0.743	0.833
Average Odds Difference	-0.154	-0.082
Equality of Opportunity	-0.117	-0.059

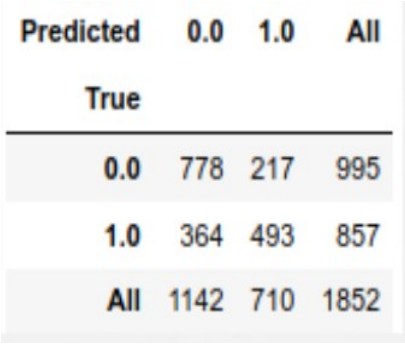
False Positive Rate(Confusion Matrix)

Original Data



Predicted	0.0	1.0	All
True			
0.0	759	236	995
1.0	357	500	857
All	1116	736	1852

Transformed Data



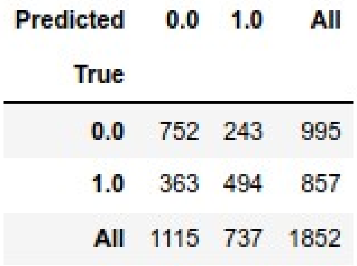
Predicted	0.0	1.0	All
True			
0.0	778	217	995
1.0	364	493	857
All	1142	710	1852

XGBoost

	Original Data	Transformed Data
Disparate Impact	0.768	0.916
Average Odds Difference	-0.132	-0.021
Equality of Opportunity	-0.104	0.002

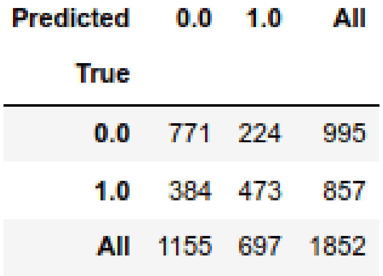
False Positive Rate(Confusion Matrix)

Original Data



Predicted	0.0	1.0	All
True			
0.0	752	243	995
1.0	363	494	857
All	1115	737	1852

Transformed Data



Predicted	0.0	1.0	All
True			
0.0	771	224	995
1.0	384	473	857
All	1155	697	1852

From the above results we see that XGBoost model with hyper-parameter tuning gave us the best results. Our motive was to increase the Disparate Impact fairness metric above 0.8. With XGBoost, we got the fairness as 0.91, well above the 0.8 limit. In both the RandomForest and XGBoost model we found that the metrics Average Odds difference and Equality of opportunity have increased from their initial negative values to close to zero. XGBoost model gave better fairness result on the Equality of opportunity. Value close to zero indicates that both group have equality of benefit. Thus, we can say that we were successfully able to improve the fairness metrics after applying the Reweighing pre processing Algorithm to mitigate the bias.

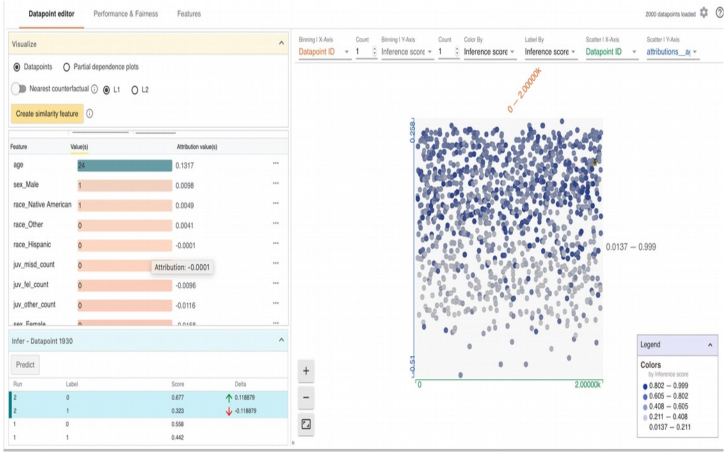
Now with the False positive rate, we can see that FPR has reduced in both the models. Significant decrease was seen in XGBoost model. FPR indicates the incorrect classification. These were the offenders belonging to the African-American race who were incorrectly labeled as Higher risk group. After mitigating the bias we see that the False positive rate has decreased and subsequently the True positive rate has increased in both the cases. Thus we can conclude that our bias mitigation tool has performed well.

Our result makes sense because after using the pre-processing reweighing algorithm there is overall improvement in the Fairness Metrics and decrease in False Positive Rate.

6.1 Model Transparency

As stated above, we implemented SHAP and What-If tool for Model Transparency. Following are few visulaizations of it.

What-If Tool

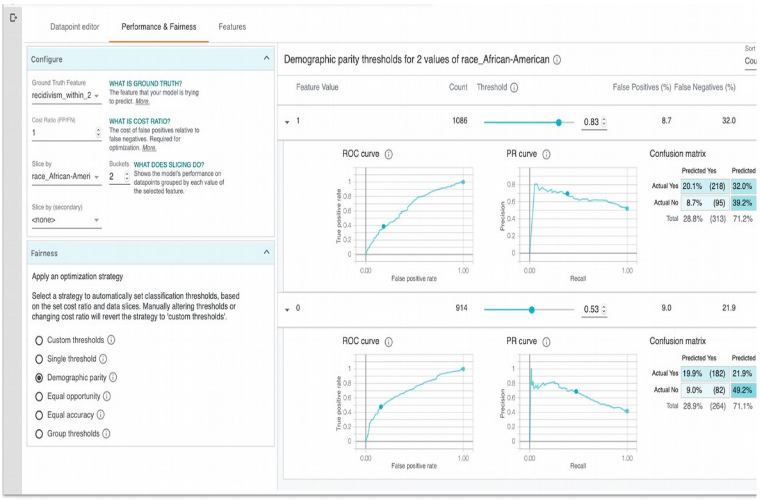


Feature	Value(s)	Attribution value(s)
age	24	0.1317
sex_Male	1	0.0098
race_Native American	1	0.0049
race_Other	0	0.0041
race_Hispanic	0	-0.0001
juv_misd_count	0	
juv_fel_count	0	-0.0096
juv_other_count	0	-0.0116

Infer - Datapoint 1930

Run	Label	Score	Delta
2	0	0.677	↑ 0.118879
2	1	0.323	↓ -0.118879
1	0	0.558	
1	1	0.442	

The above Image shows the change in the prediction when a data point is changed from race African_American to race Caucasian (Native-American).


Datapoint editor
Performance & Fairness
Features
Configure
WHAT IS GROUND TRUTH?
The feature that your model is trying to predict. More.
WHAT IS COST RATIO?
The cost of false positives relative to false negatives. Required for optimization. More.
WHAT DOES SLICING DO?
Shows the model's performance on datapoints grouped by each value of the selected feature.
Fairness
ROC curve
True positive rate
False positive rate
PR curve
Precision
Recall
0.00
1.00
Sort
Cou

Ground Truth Feature: recidivism_within_2

Cost Ratio (FP/FN): 1

Slice by: race_African-Ameri

Buckets: 2

Slice by (secondary)
<none>

Apply an optimization strategy

Select a strategy to automatically set classification thresholds, based on the set cost ratio and data slices. Manually altering thresholds or changing cost ratio will revert the strategy to 'custom thresholds'.

○ Custom thresholds
○ Single threshold
● Demographic parity
○ Equal opportunity
○ Equal accuracy
○ Group thresholds

Demographic parity thresholds for 2 values of race_African-American

Feature Value	Count	Threshold	False Positives (%)	False Negatives (%)
1	1086	0.83	8.7	32.0
0	914	0.53	9.0	21.9

Confusion matrix

	Predicted Yes		Predicted
Actual Yes	20.1%	(218)	32.0%
Actual No	8.7%	(95)	39.2%
Total	28.8%	(313)	71.2%

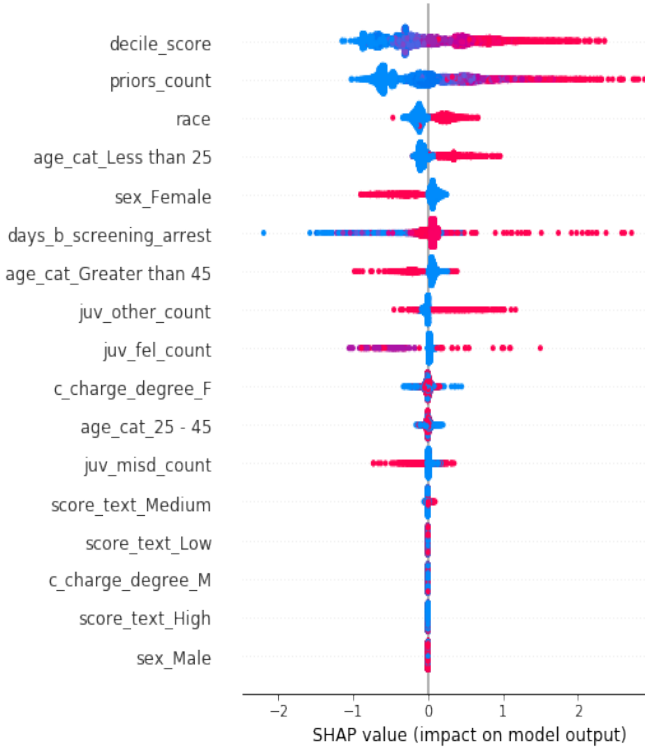
Confusion matrix

	Predicted Yes		Predicted
Actual Yes	19.9%	(182)	21.9%
Actual No	9.0%	(82)	49.2%
Total	28.9%	(264)	71.1%

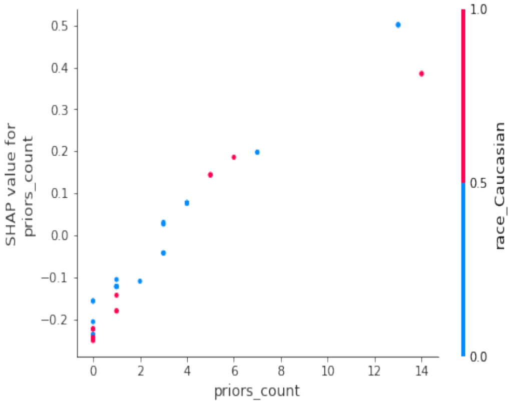
The above image shows the investigation of model fairness using *Demographic Parity* (Fairness Metric).

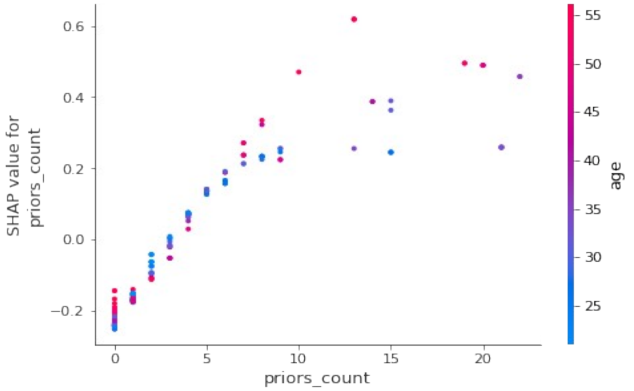
SHAP

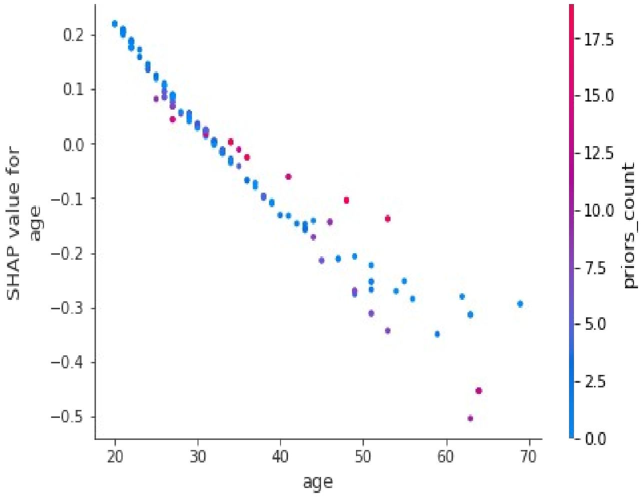
The following figure shows the feature Importance plot

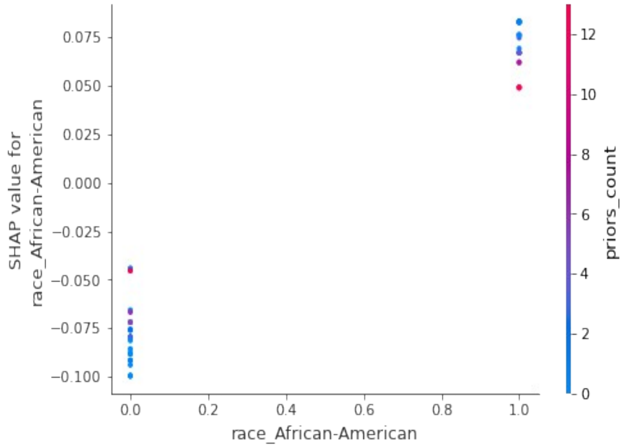


From the above Feature importance plot we see that the features decile_score and priors_count are significant for the target output. The red color indicates the positive impact i.e, how much positively the feature is correlated with the target.









The above plots show which features have the maximum interaction while predicting the target output. We see that age has maximum interaction with the feature column prior_count.

7. Data Product

The Data product that we have are the findings from our analysis which can give a better sense to any one looking for an approach to detect the bias and mitigate it so that overall a fair model is built. We have used the insights from our project like the methods, Visualizations and plots and hosted it online.

We created a full stack dynamic Node.js app using the following technologies : Express, Jquery, BootStrap, ejsengine. We hosted this website on Herokuapp. Link to the website is

https://model-fariness-project-demo.herokuapp.com/

We hope this website can be used to gain insights about the tools used and build a Fair model.

8. Lessons Learned

The project had a steep learning curve to it and took us while to figure out the in-depth understanding of several components. The learning curve includes problems like detecting the bias in the data and how to approach. We had to learn about the new tools like IBM AIF360 and Google's What-If and implement it in our project.

This project was completely different from the Assignments or the course projects where we are given some specific task of taking the data and apply ML models to do regression or classification task. Detecting Bias and mitigating it was completely new. We have to go through the IBM AIF360 tool documentation and choose the algorithm which would suit well to our requirements and for the data. We learned about the protected attributes, the privileged and unprivileged groups and how the Disparate Impact works. One other significant achievement from this project was gaining knowledge about the What-If tool and how to use the platform to get more insights from the data.

We also learnt about various fairness metrics, the concepts behind them and how they can be used to test a model for bias. Technology-wise our groups achievement was to come up with two new tools in our project to achieve the end results. We also learnt about the Accuracy-Fairness trade-off from the project. We learnt how to transform the data required for these tools to be applied. We also learned how to build a comprehensive and robust web application with appealing frontend and connect it with our backend server written by Nodejs.

9. Summary

Our project aims to make the Machine Learning model fair by mitigating the bias in data. We have used tools that help in analyzing the dataset that gives us the understanding of data and would enable in detecting the bias based on the fairness metrics. We started by analyzing the COMPAS dataset where we found the algorithm was biased towards the African-American Race. They were placed at higher risk of recidivism, however the ground truth didn't match the algorithm predictions. We used EDA, Machine Learning Models, AIF360 explainers and What-If tool to understand the distribution of data across various protected attributes (race/ gender/ age) and detect the bias. We used Reweighing-pre-processing algorithm for bias mitigation and the results obtained were checked against the standard fairness metrics. We found significant improvement in the Fairness metrics after bias removal and also found decrease in False Positive Rate which means incorrect predictions. We explained the model through SHAP explainers and What-If tool. The results and the visualization that we obtained were hosted online which would be helpful for anyone who would like to know more about checking the biasness in the model.

Concluding, we can say that we have shown a method that can be used to mitigate the bias of the Machine Learning Model and hence improve the Fairness of the model built. We have also shown the Model Transparency by explaining the Model in a simple and interpretable manner.

10. Future Work

We have implemented the bias mitigation metric during pre-processing. There are other bias mitigation metrics that facilitate the removal of bias in post-processing. We can look into that methodology. There are other ML models explainers such as LIME, AEQUITAS that can be used to gain better insights about the attributes of the dataset.

We would even like to implement the same methodology to detect bias in other datasets like Health related or Finance related, where the decisions made by the machine can have a significant impact on certain sections of the society.

11. References

1. https://arxiv.org/pdf/1810.01943.pdf
2. https://www.ibm.com/blogs/research/2018/09/ai-fairness-360/
3. https://www.technologyreview.com/2017/07/12/150510/biased-algorithms-are-everywhere-and-no-one-seems-to-care/
4. https://medium.com/@tonyxu_71807/ensuring-fairness-and-explainability-in-credit-default-risk-modeling-shap-ibm-tool-kit-aix-360-bfc519c191bf
5. https://sloanreview.mit.edu/article/the-risk-of-machine-learning-bias-and-how-to-prevent-it/