IE685: MSc-PhD Research Project 1
Topic: An Empirical Study on Model Fairness

Rishabh Agnihotri | 22N0456

Supervisor: Prof. Vishnu Narayanan

Industrial Engineering and Operations Research
Indian Institute of Technology, Bombay

Autumn, 2023-24

Acknowledgement

I would like to thank Prof. Vishnu Narayanan for his guidance and support in this project. His encouragement had enabled me to solve real-world problems. His advice and guidance have been remarkable in my learning.

Contents

Chapter 1

Introduction

Machine learning models are increasingly being used in important decision-making software such as approving bank loans, recommending criminal sentencing, hiring employees, and so on. It is important to ensure the fairness of these models so that no discrimination is made based on protected attribute (e.g., race, sex, age) while decision making. So we know that using Machine Learning model is have risk of biasness and unfairness which may cost of life of a human being may lead any company to a huge loss and so on. So it is important to ensure that the prediction is not biased toward any protected attribute. ML fairness has been studied for about past 10 years and many of the fairness metrics and mitigation techniques has been developed. Perhaps we don not able to recognize how much fairness issues is there in ML models from practice. According to paper main problem that is ”Do the models exhibit bias?” If the answer is yes, what are the different types of bias and how we can construct model related to the bias?

Recently, Holstein et al. conducted and empirical study on ML fairness by surveying and interviewing industry practitioners. They give the outlined of challenge faced by the developers and the support they need to build and model the fair ML systems. So, in this project and the taken research paper I have conducted empiricial study on ML models to understand biasness and fairness as discussed above. In the project I analyzed the fairness of 40 ML models collected from a crowd sourced platform Kaggle. In this project I try to answer these three questions:

RQ1: (Unfairness) What are the unfairness measures of the ML models in the wild, and which of them are more or less prone to bias?

RQ2: (Bias mitigation) What are the root causes of the bias in ML models and what kind of techniques can successfully mitigate those bias?

RQ3: (Impact) What are the impacts of applying different bias mitigating techniques on ML models?

Chapter 2

Methodology, Measures and Bias Mitigation Techniques

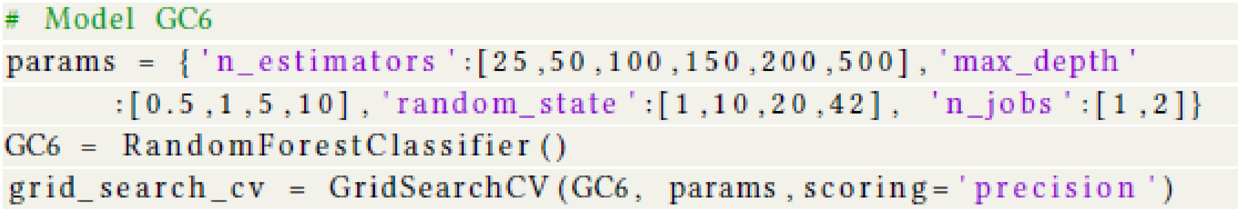
In this part of the report of the project I have shown the methodology to create the benchmark of ML models for fairness analysis. Then I have described the different metrics like metrics based on base rates, individual fairness etc. that I used for fairness and biasness analysis for experiment which was taken in this project. The bias mitigation techniques that was supposed to apply in this project are like pre-processing algorithms, in processing algorithms and post-processing algorithms. The algorithm of Preprocessing algorithms is it do not change the model and only work on the dataset before training so that model can produce fairer predictions. Inprocessing algorithms works in the original model prediction it modify the ML model to mitigate the bias. Post-processing algorithms not works on ML models or in the input data but it modifies the prediction result.

2.1 Benchmark Collection

In this section, I have discussed benchmark collection for the implementation of the project. As we know Kaggle is one of the most popular data science platform owned by Google. Many data scientist , data science researchers and many company developers can host or participate in data science competition. By taking part in competition, all participant share their dataset, task that they do and solution in kaggle. There are 376 competitions and 28,622 datasets in kaggle upto date when the taken research paper was published.

In this project, I have collected 40 kernels from the kaggle. In the each kernel there is code, description of task and their solutions based on that. Firstly I have analyzed ML models that operates on datasets utilized by prior studies on fairness and then datasets with protected attributes (e.g., sex, race). With the taken above criteria for each category I have collected the ML models with different filtering criteria. The overall process for benchmark collection was shown in Figure 2.1.

In previous studies about fairness in computer models, researchers used different sets of data to test how fair the models were. For example, Galhotra and others used two datasets called German Credit and Adult Census. Udeshi and team focused on the Adult Census dataset, and Aggarwal and team used six datasets: German Credit, Adult Census, Bank Marketing, US Executions, Fraud Detection, and Raw Car Rentals. These datasets are available on Kaggle, a platform for data science. They looked at the solutions people came up with for these datasets, collecting a total of 440 different approaches (65 for German Credit, 302



```
# Model GC6
params = {'n_estimators':[25,50,100,150,200,500],'max_depth'
      :[0.5,1,5,10],'random_state':[1,10,20,42], 'n_jobs':[1,2]}
GC6 = RandomForestClassifier()
grid_search_cv = GridSearchCV(GC6, params,scoring='precision')
```

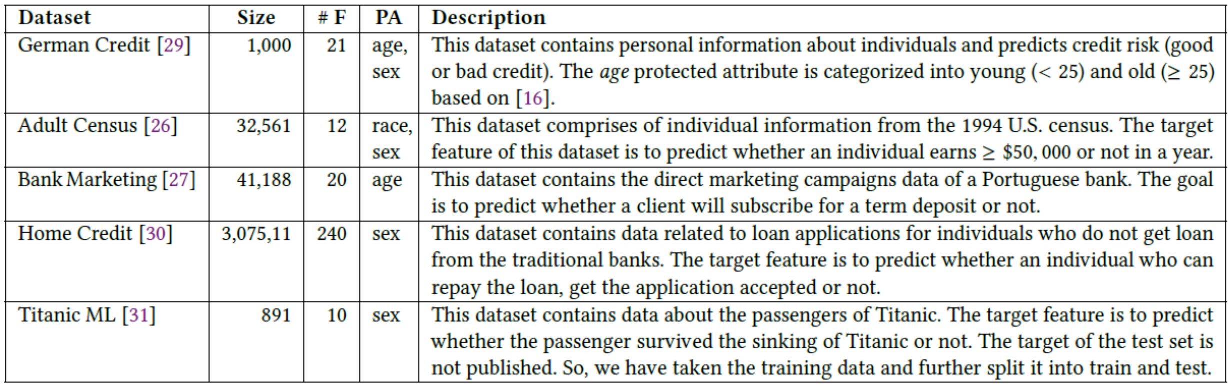
Figure 2.1: Benchmark model collection process

for Adult Census, and 73 for Bank Marketing). To narrow down our selection, they only considered the best ones. They chose kernels (code notebooks) based on three criteria: 1) they had predictive models (some just had data analysis), 2) they had at least 5 upvotes (indicating they were well-liked), and 3) their accuracy was at least 65 percent. So, in the end, we selected the top 8 models with the most upvotes for each of the 3 datasets, giving us a total of 24 machine learning models to study.

There is one more research that I studied during my project work that is the Chen and team identified 12 attributes like age, sex, and race for fairness analysis. They looked into Kaggle competitions and found 7 that included these attributes. However, they only selected competitions where predicting outcomes was linked to fairness concerns. For example, one competition involved predicting customer age and sex but was about recommending products, not fairness. After filtering, they found two suitable competitions with multiple solutions. They applied the same criteria as before, selecting 8 models with the most upvotes for each dataset. In the end, we compiled a benchmark of 40 top-rated Kaggle models working on 5 datasets.

The characteristics of the datasets and tasks in the benchmark are shown in below table.

Table 1: The datasets used in the fairness experimentation. # F: Feature count. PA: Protected attribute.



Dataset	Size	# F	PA	Description
German Credit [29]	1,000	21	age, sex	This dataset contains personal information about individuals and predicts credit risk (good or bad credit). The *age* protected attribute is categorized into young ($<$ 25) and old ($\geq$ 25) based on [16].
Adult Census [26]	32,561	12	race, sex	This dataset comprises of individual information from the 1994 U.S. census. The target feature of this dataset is to predict whether an individual earns $\geq \$50,000$ or not in a year.
Bank Marketing [27]	41,188	20	age	This dataset contains the direct marketing campaigns data of a Portuguese bank. The goal is to predict whether a client will subscribe for a term deposit or not.
Home Credit [30]	3,075,11	240	sex	This dataset contains data related to loan applications for individuals who do not get loan from the traditional banks. The target feature is to predict whether an individual who can repay the loan, get the application accepted or not.
Titanic ML [31]	891	10	sex	This dataset contains data about the passengers of Titanic. The target feature is to predict whether the passenger survived the sinking of Titanic or not. The target of the test set is not published. So, we have taken the training data and further split it into train and test.

2.2 Experiment Design

Whole process for experiment design is explained in the above Fig. 2.2.

In experiment I, after creating our benchmark of 40 top-rated kaggle models, I conducted experiments to test their performance and fairness. I applied different techniques to address biasness and evaluated the impact. In the benchmark, I used models from five dataset categories. To compare fairness across different

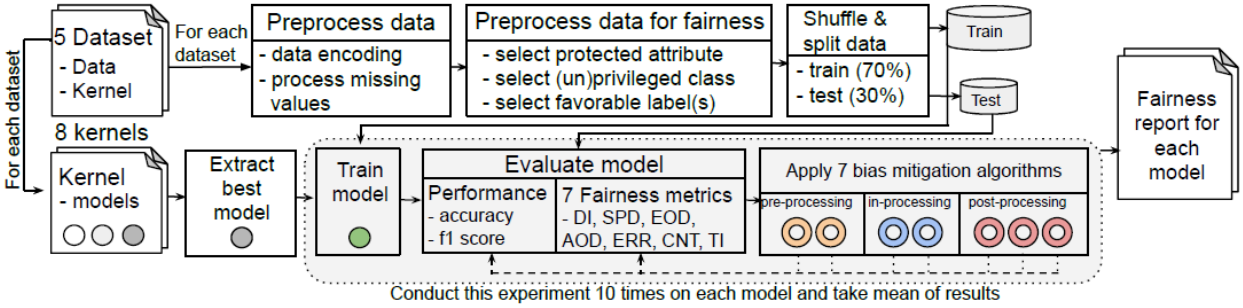


Figure 2.2: Experimentation to compute performance, fairness and mitigation impacts of machine learning models.

models in each category, I applied the same data pre-processing strategy. This involved handling missing or invalid values, transforming age into categories (e.g., young or old), and converting non-numerical features into numerical ones (e.g., female: 0, male: 1).

For fairness analysis, I specified protected attributes, privileged and unprivileged groups, and the favorable label or prediction outcome. For example, in the Home Credit dataset, sex is the protected attribute, with male as the privileged group, female as the unprivileged group, and the prediction label being credit risk (good or bad).

I standardized the dataset preparation by shuffling and splitting it into a ratio of 70:30 train-test ratio before feeding it to the models. Each dataset category had eight Kaggle kernels, representing Python code for solving classification problems. The kernels typically covered data exploration, pre-processing, feature selection, modeling, training, evaluation, and prediction.From these kernels, I manually extracted code for modeling, training, and evaluation. Some kernels tried multiple models, with the best-performing model chosen. If a kernel didn't specify the best model, I selected the one with the highest accuracy. For instance, a kernel working on the Adult Census dataset implemented four models, and we chose the Gradient Boosting classifier for its superior accuracy.

After extracting the best model from the above process, I train the model and evaluate performance (accuracy, F1 score). After that I have evaluated 7 different metrics that I have described in next section.

2.3 Measures

I have used various different types of metrics in this project and also used in this research paper for fairness that is our main goal. I have computed the algorithmic features of each subject model in our benchmark collection process that I described in intial part of chapter 2 in this report.Suppose D =(X,Y,Z) be a dataset where X is the training data, Y is the binary classification label (Y=1 if the label is favourable, otherwise Y =0), Z is the protected attribute (Z=1 for privileged group, otherwise Z=0), and $\hat{Y}$ is the prediction label (i.e., 1 for favourable decision and 0 for unfavourable decision). If there are multiple groups for protected attributes, we have employed a binary grouping strategy (e.g., race attribute in Adult Census dataset has been changed to white/non-white). In next subsection I have all metrics that is used in this project.

2.3.1 Accuracy Measure

I have computed the performance in terms of accuracy, and F1 score, before measuring fairness of the model.

Accuracy: Accuracy is given by the ratio of truly classified items and total number of items.

$$\text{Accuracy} = \frac{\text{No. of True Positive} + \text{No. of True Negative}}{\text{No. of Total}} \tag{2.1}$$

F1 Score: This metric is given by the harmonic mean of precision and recall.

$$F1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2.2}$$

2.3.2 Fairness Measure

Lots of fairness metrics have been suggested in the research world, like AIF 360 toolkit has APIs for computing 71 fairness metrics. In our study, we chose a selection of metrics, not covering all but a good representation. We followed the suggestions of Friedler and team and added some individual fairness metrics to evaluate how fair our machine learning models are.

Metrics based on base rates:

Disparate Impact(DI): This metric is given by the ratio between the probability of unprivileged group gets favorable prediction and the probability of privileged group gets favorable prediction.

$$DI = \frac{P\left[\hat{Y} = 1 \mid Z = 0\right]}{P\left[\hat{Y} = 1 \mid Z = 1\right]} \tag{2.3}$$

Statistical Parity Difference (SPD): This measure is similar to DI but instead of the ratio of probabilities, difference is calculated.

$$SPD = P\left[\hat{Y} = 1 \mid Z = 0\right] - P\left[\hat{Y} = 1 \mid Z = 1\right] \tag{2.4}$$

Metrics based on group conditional rates:

Equal Opportunity Difference (EOD): This is given by the true-positive rate (TPR) difference between unprivileged and privileged groups.

$$TPR(u) = P\left[\hat{Y} = 1 \mid Y = 1, Z = 0\right] \tag{2.5}$$

$$TPR(p) = P\left[\hat{Y} = 1 \mid Y = 1, Z = 1\right] \tag{2.6}$$

$$EOD = TPR(u) - TPR(p) \tag{2.7}$$

Average Odds Difference (AOD): This is given by the average of false-positive rate (FPR) difference and true-positive rate difference between unprivileged and privileged groups.

$$FPRu = P\left[\hat{Y} = 1 \mid Y = 0, Z = 0\right] \tag{2.8}$$

$$FPRp = P\left[\hat{Y} = 1 \mid Y = 0, Z = 1\right] \tag{2.9}$$

$$AOD = \frac{1}{2}\left\{(FPRu - FPRp) + (TPRu - TPRp)\right\} \tag{2.10}$$

Error Rate Difference (ERD): Error rate is given by the addition of false-positive rate (FPR) and false-negative rate (FNR).

$$ERR = FPR + FNR \tag{2.11}$$

$$ERD = ERRu - ERRp \tag{2.12}$$

Metrics based on individual fairness:

Consistency(CNT): This individual fairness metric measures how similar the predictions are when the instances are similar. Here, n$_{n}eighborsisthenumberofneighborsfortheKNNalgorithm.$

$$CNT = 1 - \frac{1}{n \cdot n_{\text{neighbors}}} \sum_{i=1}^{n} \left| \hat{y}_i - \frac{1}{\mathcal{N}_{n_{\text{neighbors}}}(x_i)} \sum_{j \in \mathcal{N}_{n_{\text{neighbors}}}(x_i)} \hat{y}_j \right| \tag{2.13}$$

Theil Index (TI): This metric is also called the entropy index which measures both the group and individual fairness. Theil index is given by the following equation where

$$TI = \frac{1}{n} \sum_{i=1}^{n} \frac{b_i}{\mu} \ln\left(\frac{b_i}{\mu}\right) \tag{2.14}$$

2.4 Bias Mitigation Techniques

In this section, we have discussed the bias mitigation techniques that have been applied to the models. These techniques can be broadly classified into pre-processing, in-processing, and post-processing approaches.

Preprocessing Algorithms. Preprocessing algorithms do not change the model and only work on the dataset before training so that models can produce fairer predictions.

Reweighing: In a biased dataset, different weights are assigned to reduce the effect of favoritism of a specific group. If a class of input has been favored, then a lower weight is assigned in comparison to the class not been favored.

Disparate Impact Remover: This algorithm is based on the concept of the metric DI that measures the fraction of individuals achieves positive outcomes from an unprivileged group in comparison to the privileged group. To remove the bias, this technique modifies the value of protected attribute to remove distinguishing factors.

In-processing Algorithms: In-processing algorithms modify the ML model to mitigate the bias in the original model prediction.

Adversarial Debiasing: This approach modifies the ML model by introducing backward feedback (negative gradient) for predicting the protected attribute. This is achieved by incorporating an adversarial model that learns the difference between protected and other attributes that can be utilized to mitigate the bias.

Prejudice Remover Regularizer: If an ML model relies on the decision based on the protected attribute, we call that direct prejudice. In order to remove that, one could simply remove the protected attribute or regulate the effect in the ML model. This technique applies the latter approach, where a regularizer is implemented that computes the effect of the protected attribute.

Post-processing Algorithms: This genre of techniques modifies the prediction result instead of the ML models or the input data.

Equalized Odds (E): This approach changes the output labels to optimize the EOD metric. In this approach, a linear program is solved to obtain the probabilities of modifying prediction.

Calibrated Equalized Odds: To achieve fairness, this technique also optimizes EOD metric by using the calibrated prediction score produced by the classifier.

Reject Option Classification: This technique favors the instances in privileged group over unprivileged ones that lie in the decision boundary with high uncertainty.

Chapter 3

Unfairness in ML Models and Mitigation

3.1 Unfairness

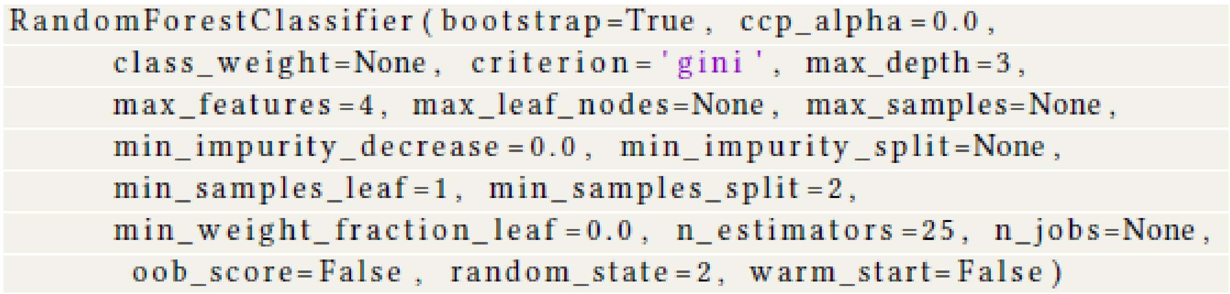
We have found that all the models exhibit unfairness and models specific to a dataset show similar bias patterns. From Figure 3.1, we can see that all the models exhibit bias with respect to most of the fairness metrics. For a model, metric values vary since the metrics follow different definitions of fairness. Therefore, we have compared bias of different models both cumulatively and using the specific metric individually. To compare total bias across all the metrics, we have taken the absolute value of the measures and computed the sum of bias for each model. In Figure 3.2, we can see the total bias exhibited by the models. Although the bias exhibited by models for each dataset follow similar pattern, certain models are fairer than others.

Finding 1: Model optimization goals seek overall performance improvement, which is causing unfairness.

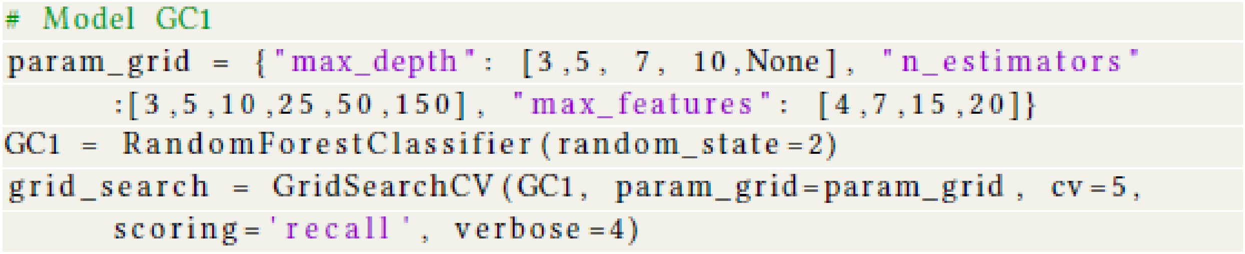
Among German Credit models, GC1 stands out with the least bias. It's a Random Forest (RFT) classifier created through a grid search for the best hyperparameters. The winning classifier configuration is detailed in the code provided.

On the other hand, GC6, also a Random Forest classifier from a grid search, isn't as fair based on cumulative bias and individual metrics , except for error rate difference (ERD). To understand this fairness difference, we tested both models by tweaking one hyperparameter at a time. The key reason for the fairness gap was the scoring mechanism. GC1 uses scoring='recall', while GC6 uses scoring='precision,' as indicated in the provided code snippet.

Upon delving deeper into the German Credit dataset, it's revealed that the data rows contain personal information about individuals, and the task is to predict their credit risk. Notably, the dataset isn't balanced regarding the sex of individuals, with 69 percent being male and 31 percent female. Optimizing the model towards recall (GC1) rather than precision (GC6) results in a decrease in true positives and an increase in false negatives. Since there are more instances for the privileged group (male), the decrease in true positives raises the chance of the unprivileged group (female) being incorrectly classified as unfavorable. This is why GC1 is considered more fair than GC6, even though its accuracy is lower. Unlike other fairness metrics, error rate difference (ERD) considers the difference in false-positive and false-negative



```
RandomForestClassifier(bootstrap=True, ccp_alpha=0.0,
      class_weight=None, criterion='gini', max_depth=3,
      max_features=4, max_leaf_nodes=None, max_samples=None,
      min_impurity_decrease=0.0, min_impurity_split=None,
      min_samples_leaf=1, min_samples_split=2,
      min_weight_fraction_leaf=0.0, n_estimators=25, n_jobs=None,
       oob_score=False, random_state=2, warm_start=False)
```



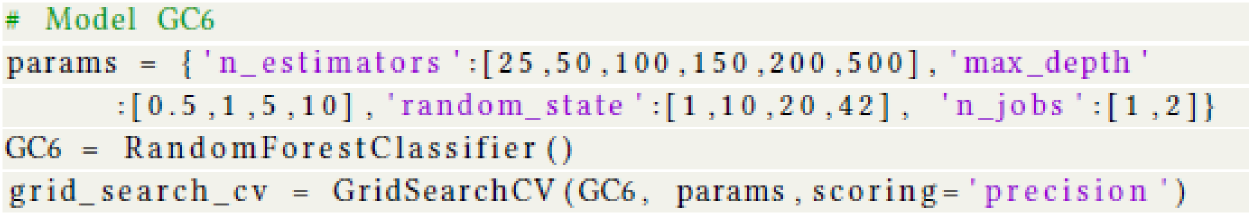
```
# Model GC1
param_grid = {"max_depth": [3,5, 7, 10,None], "n_estimators"
      :[3,5,10,25,50,150], "max_features": [4,7,15,20]}
GC1 = RandomForestClassifier(random_state=2)
grid_search = GridSearchCV(GC1, param_grid=param_grid, cv=5,
      scoring='recall', verbose=4)
```

rates between privileged and unprivileged groups. Optimizing for recall increases the total number of false negatives. We observed that the percentage of males categorized as favorable is less than the percentage of females categorized as favorable. Consequently, the overall increase in false negatives also raises the error rate for the unprivileged group, making GC1 more biased than GC6 in terms of ERD.

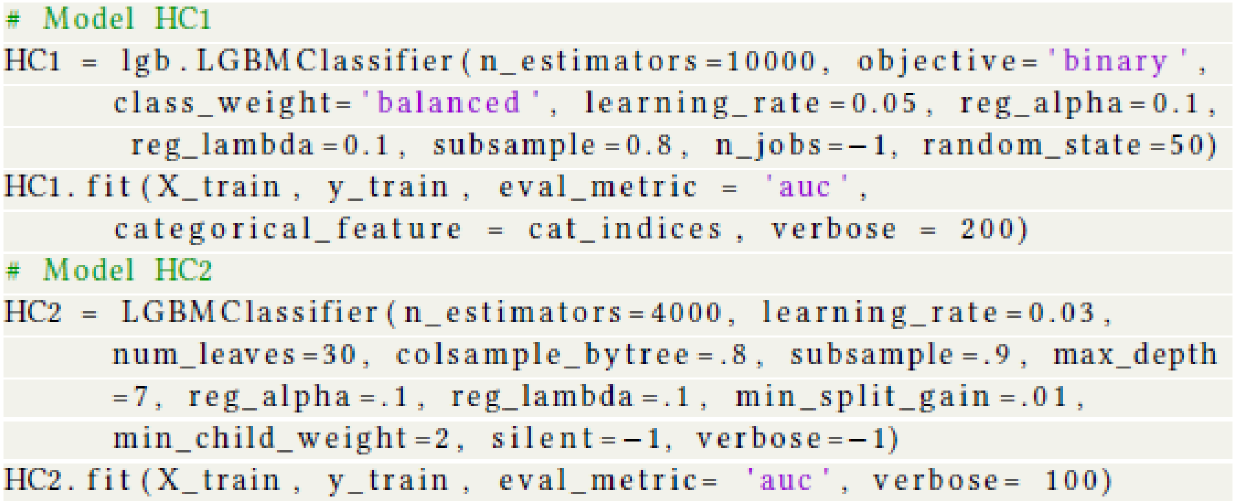
Finding 2: Libraries for model creation do not explicitly mention fairness concerns in model constructs.

From figure, we can see that HC1 and HC2 show difference in most of the fairness metrics, while operating on the same dataset i.e., Home Credit. HC2 is fairer than HC1 with respect to all the metrics except DI. From Table 2, we can see that HC1 has positive bias, whereas HC2 exhibit negative bias. This indicates that HC1 is biased towards unprivileged group and HC2 is biased towards privileged group. We have found that HC1 and HC2 both are using Light Gradient Boost (LGB) model for prediction.

We have executed both the models with varied hyperparameter combinations and found that `class_weight='balanced'` is causing HC1 not to be biased towards the privileged group. By specifying `class_weight`, we can set more weight to the data items belonging to an infrequent class. Higher class weight implies that the data items are getting more emphasis in prediction. When the class weight is set to `balanced`, the model automatically accounts for class imbalance and adjusts the weight of data items inversely proportional to the frequency of the class . In this case, HC1 mitigates the male-female imbalance in its prediction. Therefore,



```
# Model GC6
params = {'n_estimators':[25,50,100,150,200,500],'max_depth'
      :[0.5,1,5,10],'random_state':[1,10,20,42], 'n_jobs':[1,2]}
GC6 = RandomForestClassifier()
grid_search_cv = GridSearchCV(GC6, params,scoring='precision')
```



```
# Model HC1
HC1 = lgb.LGBMClassifier(n_estimators=10000, objective='binary',
      class_weight='balanced', learning_rate=0.05, reg_alpha=0.1,
       reg_lambda=0.1, subsample=0.8, n_jobs=-1, random_state=50)
HC1.fit(X_train, y_train, eval_metric = 'auc',
      categorical_feature = cat_indices, verbose = 200)
# Model HC2
HC2 = LGBMClassifier(n_estimators=4000, learning_rate=0.03,
      num_leaves=30, colsample_bytree=.8, subsample=.9, max_depth
      =7, reg_alpha=.1, reg_lambda=.1, min_split_gain=.01,
      min_child_weight=2, silent=-1, verbose=-1)
HC2.fit(X_train, y_train, eval_metric= 'auc', verbose= 100)
```

it does not exhibit bias towards the privileged group (male). On the other hand, HC2 has less bias but it is biased towards privileged group. Although we want models to be fair with respect to all groups and individuals, trade-off might be needed and in some cases, bias toward unprivileged may be a desirable trait. We have observed that `class_weight` hyperparameter in `LGBMClassifier` allows developers to control group fairness directly. However, the library documentation of LGB classifier suggests that this parameter is used for improving performance of the models . Though the library documentation mentions about probability calibration of classes to boost the prediction performance using this parameter, however, there is no suggestion regarding the effect on the bias introduced due to the wrong choice of this parameter.

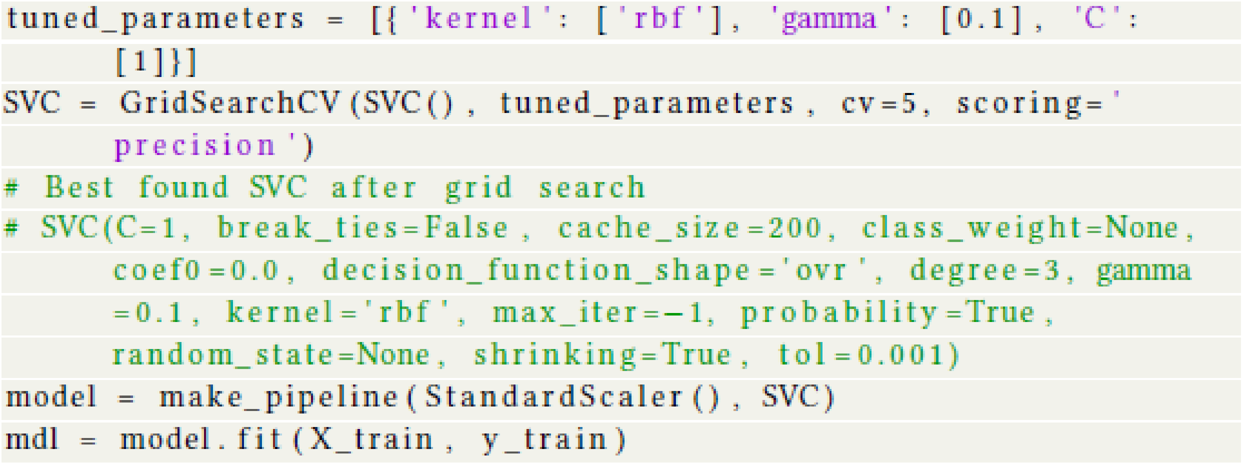
Finding 3:Standardizing features before training models can help to remove disparity between groups in the protected class.

We found that using 'StandardScaler' in the model pipeline makes BM5 fairer. Particularly, the disparate impact (DI) of BM5 is 0.14, while the mean DI of the other seven BM models is much higher (0.74). 'StandardScaler' transforms data features independently, setting their mean value to 0 and standard deviation to 1. This is crucial when features have varying magnitudes, preventing the model from overly relying on one dominant feature. In the Bank Marketing dataset with 55 features, 41 have a mean close to 0 ([0, 0.35]). However, age, the protected attribute, has a mean value of 0.97 (older: 1, younger: 0) due to the significant imbalance in the number of older individuals. Consequently, age becomes a dominant feature in these BM models. BM5 mitigates this effect by applying standard scaling to all features, balancing the importance of the protected feature with others. Understanding the properties of protected features is crucial for reducing bias in models and improving predictions.

Finding 4: Dropping a feature from the dataset can change the model fairness effectively

Both AC5 and AC6 use the XGBoost (XGB) classifier, but AC6 is fairer. In terms of consistency (CNT), AC5 shows 3.61 times more bias than AC6. We looked into the model construction and found three differences: features used, number of trees, and learning rate. Changing the number of trees and learning rate didn't affect bias. AC5 excluded a feature related to years of education, considering another categorical feature covered education types (e.g., bachelor's, doctorate). AC6 uses all features. CNT measures how similar individuals are classified differently. Excluding education years in AC5 leads to classifying similar individuals differently, causing individual unfairness.

Finding 5: Different metrics are needed to understand bias in different models.



```
tuned_parameters = [{'kernel': ['rbf'], 'gamma': [0.1], 'C':
        [1]}]
SVC = GridSearchCV(SVC(), tuned_parameters, cv=5, scoring='
        precision')
# Best found SVC after grid search
# SVC(C=1, break_ties=False, cache_size=200, class_weight=None,
        coef0=0.0, decision_function_shape='ovr', degree=3, gamma
        =0.1, kernel='rbf', max_iter=-1, probability=True,
        random_state=None, shrinking=True, tol=0.001)
model = make_pipeline(StandardScaler(), SVC)
mdl = model.fit(X_train, y_train)
```

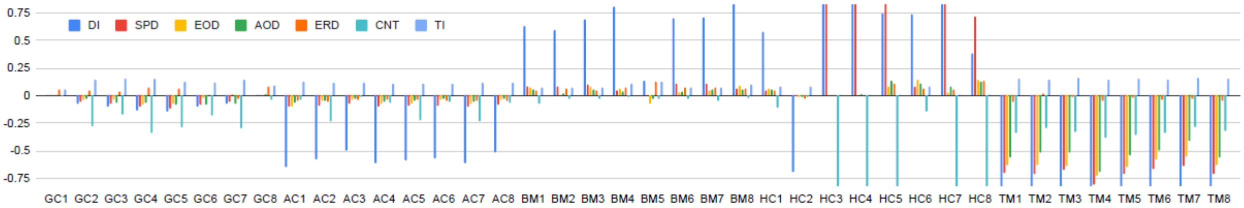


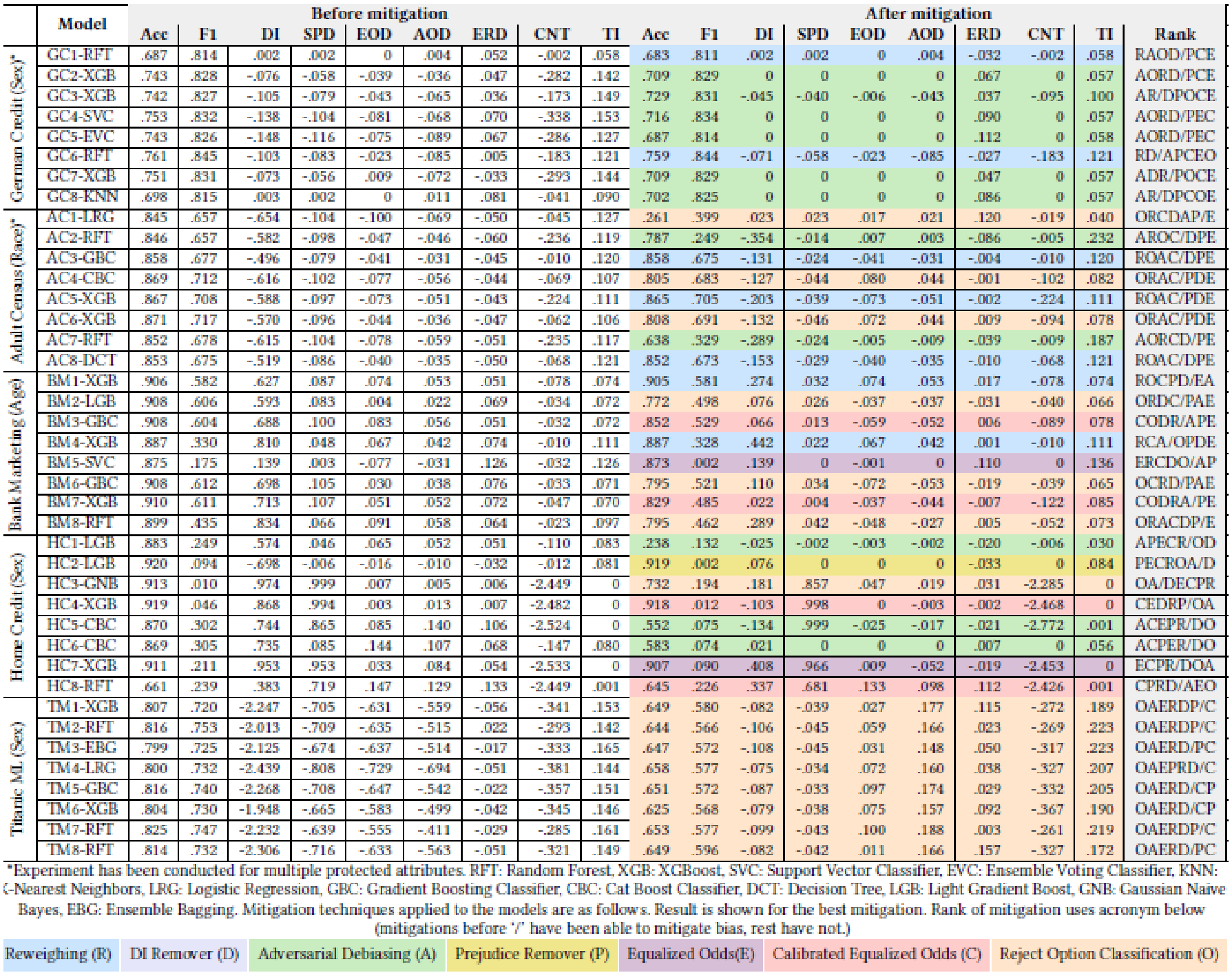
Figure 3.1: Unfairness exhibited by the ML models with respect to different metrics

The models exhibit diverse bias patterns across different fairness metrics. For instance, BM5 in Bank Marketing has a DI less than half but an ERD more than twice compared to other models. Relying only on DI might make the model seem fairer than it truly is. Similarly, GC6 is fairer than 90 percent of models in total bias but only fairer than 50 percent in terms of CNT.

Achieving fairness across all metrics is challenging and, in some cases, mathematically impossible. The definition of fairness varies based on the application and stakeholders. Hence, reporting a comprehensive set of fairness measures and considering trade-offs between them is crucial for building fair models. Some metric pairs, such as (SPD, EOD) and (SPD, AOD), show similar correlations in both datasets due to their definitions using the same or correlated rates. While many metric pairs have positive or negative correlations, there is no consistent pattern in correlation values between the two datasets. For example, CNT and TI are highly negatively correlated in German Credit models but positively correlated in Titanic ML models. Hence, a comprehensive set of metrics is necessary to evaluate fairness.

Table 3.1: Caption

Table 2: Unfairness measures of the models before and after the mitigations



	Model	Before mitigation									After mitigation									Rank
		Acc	F1	DI	SPD	EOD	AOD	ERD	CNT	TI	Acc	F1	DI	SPD	EOD	AOD	ERD	CNT	TI	
German Credit (Sex)*	GC1-RFT	.687	.814	.002	.002	0	.004	.052	-.002	.058	.683	.811	.002	.002	0	.004	-.032	-.002	.058	RAOD/PCE
	GC2-XGB	.743	.828	-.076	-.058	-.039	-.036	.047	-.282	.142	.709	.829	0	0	0	0	.067	0	.057	AORD/PCE
	GC3-XGB	.742	.827	-.105	-.079	-.043	-.065	.036	-.173	.149	.729	.831	-.045	-.040	-.006	-.043	.037	-.095	.100	AR/DPOCE
	GC4-SVC	.753	.832	-.138	-.104	-.081	-.068	.070	-.338	.153	.716	.834	0	0	0	0	.090	0	.057	AORD/PEC
	GC5-EVC	.743	.826	-.148	-.116	-.075	-.089	.067	-.286	.127	.687	.814	0	0	0	0	.112	0	.058	AORD/PEC
	GC6-RFT	.761	.845	-.103	-.083	-.023	-.085	.005	-.183	.121	.759	.844	-.071	-.058	-.023	-.085	-.027	-.183	.121	RD/APCEO
	GC7-XGB	.751	.831	-.073	-.056	.009	-.072	-.033	-.293	.144	.709	.829	0	0	0	0	.047	0	.057	ADR/POCE
	GC8-KNN	.698	.815	.003	.002	0	.011	.081	-.041	.090	.702	.825	0	0	0	0	.086	0	.057	AR/DPCOE
Adult Census (Race)*	AC1-LRG	.845	.657	-.654	-.104	-.100	-.069	-.050	-.045	.127	.261	.399	.023	.023	.017	.021	.120	-.019	.040	ORCDAP/E
	AC2-RFT	.846	.657	-.582	-.098	-.047	-.046	-.060	-.236	.119	.787	.249	-.354	-.014	.007	.003	-.086	-.005	.232	AROC/DPE
	AC3-GBC	.858	.677	-.496	-.079	-.041	-.031	-.045	-.010	.120	.858	.675	-.131	-.024	-.041	-.031	-.004	-.010	.120	ROAC/DPE
	AC4-CBC	.869	.712	-.616	-.102	-.077	-.056	-.044	-.069	.107	.805	.683	-.127	-.044	.080	.044	-.001	-.102	.082	ORAC/PDE
	AC5-XGB	.867	.708	-.588	-.097	-.073	-.051	-.043	-.224	.111	.865	.705	-.203	-.039	-.073	-.051	-.002	-.224	.111	ROAC/PDE
	AC6-XGB	.871	.717	-.570	-.096	-.044	-.036	-.047	-.062	.106	.808	.691	-.132	-.046	.072	.044	.009	-.094	.078	ORAC/PDE
	AC7-RFT	.852	.678	-.615	-.104	-.078	-.059	-.051	-.235	.117	.638	.329	-.289	-.024	-.005	-.009	-.039	-.009	.187	AORCD/PE
	AC8-DCT	.853	.675	-.519	-.086	-.040	-.035	-.050	-.068	.121	.852	.673	-.153	-.029	-.040	-.035	-.010	-.068	.121	ROAC/DPE
Bank Marketing (Age)	BM1-XGB	.906	.582	.627	.087	.074	.053	.051	-.078	.074	.905	.581	.274	.032	.074	.053	.017	-.078	.074	ROCPD/EA
	BM2-LGB	.908	.606	.593	.083	.004	.022	.069	-.034	.072	.772	.498	.076	.026	-.037	-.037	-.031	-.040	.066	ORDC/PAE
	BM3-GBC	.908	.604	.688	.100	.083	.056	.051	-.032	.072	.852	.529	.066	.013	-.059	-.052	006	-.089	078	CODR/APE
	BM4-XGB	.887	.330	.810	.048	.067	.042	.074	-.010	.111	.887	.328	.442	.022	.067	.042	.001	-.010	.111	RCA/OPDE
	BM5-SVC	.875	.175	.139	.003	-.077	-.031	.126	-.032	.126	.873	.002	.139	0	-.001	0	.110	0	.136	ERCDO/AP
	BM6-GBC	.908	.612	.698	.105	.030	.038	.076	-.033	.071	.795	.521	.110	.034	-.072	-.053	-.019	-.039	.065	OCRD/PAE
	BM7-XGB	.910	.611	.713	.107	.051	.052	.072	-.047	.070	.829	.485	.022	.004	-.037	-.044	-.007	-.122	.085	CODRA/PE
	BM8-RFT	.899	.435	.834	.066	.091	.058	.064	-.023	.097	.795	.462	.289	.042	-.048	-.027	.005	-.052	.073	ORACDP/E
Home Credit (Sex)	HC1-LGB	.883	.249	.574	.046	.065	.052	.051	-.110	.083	.238	.132	-.025	-.002	-.003	-.002	-.020	-.006	.030	APECR/OD
	HC2-LGB	.920	.094	-.698	-.006	-.016	-.010	-.032	-.012	.081	.919	.002	.076	0	0	0	-.033	0	.084	PECROA/D
	HC3-GNB	.913	.010	.974	.999	.007	.005	.006	-2.449	0	.732	.194	.181	.857	.047	.019	.031	-2.285	0	OA/DECPR
	HC4-XGB	.919	.046	.868	.994	.003	.013	.007	-2.482	0	.918	.012	-.103	.998	0	-.003	-.002	-2.468	0	CEDRP/OA
	HC5-CBC	.870	.302	.744	.865	.085	.140	.106	-2.524	0	.552	.075	-.134	.999	-.025	-.017	-.021	-2.772	.001	ACEPR/DO
	HC6-CBC	.869	.305	.735	.085	.144	.107	.068	-.147	.080	.583	.074	.021	0	0	0	.007	0	.056	ACPER/DO
	HC7-XGB	.911	.211	.953	.953	.033	.084	.054	-2.533	0	.907	.090	.408	.966	.009	-.052	-.019	-2.453	0	ECPR/DOA
	HC8-RFT	.661	.239	.383	.719	.147	.129	.133	-2.449	.001	.645	.226	.337	.681	.133	.098	.112	-2.426	.001	CPRD/AEO
Titanic ML (Sex)	TM1-XGB	.807	.720	-2.247	-.705	-.631	-.559	-.056	-.341	.153	.649	.580	-.082	-.039	.027	.177	.115	-.272	.189	OAERDP/C
	TM2-RFT	.816	.753	-2.013	-.709	-.635	-.515	.022	-.293	.142	.644	.566	-.106	-.045	.059	.166	.023	-.269	.223	OAERDP/C
	TM3-EBG	.799	.725	-2.125	-.674	-.637	-.514	-.017	-.333	.165	.647	.572	-.108	-.045	.031	.148	.050	-.317	.223	OAERD/PC
	TM4-LRG	.800	.732	-2.439	-.808	-.729	-.694	-.051	-.381	.144	.658	.577	-.075	-.034	.072	.160	.038	-.327	.207	OAEPRD/C
	TM5-GBC	.816	.740	-2.268	-.708	-.647	-.542	-.022	-.357	.151	.651	.572	-.087	-.033	.097	.174	.029	-.332	.205	OAERD/CP
	TM6-XGB	.804	.730	-1.948	-.665	-.583	-.499	-.042	-.345	.146	.625	.568	-.079	-.038	.075	.157	.092	-.367	.190	OAERD/CP
	TM7-RFT	.825	.747	-2.232	-.639	-.555	-.411	-.029	-.285	.161	.653	.577	-.099	-.043	.100	.188	.003	-.261	.219	OAERDP/C
	TM8-RFT	.814	.732	-2.306	-.716	-.633	-.563	-.051	-.321	.149	.649	.596	-.082	-.042	.011	.166	.157	-.327	.172	OAERD/PC

*Experiment has been conducted for multiple protected attributes. RFT: Random Forest, XGB: XGBoost, SVC: Support Vector Classifier, EVC: Ensemble Voting Classifier, KNN: K-Nearest Neighbors, LRG: Logistic Regression, GBC: Gradient Boosting Classifier, CBC: Cat Boost Classifier, DCT: Decision Tree, LGB: Light Gradient Boost, GNB: Gaussian Naive Bayes, EBG: Ensemble Bagging. Mitigation techniques applied to the models are as follows. Result is shown for the best mitigation. Rank of mitigation uses acronym below (mitigations before '/' have been able to mitigate bias, rest have not.)

Reweighing (R)	DI Remover (D)	Adversarial Debiasing (A)	Prejudice Remover (P)	Equalized Odds(E)	Calibrated Equalized Odds (C)	Reject Option Classification (O)

3.2 Mitigation

In this section, we explored the fairness outcomes of the models following the application of bias mitigation techniques. We individually applied seven different bias mitigation algorithms to each of the 40 models, comparing the fairness results with the original model outcomes. For each model, we identified the most effective mitigation algorithm and visualized the fairness values post-mitigation.

Finding 6: Models with effective preprocessing mitigation technique is preferable than others.

We discovered that the Reweighing algorithm effectively debiased several models: GC1, GC6, AC3, AC5, AC8, BM1, and BM4. These models exhibited fairer results when the dataset was pre-processed with Reweighing, indicating they didn't inherently propagate bias. In cases where pre-processing techniques were ineffective, model changes or prediction alterations were necessary, suggesting bias induction or propagation by the models. Notably, in these Reweighing-processed models, accuracy and F1 score were retained post-mitigation, unlike other mitigation techniques that often impacted model performance. For a few models (GC3, GC8, AC1, AC2, AC4, AC6, BM2, BM5, BM8), Reweighing ranked as the second most successful mitigation algorithm. In specific cases (AC1, AC2, BM2, BM5), the most successful algorithm, though reducing bias, led to a loss of at least 22 percent in accuracy or F1 score. In contrast, Reweighing maintained both accuracy and F1 score in these instances.

Finding 7: Models with more bias are debiased effectively by post-processing techniques, whereas originally fairer models are debiased effectively by pre-processing or in-processing techniques.

From Table 2, we observe that 21 out of 40 models are debiased by one of the three post-processing algorithms: Equalized Odds (EO), Calibrated Equalized Odds (CEO), and Reject Option Classifier (ROC). These algorithms effectively mitigated bias (not necessarily the most successful) in 90 percent of the models, with ROC and CEO being dominant. ROC assigns favorable outcomes to the unprivileged group and unfavorable outcomes to the privileged group with a confidence around the decision boundary. CEO utilizes the probability distribution score from the classifier to maximize equalized odds . EO changes the outcome label based on a certain probability obtained by solving a linear program . These post-processing methods proved more effective when the original model produced more biased results. Figure 4 shows the most biased models (TM4, TM8, TM5, TM1, HC7), where post-processing was highly successful. Conversely, for the five least biased models (GC1, GC8, BM5, GC6, GC3), all three post-processing techniques increased bias rather than mitigating it.

Chapter 4

Impact and Results

4.1 Impact

While mitigating bias, there is a possibility that the model's performance may be compromised. The most successful algorithm in debiasing a model doesn't always ensure good performance. Consequently, developers often need to make trade-offs between fairness and performance. In this section, we delve into addressing Research Question 3 (RQ3): What are the impacts of applying bias mitigation algorithms to the models?

We scrutinized the accuracy and F1 score of the models post-application of the mitigation algorithms. Initially, we examined the impacts of the most effective mitigation algorithms for removing bias on each model. In the figure, we have plotted the change in accuracy, F1 score, and total bias when the most successful mitigating algorithms are applied. We can see that while mitigating bias, many models are losing their performance.

Finding 8: When mitigating bias effectively, in-processing mitigation algorithms show uncertain behavior in their performance.

Among in-processing algorithms, Adversarial Debiasing has proven most effective in 11 models (GC2, GC3, GC4, GC5, AC2, AC7, HC1, HC5, HC6), while Prejudice Remover has been the most effective in one model (HC2). For German Credit models, Adversarial Debiasing has been effective without sacrificing performance. However, in other cases like AC1, AC7, HC1, and HC7, the accuracy has decreased by at least 21.4 percent. Notably, in HC2, Prejudice Remover also experiences a loss in F1 score while mitigating bias.Since in-processing techniques introduce a new model and disregard the predictions of the original model, they do not necessarily yield better performance in all situations (datasets and tasks). In our assessment, Adversarial Debiasing exhibits good performance with the German Credit dataset but not with the Adult Census or Home Credit datasets. Therefore, in-processing techniques may not be suitable when altering the original modeling is not possible. Moreover, given the uncertainty in retaining performance, developers should exercise caution regarding prediction accuracy after intervention.

Finding 9: Although post-processing algorithms are the most dominating in debiasing, they are always diminishing the model accuracy and F1 score.

From Table 2, we can see that in 21 out of 40 models, one of the post-processing algorithms are being the most successful. But in all of the cases they are losing performance. The average accuracy reduction in these models is 7.49 percent and average F1 decrease is 10.07 percent. For example, in AC1, the most bias

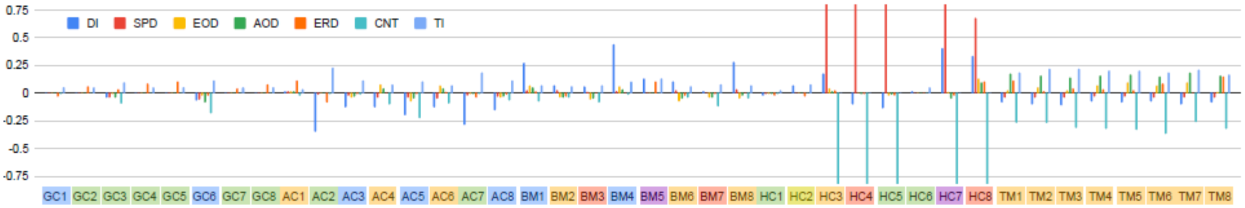


Figure 4.1: The fairness exhibited by the models after applying the bias mitigation techniques. The color coding in Table 2 is used to denote the most successful mitigation algorithm for each model.

mitigating algorithm is Reject option classification but the model is loosing 26.1 percent accuracy and 40 percent F1 score. In these cases, developers should move to the next best mitigation algorithm. In a few other cases such as HC8, the Reject Option classification mitigates bias with only 1.6 percent loss in accuracy and 1.3 percent loss in f1 score. In such situations, post-processing techniques can be applied to mitigate the bias.

4.2 Threats to Validity

Benchmark Creation: To avoid experimenting on low-quality kernels, we exclusively considered kernels with more than 5 votes. Furthermore, we excluded kernels where the model accuracy was very low (less than 65 percent). The top-voted kernels were then selected. We ensured the collected kernels were runnable. To guarantee the suitability of models for a fairness study, we initially selected fairness analysis datasets from previous works and searched for models for those datasets. Finally, we searched for competitions that used datasets with protected attributes mentioned in the literature.

Fairness and Performance Evaluation: Our collected models exhibit the same performance as reported in the corresponding Kaggle kernels. To evaluate fairness and apply mitigation algorithms, we utilized the AIF 360 toolkit developed by IBM. Bellamy et al. presented fairness results (4 metrics) for two models (Logistic Regression and Random Forest) on the Adult Census dataset with the protected attribute "race." We conducted experiments using the same setup and validated our results. Similar to their approach, we evaluated each metric 10 times and computed the mean values.

Fairness Comparison: As different metrics are computed based on different definitions of fairness, we have compared bias using a specific metric or cumulatively. Finally, in this project, we have focused on comparing fairness of different models. Therefore, for each dataset, we followed the same method to pre-process training and testing data.

Chapter 5

Conclusion

Conclusion

In this Project, ML fairness has received much attention recently. However, ML libraries do not provide enough support to address the issue in practice. I have empirically evaluated the fairness of ML models and discussed our findings of software engineering aspects. Initially, we established a benchmark comprising 40 ML models across 5 diverse problem domains. Subsequently, we employed a comprehensive set of fairness metrics to assess fairness. Following this, we applied 7 mitigation techniques to the models and recalculated the fairness metrics. Additionally, we evaluated the impact on model performance after applying mitigation techniques. Our findings shed light on prevalent types of bias and effective ways to address them. The study underscores the need for further Software Engineering (SE) research and enhancements to libraries to make fairness considerations more accessible to developers.

5.1 Bibliography

(1) `https://paperswithcode.com/paper/do-the-machine-learning-models-on-a-crowd`

(2) Toon Calders and Sicco Verwer. 2010. Three naive Bayes approaches for discrimination-free classification. Data Mining and Knowledge Discovery 21,2 (2010), 277–292.

(3) Kaggle. 2010. The world's largest data science community with powerful tools and resources to help you achieve your data science goals. www.kaggle.com.

[4] Kaggle. 2017. Adult Census Dataset. https://www.kaggle.com/uciml/adultcensus- income.

[5] Kaggle. 2017. Bank Marketing Dataset. https://www.kaggle.com/c/bankmarketing- uci.

[6] Kaggle. 2017. Competition: Santander Product Recommendation. https://www. kaggle.com/c/santander-product-recommendation/overview.

[7] Kaggle. 2017. German Credit Dataset. https://www.kaggle.com/uciml/germancredit.

[8] Kaggle. 2017. Home Credit Dataset. https://www.kaggle.com/c/home-creditdefault- risk.

[9] Kaggle. 2017. Titanic ML Dataset. https://www.kaggle.com/c/titanic/data.

[10] Kaggle. 2019. Adult Census Kernel: Multiple ML Techniques and Analysis. https://www.kaggle.com/bananuhbeatdo

ml-techniques-andanalysis- of-dataset.

[11] Kaggle. 2019. Kernel: German Credit Risk Analysis. https://www.kaggle.com/ pahulpreet/german-credit-risk-analysis-beginner-s-guide.