Final Presentation

German Credit Risk Classification

Software Development for DSAI, 2022

Introduction

Dataset Detail

The dataset contains 1000 entries with 20 categorical attributes

Each entry represents a person who takes a credit by a bank

Each person is classified as good or bad credit risks according to the set of attributes

Abstract: This dataset classifies people described by a set of attributes as good or bad credit risks. Comes in two formats (one all numeric). Also comes with a cost matrix

Data Set Characteristics:	Multivariate	**Number of Instances:**	1000	**Area:**	Financial
Attribute Characteristics:	Categorical, Integer	**Number of Attributes:**	20	**Date Donated**	1994-11-17
Associated Tasks:	Classification	**Missing Values?**	N/A	**Number of Web Hits:**	843242

Source:

Professor Dr. Hans Hofmann
Institut f"ur Statistik und "Okonometrie
Universit"at Hamburg
FB Wirtschaftswissenschaften
Von-Melle-Park 5
2000 Hamburg 13

Milestone 1

Milestone #1

- Data characteristics
- Data quality issues found
- Defined metrics
- The goals of the project

Data Characteristics

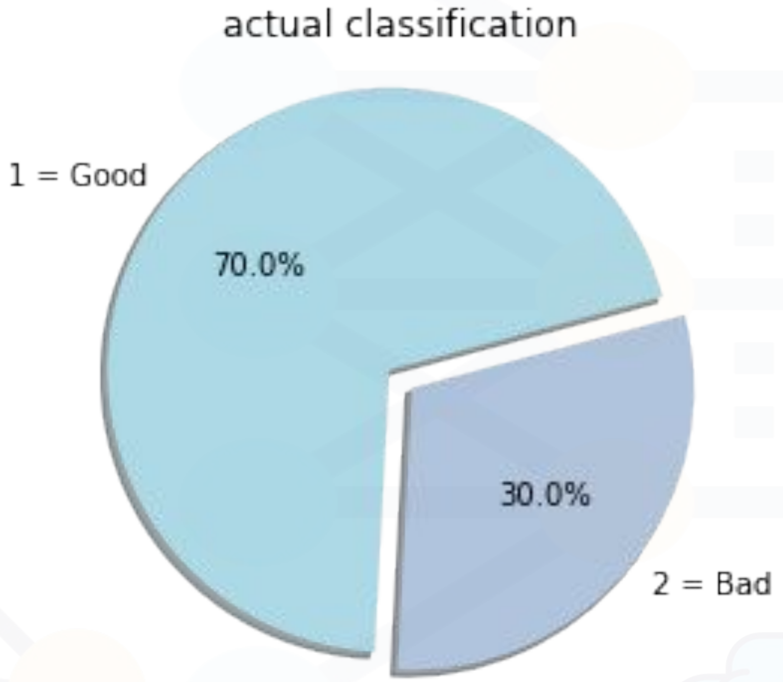
1. **Consistency and Uniqueness**
   Categorical features recorded with same format (AXXX)

2. **Timeliness**
   Dataset is out of date .

3. **Completeness**
   All relevant data were recorded. But including both categorical and numerical (multivariate) data make it difficult to select features to train the model and the class feature is imbalanced.

4. **Reliability**
   The original dataset came from UCI (the trusted source)

Visualizing the imbalanced class



Data Quality Issues

- **Bias**
  - Imbalanced class could caused the prediction bias
- **Performance**
  - Overfitting, high accuracy, but low precision and recall
- **Fairness**
  - Prediction rely on specific people group (specific features)
- **Reliability**
  - Data’s bias and unfairness caused the prediction unreliable

Metrics to measure quality issues

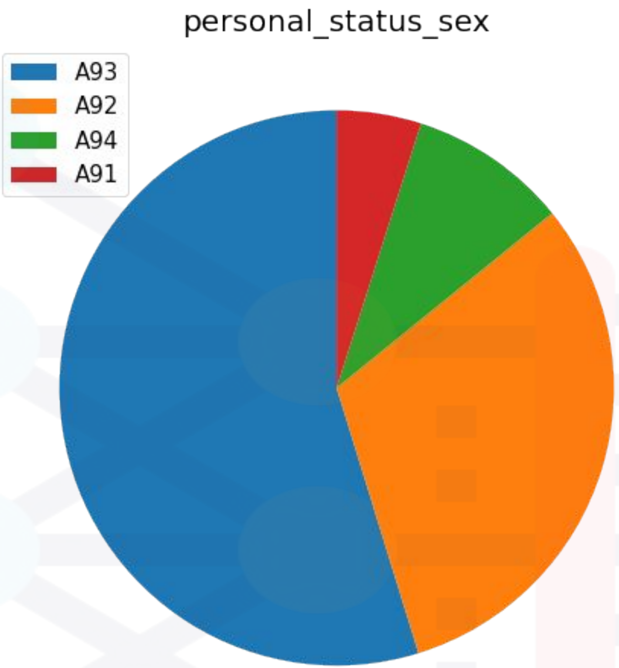
Performance

Accuracy, precision, recall and F1 score for measuring the performance the performance of the classifiers

Bias and Fairness

➔ Unbalanced in terms of **“sensitive attributes”** (age,sex)
➔ The issue with an unbalanced dataset is that model parameters can become skewed towards the majority
➔ **“Unbalanced dataset”** could be one of the reasons that the model is biased
➔ **“Bias”** as any error that has led the model to become unfair

According to the graph above, there are different age range candidates who applied for the loan. The majority are among the age group around 20-45 years.



In the pirechart for personal_status_sex attribute,there are different types of status but the majority are A93(male:Single)

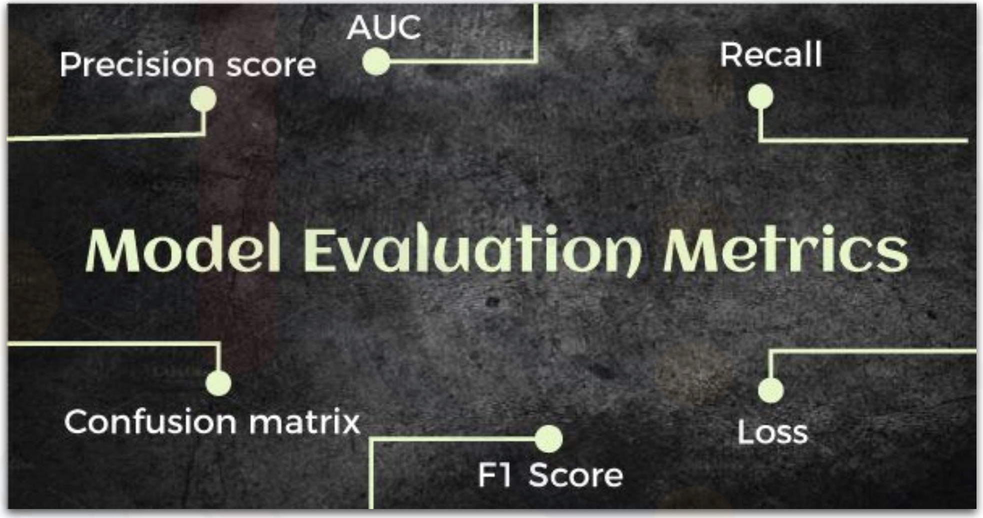
For the measuring the classification fairness, we are using the following metrics :

- **Average odd difference (AOD)** : average of difference in false positive rates and true positive rates between unprivileged and privileged groups. This metric **must be close to zero (0)** to ensure classification fairness.
- **Equal opportunity difference (EOD)** : difference in true positive rates between unprivileged and privileged groups. **Value of zero (0) implies the classification fairness**

Metrics to measure quality issues

Reliability

Data's bias and unfairness caused the prediction unreliable



We will use “Precision” to measure the reliability

We define

1=Good → Positive Class

2=Bad → Negative Class

It is worse to class a customer as good when they are bad. So we will focus on Precision(Checking precision values if the models have false positives)

False positive (FP): This means that the **prediction was positive class** and the **actual class was negative**.

Define the goals

1.Improve the prediction performance

Measures:

accuracy, precision, recall, and f1-score

confusion matrix

Processes :

Create a baseline prediction model using default parameter and features

Create improved version of prediction models

Compare those 4 measure with the baseline

2.Reduce the prediction bias (in term of imbalance class)

Measures:

precision, recall, and f1-score

Processes :

EDA on dataset

Apply data sampling techniques (over, under, combined)

Create models using balanced dataset and compare with baseline model

Milestone 2

Milestone #2

- Model selection
- Model comparison
    - Performance
    - Bias reduction

Model selection

We use 6 models from 4 developers follow by:

1. **Logistic Regression** - This is one of the supervised learning machine by Statistical Regression
2. **GaussianNaiveBayes** - This is a probabilistic classification algorithm based on applying Bayes' theorem
3. **Support Vector Machine** - This is supervised learning models with associated learning algorithms that analyze data for classification and regression analysis.
4. **Random Forest** - This is an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees
5. **Extreme Gradient Boosting** - Implementation of the stochastic gradient boosting ensemble algorithm for classification and regression problems in machine learning competitions.
6. **Ridge Regression** - Ridge regression is a method of estimating the coefficients of multiple-regression models in scenarios where the independent variables are highly correlated.

Model comparison results

Classifiers	Accuracy (%)	Training time(s)	Training Memory used (MB)	Testing time(s)	Testing Memory used (MB)	Model size (MB) (.pkl format)	Developer
XGBoosting	76.88	0.13	0.40	**0.01**	~ 0	0.20	Frong
LinearSVM	77.39	**0.03**	0.70	**0.01**	~ 0	**0.01**	Punch
Random Forest	76.88	0.2	1.68	0.02	~ 0	2.67	
GaussianNaive Bayes	73.4	0.043	**0.031**	0.033	0.211	**0.01**	Wendy
Linear Logistic Regression	**77.9**	0.050	0.242	0.050	0.020	**0.01**	
SMOTEENN with StandardScaler and RidgeClassifier	74.1	0.33	1.72	0.02	~ 0	2.89	Jincheng Zhang

Confusion Matrix

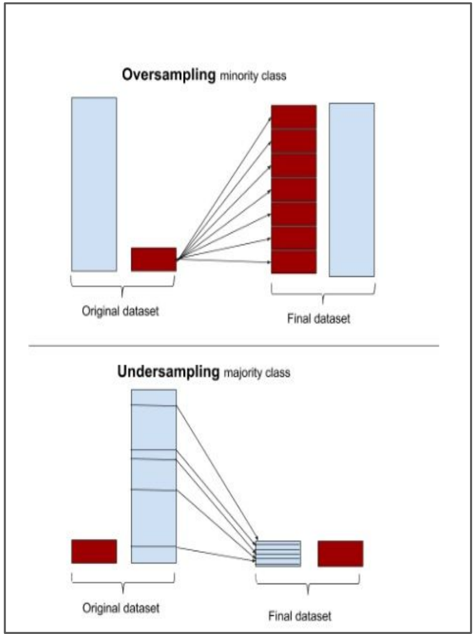
	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	True Positives (TPs)	False Positives (FPs)
Predicted Negative (0)	False Negatives (FNs)	True Negatives (TNs)

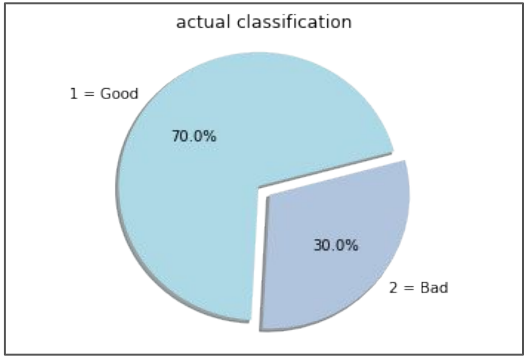
Precision = TPs / (TPs + FPs)

Recall = TPs/(TPs+FNs)

Accuracy = (TPs + TNs) / (TPs+TNs+FPs + FNs)

F1= 2 x (Precision x Recall)/(Precision + Recall)





"It is worse to class a customer as good when they are bad (5), than it is to class a customer as bad when they are good (1)." -(UCI Machine Learning Repository: Statlog (German Credit Data) Data Set)

Focus on amount of FNs will be minimize because model must not predict bad as good.

Model performance comparison after applied sampling techniques

Model	Sampling method	Accuracy (%)	Precision (%)	**Recall (%)**	**F1-score (%)**
XGBoosting	Under-sampling	67.84	71.74	67.84	68.99
	Over-sampling	73.37	73.77	73.37	73.55
	Non-sampling	**76.88**	**75.97**	**76.88**	**76.21**
LinearSVM	Under-sampling	72.36	75.47	72.36	73.27
	Over-sampling	72.86	74.29	75.38	73.93
	Non-sampling	**77.39**	**76.72**	**77.39**	**76.95**
Random Forest	Under-sampling	69.35	74.25	69.35	70.57
	Over-sampling	75.38	74.29	75.38	74.58
	Non-sampling	**76.88**	**75.67**	**76.88**	**74.85**
KNNs	Under-sampling	72.4	72.8	72.4	72.6
	Over-sampling	**76.4**	**80.6**	**76.4**	**77.3**
	Non-sampling	75.9	74.4	75.9	73.5
Ridge Classifier	**Under-sampling**	**74.16**	**75.12**	**77.21**	**76.95**
	Over-sampling	75.27	73.98	75.44	73.94
	Non-sampling	75.98	73.27	74.29	75.66

Bias Mitigation

Bias can be injected to the system in three stages :

- Pre-processing : dataset may be biased
- In-processing: algorithm may be biased
- Post-processing: test dataset may be biased

For mitigate biases of the classification , we select 2 bias mitigation techniques :

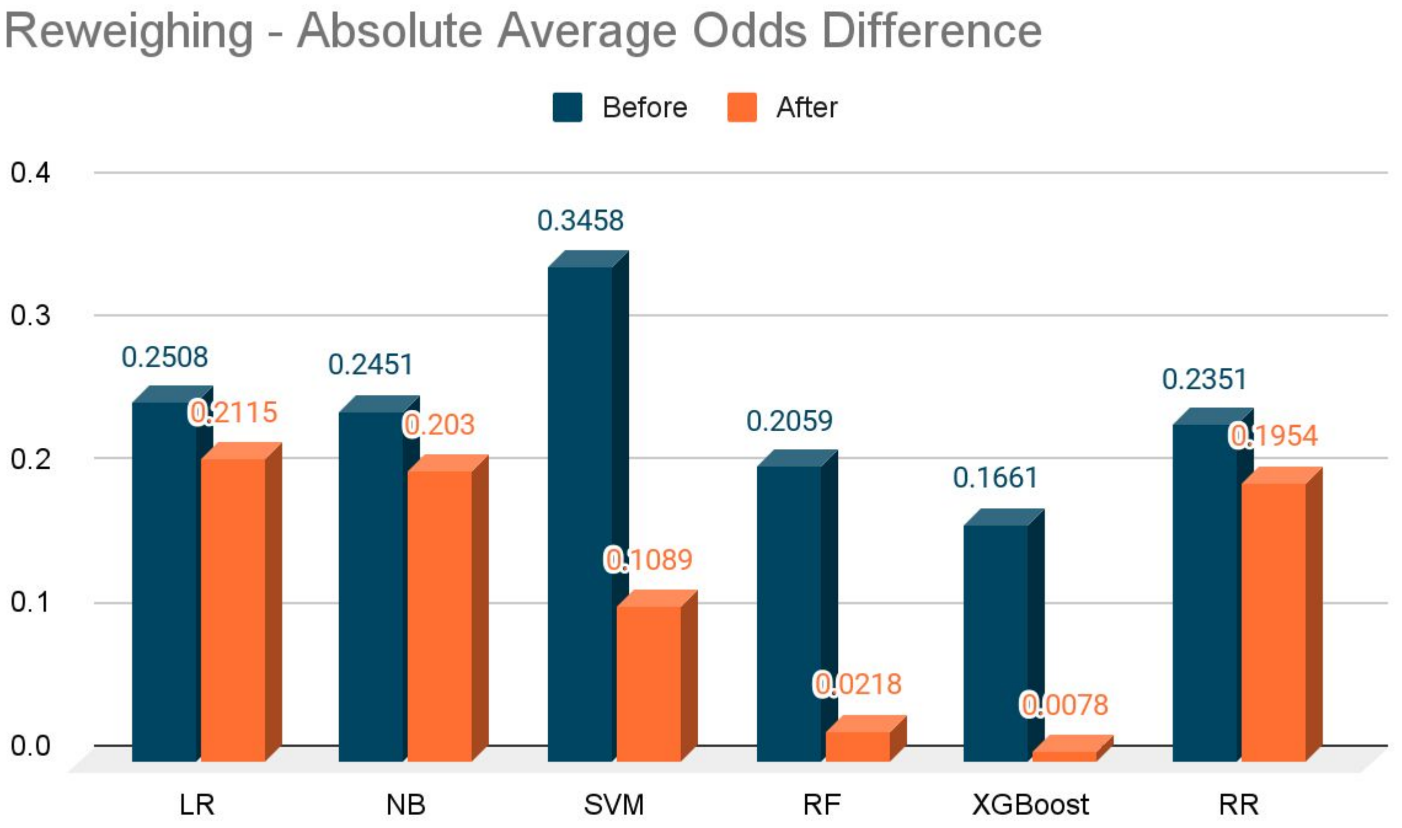
- Reweighing
- Reject option classification

Reweighing

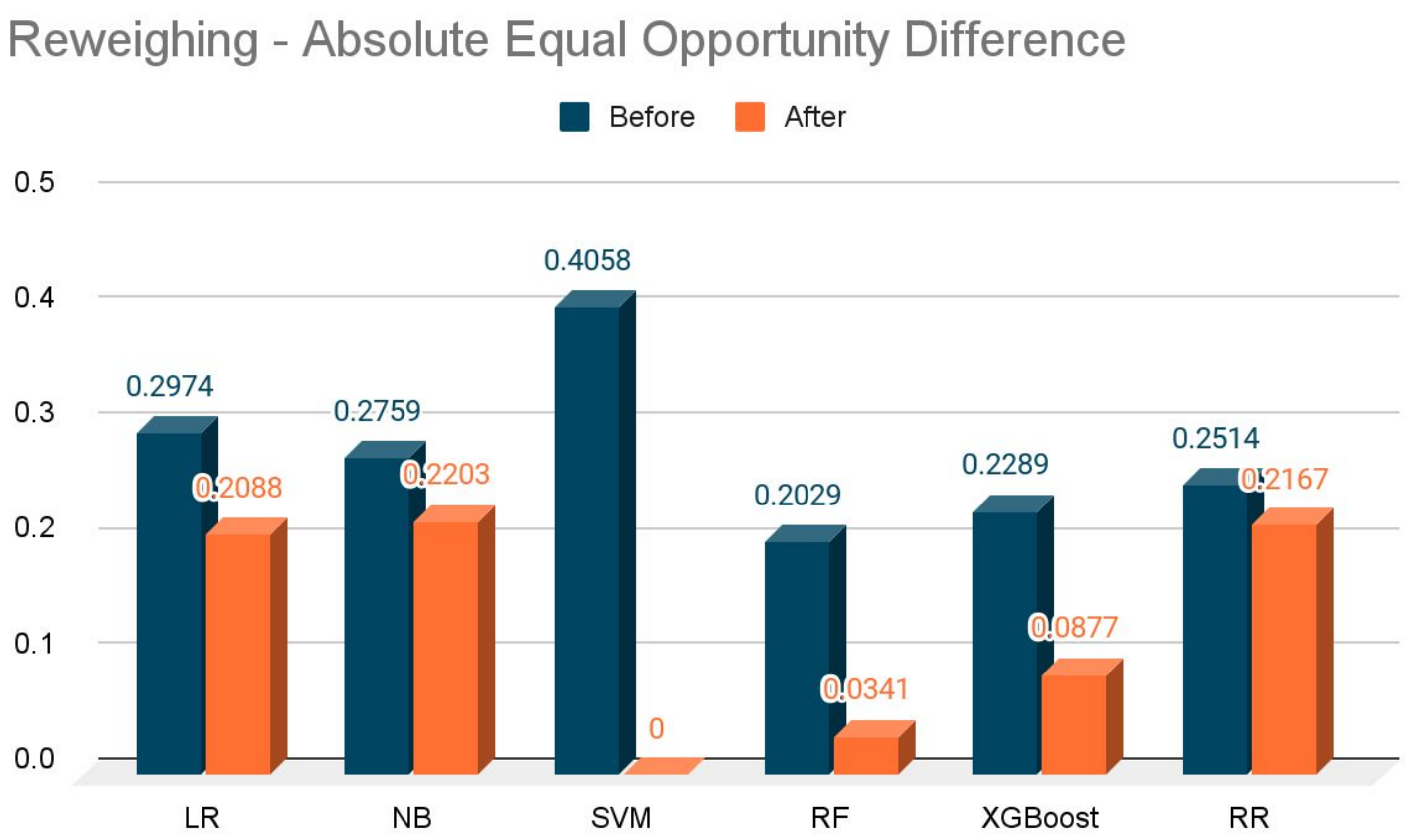
- Pre-processing algorithm : applied to training data
- Making modifications on the training data
- Compute and apply set of weights

```
RW = Reweighing(unprivileged_groups=unprivilege_groups, privileged_groups=privileged_groups)

trans_train_set = RW.fit_transform(og_train_set)
```



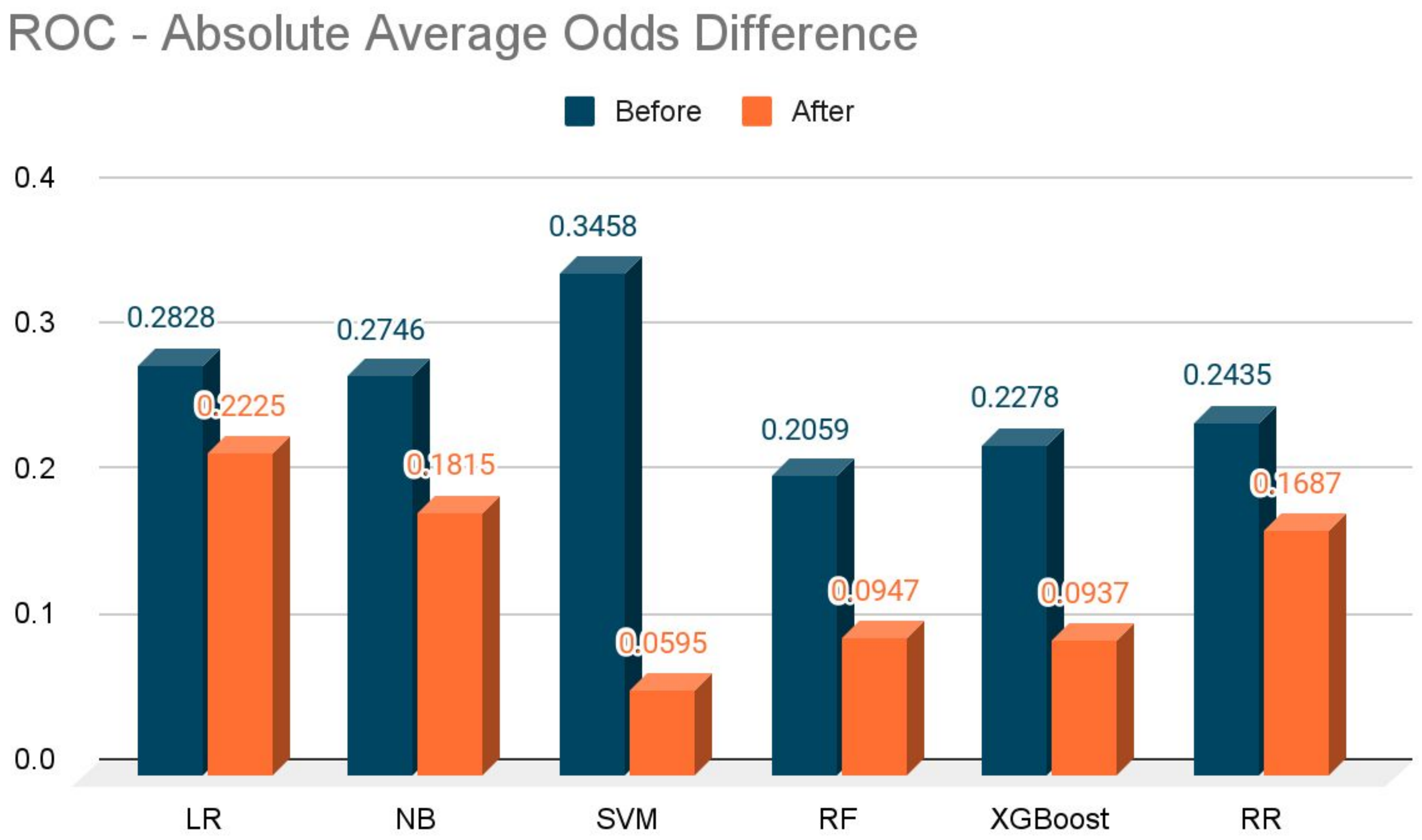
Reweighing - Absolute Average Odds Difference
Before
After
0.4
0.3
0.2
0.1
0.0
0.2508
0.2115
0.2451
0.203
0.3458
0.1089
0.2059
0.0218
0.1661
0.0078
0.2351
0.1954
LR
NB
SVM
RF
XGBoost
RR

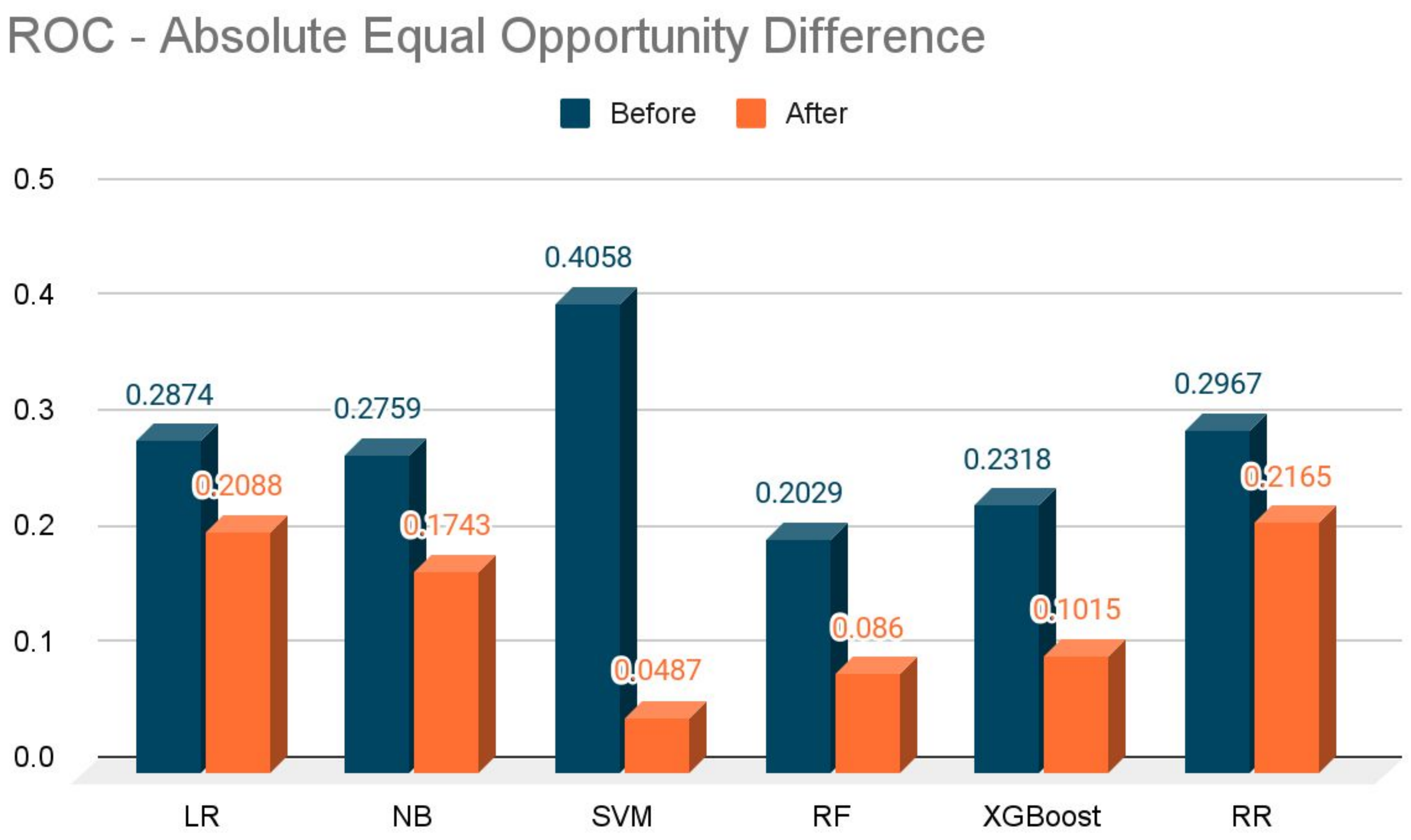

Reweighing - Absolute Equal Opportunity Difference
Before
After
0.5
0.4
0.3
0.2
0.1
0.0
0.2974
0.2088
0.2759
0.2203
0.4058
0
0.2029
0.0341
0.2289
0.0877
0.2514
0.2167
LR
NB
SVM
RF
XGBoost
RR

Reject option classification

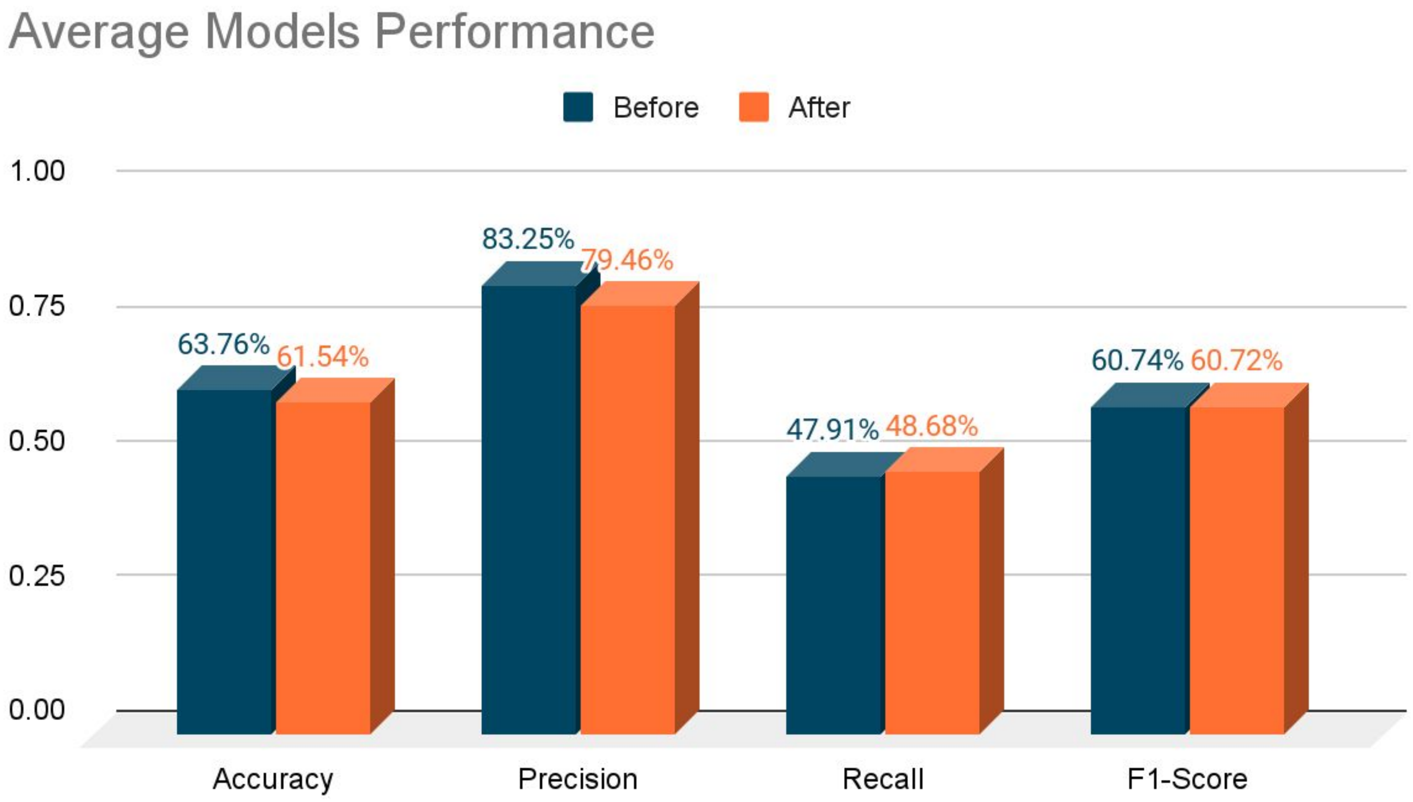
- Post-processing algorithm : applied to testing data
- Concern on the least certain of the prediction where most discrimination occurs i.e. around the decision boundary (classification threshold)
- Exploit the low confidence region and reject the predictions

```
ROC = RejectOptionClassification(unprivileged_groups=unprivilege_groups,
                                 privileged_groups=privileged_groups,
                                 low_class_thresh=0.01, high_class_thresh=0.99,
                                 num_class_thresh=100, num_ROC_margin=50,
                                 metric_name="Statistical parity difference",metric_ub=0.05, metric_lb=-0.05)
ROC = ROC.fit(og_valid_set, og_valid_set_pred)
```


ROC - Absolute Average Odds Difference
Before
After
0.4
0.3
0.2
0.1
0.0
0.2828
0.2225
0.2746
0.1815
0.3458
0.0595
0.2059
0.0947
0.2278
0.0937
0.2435
0.1687
LR
NB
SVM
RF
XGBoost
RR



ROC - Absolute Equal Opportunity Difference
Before
After
0.5
0.4
0.3
0.2
0.1
0.0
0.2874
0.2088
0.2759
0.1743
0.4058
0.0487
0.2029
0.086
0.2318
0.1015
0.2967
0.2165
LR
NB
SVM
RF
XGBoost
RR


Average Models Performance
Before
After
1.00
0.75
0.50
0.25
0.00
63.76%
61.54%
83.25%
79.46%
47.91%
48.68%
60.74%
60.72%
Accuracy
Precision
Recall
F1-Score

Models		Reweighting				Reject option classification			
		Accuracy	Precision	Recall	F1-score	Accuracy	Precision	Recall	F1-score
Linear Regression	Before	0.6042	0.7846	0.5000	0.6108	0.6975	0.9077	0.5413	0.6782
	After	0.6085	0.7826	0.5294	0.6316	0.6685	0.8788	0.5321	0.6629
Gaussian Naive Bayes	Before	0.6048	0.7903	0.4804	0.5976	0.7021	0.9091	0.5505	0.6857
	After	0.6097	0.7937	0.4902	0.6061	0.6425	0.8594	0.5046	0.6358
SVM	Before	0.6066	0.8113	0.4216	0.5548	0.6066	0.8113	0.4216	0.5548
	After	0.6232	0.8511	0.3922	0.5369	0.5980	0.7714	0.5294	0.6279
Random Forest	Before	0.6158	0.8103	0.4608	0.5875	0.6158	0.8103	0.4608	0.5875
	After	0.6097	0.7937	0.4902	0.6061	0.6011	0.8000	0.4314	0.5605
XGBoost	Before	0.5913	0.7925	0.4118	0.5419	0.6761	0.8906	0.5229	0.6590
	After	0.5619	0.7660	0.3529	0.4832	0.6190	0.7882	0.6381	0.7053
Ridge Regression	Before	0.6891	0.7962	0.4839	0.6034	0.6423	0.8761	0.4937	0.6285
	After	0.6132	0.7973	0.4768	0.6184	0.6295	0.8538	0.4752	0.6121

Noted: the accuracy might be different from the results from previous section due to different trained feature and data preparation process.

Milestone 3

Milestone #3

- Result analysis
- Possibility to improve classifiers’ quality
- Solutions & best practices to mitigate quality issues

Results Analysis

- No significant differences between the models
    - trained by using all attributes : the average accuracy is ~75%
    - trained by using “age” as a protected attribute: the average accuracy is ~63%
- No significant differences among the data sampling methods
    - The f1-score archived around 68 - 77%
- Bias mitigation techniques could effectively work on some models such as SVM, Random forest, and XGBoost

Improvement possibility

- Bias reduction
    - Reduce from the first stage as data collection
- Feature collection and selection
- Experimenting various types of machine learning algorithms

Solutions and Best practices

Experienced perspective :

- Always monitor and check for bias and anomalies of data
- Collaboration between customer and data scientist
- Pay attention to data privacy and protected attributes
- Avoid black-box
- Having clear system’s goal(s) and measurable metrics

Solutions and Best practices

Theoretical perspective :

- Make the intelligent system to be generalized
- Make sure the system product the right type of mistake
- Gathering and using right amount of data
- Setting the right measurement metrics

Thank you

Q & A

References

- https://github.com/Trusted-AI/AIF360/tree/master/examples
- https://www.mathworks.com/help/risk/explore-fairness-metrics-for-credit-scoring-model.html
- https://medium.com/sfu-cspmp/model-transparency-fairness-552a747b444
- https://aif360.readthedocs.io/en/latest/?badge=latest
- https://medium.com/analytics-vidhya/machine-learning-is-requirements-engineering-8957aee55ef4
- UCI Machine Learning Repository: Statlog (German Credit Data) Data Set