University of Calabria

Department of Mathematics and Computer Science

Data Analytics (Machine Learning) project

Authors
Mahbub Hasan [232759]
Renier Cristino Perez Gallardo [232693]
Gebreyowhans Hailekiros Bahre [233619]
Tesfay Gebremeskel Chekole [231856]

Respective Faculty
Prof.P. Rullo
Angelica Liguori

June 23, 2022

Contents

List of Figures

List of Tables

1 Business Understanding

Correctional Offender Management Profiling for Alternative Sanctions (COMPAS) is a case management and decision support tool developed and owned by Northpointe used by U.S. courts to assess the likelihood of a defendant becoming a recidivist. The COMPAS tool assigns defendants scores from 1 to 10 that indicate how likely they are to reoffend based on 47 factors, including age, sex and criminal history. These scores profoundly affect defendants' lives: defendants who are defined as medium or high risk, with scores of 5-10, are more likely to be detained while awaiting trial than are low-risk defendants, with scores of 1-4.

1.1 Objective

The goal is to understand whether a defendant had reoffended after the arrest or not.

1.2 Data Mining Objectives

The objective is to train a model capable of discerning whether a defendant will reoffend after arrest or not.

2 Data Understanding

The first step of this project is to understand the data. In this section we addressed the activities of data understanding. We started with data collection, selection up to balance.

2.1 Data Collection

The first and preliminary step of data understanding is to collect the data from valid source. As we mentioned in our goal, we collected our data from the github repository[1]. In the above repository, we got plenty of dataset but according to our goal, we used "compas_score.csv" file.

2.2 Data Description

The dataset has the following characteristics:

- Multivariate
- 11,757 rows representing the number of records
- 47 columns representing the number of attributes in the dataset
- 552,579 total total data

Out of the 47 attributes, we initially have the following division of data-types:

- 12 Datetime attributes
- 16 numeric attributes
- 19 nominal attributes

2.3 Attribute Description

As we mentioned earlier we have 47 attribute and each attribute holds either numerical or categorical information. The following table-2.1 describe almost each attribute information and their description.

Name	Data Type	Null Values	Description
id	int64	0	Unique number of the dataset
Name	Object	0	Name of the person
first	Object	0	First name of the person
last	Object	0	Last name of the person
compas_screening_date	datetime64	0	This attribute indicate a date, when a person's (comitted crime) information enter into the COMPAS system
sex	Object	0	This attribute indicate a person's gender; Male or Female
dob	datetime64	0	This attribute indicate a person's date of birth
age	int64	0	This attribute indicate a person's age
age_cat	Object	0	This attribute indicate a person's age in category; less_than_25, 25-45, and grater_than_45
decile_score	int64	0	This attribute indicate COMPAS score between 1 to 10
decile_score	int64	0	This attribute indicate COMPAS score between 1 to 10
days_b_screening_arrest	float64	0	This attribute indicate a difference between crime date and compas date
c_offense_date	datetime64	2600	This attribute indicate a person's crime date
c_arrest_date	datetime64	9899	This attribute indicate a person's arrest date after occurring a crime
is_recid	int64	0	This attribute indicate commit new offenses after he/she has committed a crime in the past
r_offense_date	datetime64	8054	This attribute indicate a person's regular offence date
is_violent_recid	int64	0	This attribute indicate a person is committing a violent/serious crime or not
vr_offense_date	datetime64	10875	This attribute indicate in which date a person is committed serious/violent crime
v_decile_score	int64	0	This attribute indicate a score of a person who committed a serious/violent crime
score_text	Object	15	This attribute indicate the priority of crime according to the declin_score ; low: 1-4, medium: 5-6, and high: 7-10

Table 2.1: Attribute Description

2.4 Data Exploration

2.4.1 Missing Data

Analyzing the occurrence of missing values we see that when the value is missing we find a "NaN" or "NaT" values. Furthermore, we summarized the number of missing values and their percentage from the total for each of the attributes in the table-2.1 below. From the table 2.1 we can observe that the attributes with the highest number of missing values are "num_r_cases","num_vr_cases","c_arrest_date","r_days_from_arrest", "r_offense_date","r_charge_desc","r_jail_in","r_jail_out","vr_case_number","vr_charge_degree", "vr_offense_date" and "vr_charge_desc".

Attribute	Number of missing values	percentage Missing values
days_b_screening_arrest	1180	10%
c_jail_in	1180	10%
c_jail_out	1180	10%
c_case_number	742	6.3%
c_arrest_date	9899	84%
c_days_from_compas	742	6.3%
c_charge_desc	749	6.3%
num_r_cases	11757	100%
r_days_from_arrest	9297	79%
r_offense_date	8054	68%
r_charge_desc	8114	69%
r_jail_in	9297	79%
r_jail_out	9297	79%
num_vr_cases	11757	100%
vr_case_number	10875	92.4%
vr_charge_degree	10875	92.4%
vr_offense_date	10875	92.4%
vr_charge_desc	10875	92.4%
c_offense_date	2600	22%
r_case_number	8054	68.5%

Figure 2.1: Missing values

2.4.2 Balance

With respect to the target class label i.e **"is_recid"** attribute,we tried to check whether the dataset is balanced or not. From the initial analysis of the data set the target class label was distributed as in the below three different values:

- 719 records with target class label of -1
- 7335 records with target class label of 0

- 3703 records with target class label of 1

But despite the importance of balance, in our case balancing decreases the accuracy of our models, which is why we decided not to apply it.

3 Data Cleaning

After data understanding, the most important part is data cleaning. In this section we removed mostly missing columns, some unwanted features and duplicate values.

3.1 IDs

In our "compas_score" dataset, "id" attribute is most unique column. It contains 11,757 unique values and is base attribute of others. We used this attribute to check and compare with others.

3.2 Remove Missing Values

	count	mean	std	min	25%	50%	75%	max
id	11757.0	5879.000000	3394.097892	1.0	2940.0	5879.0	8818.0	11757.0
age	11757.0	35.143319	12.022894	18.0	25.0	32.0	43.0	96.0
juv_fel_count	11757.0	0.061580	0.445328	0.0	0.0	0.0	0.0	20.0
decile_score	11757.0	4.371268	2.877598	-1.0	2.0	4.0	7.0	10.0
juv_misd_count	11757.0	0.076040	0.449757	0.0	0.0	0.0	0.0	13.0
juv_other_count	11757.0	0.093561	0.472003	0.0	0.0	0.0	0.0	17.0
priors_count	11757.0	3.082164	4.687410	0.0	0.0	1.0	4.0	43.0
days_b_screening_arrest	10577.0	-0.878037	72.889298	-597.0	-1.0	-1.0	-1.0	1057.0
c_days_from_compas	11015.0	63.587653	341.899711	0.0	1.0	1.0	2.0	9485.0
is_recid	11757.0	0.253806	0.558324	-1.0	0.0	0.0	1.0	1.0
num_r_cases	0.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
r_days_from_arrest	2460.0	20.410569	74.354840	-1.0	0.0	0.0	1.0	993.0
is_violent_recid	11757.0	0.075019	0.263433	0.0	0.0	0.0	0.0	1.0
num_vr_cases	0.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
v_decile_score	11757.0	3.571489	2.500479	-1.0	1.0	3.0	5.0	10.0
decile_score.1	11757.0	4.371268	2.877598	-1.0	2.0	4.0	7.0	10.0

Figure 3.1: Statistics of (Compas Score) dataset

According to the information of 2.1 and 3.1, the attribute "num_r_cases" has a high amount of missing values ($\approx 100\%$), therefore we removed this attributes for our analysis.

The option to fill the missing values was not reasonable because when the percentage of missing values increases, it also increases the possibility of introducing errors or bias by imputing. For the same reason, we removed the attribute "num_vr_cases", which has (≈ 90% more) of missing values.

3.3 Removing Negative Values

There are some attributes which are very important but having some unwanted values that we don't use them for our analysis. To take the analysis into account, we first need to remove these unwanted values(like any negative values). The figure -3.1 in the above indicates "decile_score", "is_recid", and "vr_decile_score" attributes has some negative values. We first removed those rows to make it reasonable for our analysis.

3.4 Removing out of range values

In the compas_score dataset; there are also some features related to date and from those we selected five date related attributes as in the below :

1. compas_screening_date
2. dob
3. days_b_screening_arrest
4. r_offense_date
5. vr_offense_date

According to the (ProPublica)[2], "to match COMPAS scores with accompanying cases, we considered cases with arrest dates or charge dates within 30 days of a COMPAS assessment being conducted. In some instances, we could not find any corresponding charges to COMPAS scores. We removed those cases from our analysis."

To hold the consistency and validity the analysis,we considered (<30) and (>-30) days values from the "days_b_screening_arrest" attribute. This data indicates, if any person whose information is at least twice or more between this date interval, we suspect that person is more likely to commit new offenses after he/she has committed a crime in the past.

3.5 Removing duplicate values

After removing the negative(unwanted) values, we have also to check for duplicate values. To check duplicate values, we took four basic attributes that we thought of they can uniquely identify a record namely ("name", "sex", "dob", and "race") to if there are rows consisting multiple times or not. If we found any information which consist multiple time we remove that row.

4 Data Preparation

After data cleaning step, the next step is data preparation. In this step, we check our selected attributes according to the target class and check correctness of the values.

4.1 Attribute Compas Screening Date

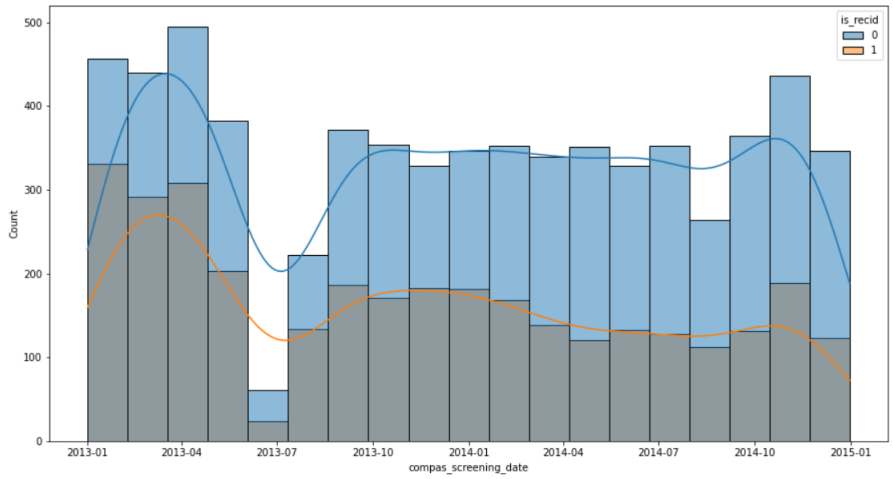

Figure 4.1: Plotting of Compas-screening-date and Target Class

The attribute "compas_screening_date" must be belong to 2 years information. The above figure-4.1 is about "compas_screening_date" and target class, where we can see all the information is within 2 years. So, we can take this attribute for further analysis.

4.2 Attribute sex

The attribute "sex" must belong to either male or female. The above figure-4.2 is about sex and target class, where we can see all the information is consisting either male or female. So, we can take this attribute for further analysis.

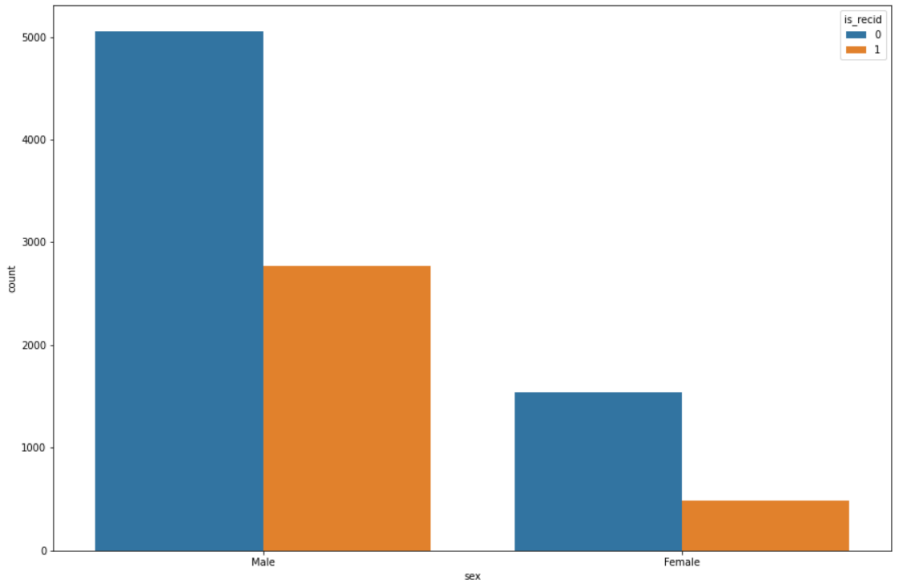

Figure 4.2: Plotting of Sex and Target Class

4.3 Attribute age

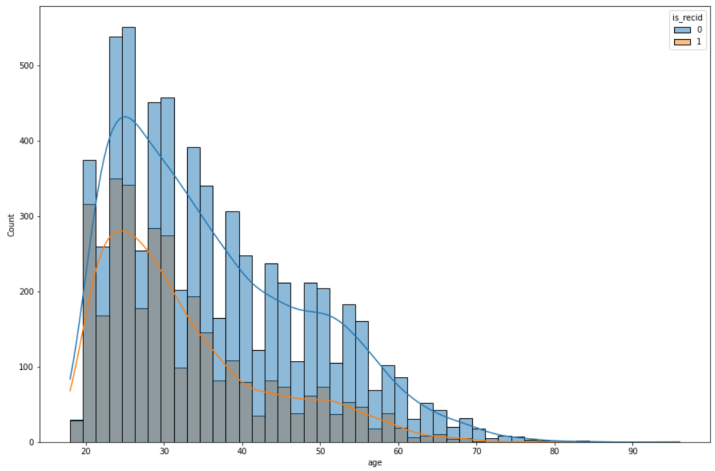

Figure 4.3: Plotting of Age and Target Class

The attribute “age” must belong to positive number. The above figure-4.3 is about age and target class, where we can see all the information is grater than zero. So, we can take this attribute for further analysis too.

4.4 Attribute race

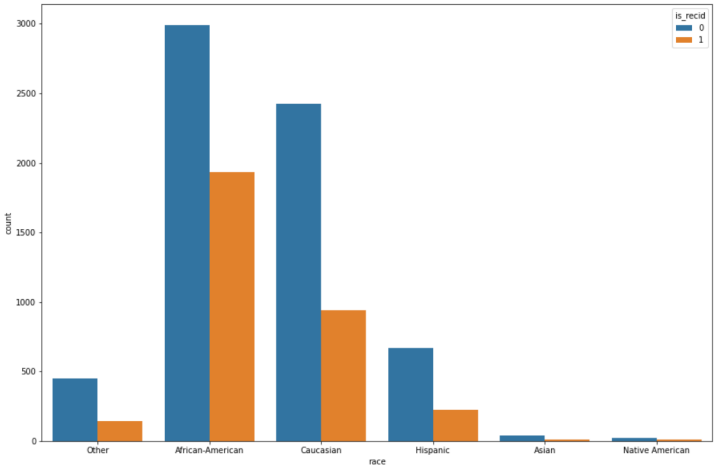

Figure 4.4: Plotting of Race and Target Class

The attribute "race" is of the most important attribute in our dataset. In our dataset,the race attribute contains 6 different values. The above figure-4.4 to is make a relation between race and target class. From the figure we can see all the information are not outside of that 6 different race. So, we take this attribute for our further analysis too.

4.5 Attribute decile score

This attribute "declie_score" is compas score. All values of this attribute ranges between 1 to 10. The above figure-4.5 indicates that, no information is our of range. So, we take this attribute for further analysis too.

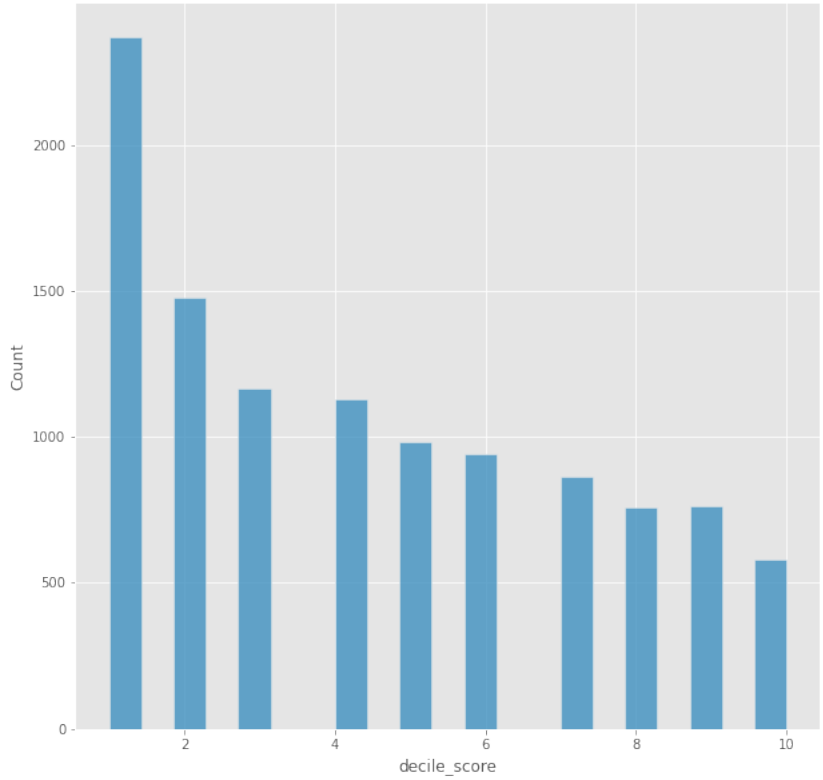

Figure 4.5: Plotting of Decile Score

4.6 Attribute Violent(V) decile score

"v_decile_score" attribute indicates a numeric values (between 1 to 10) of a person who re-offended after a previous crime. From the above figure-4.6, we can see all values are within the range 1 to 10 . So, we take this attribute for further analysis too.

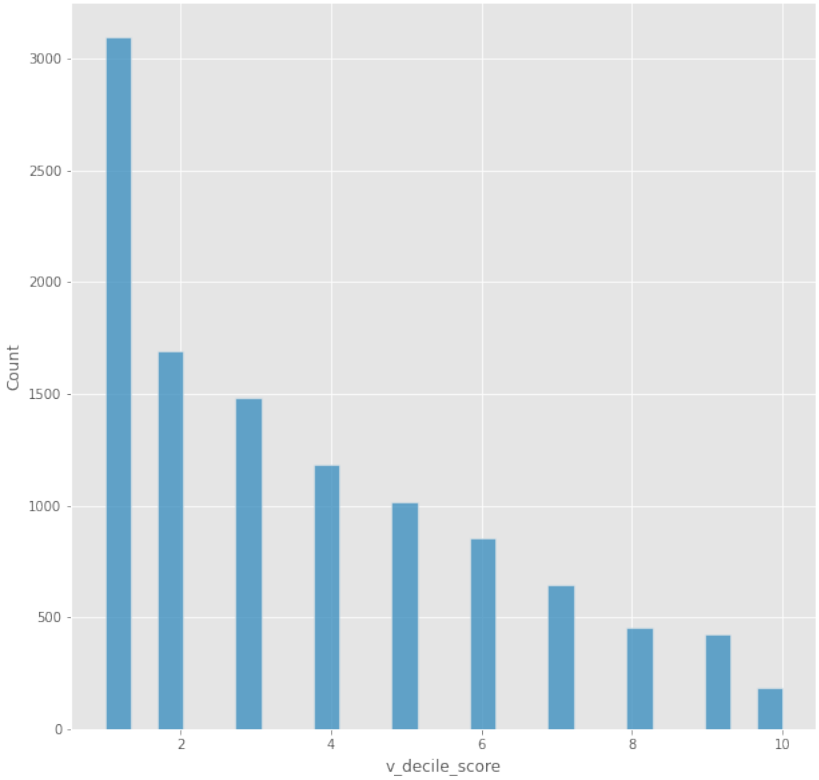

Figure 4.6: Plotting of Violent Decile Score

4.7 Attribute "Is_recid"

As we mentioned earlier, "is_recid" is our target attribute. This attribute holds only 2 values, 0 indicates a person is didn't re-offended and 1 means the reverse. According to the above figure-4.7, indicates that all values belongs to 0 or 1. So, there is no problem to take this attribute for our further analysis.

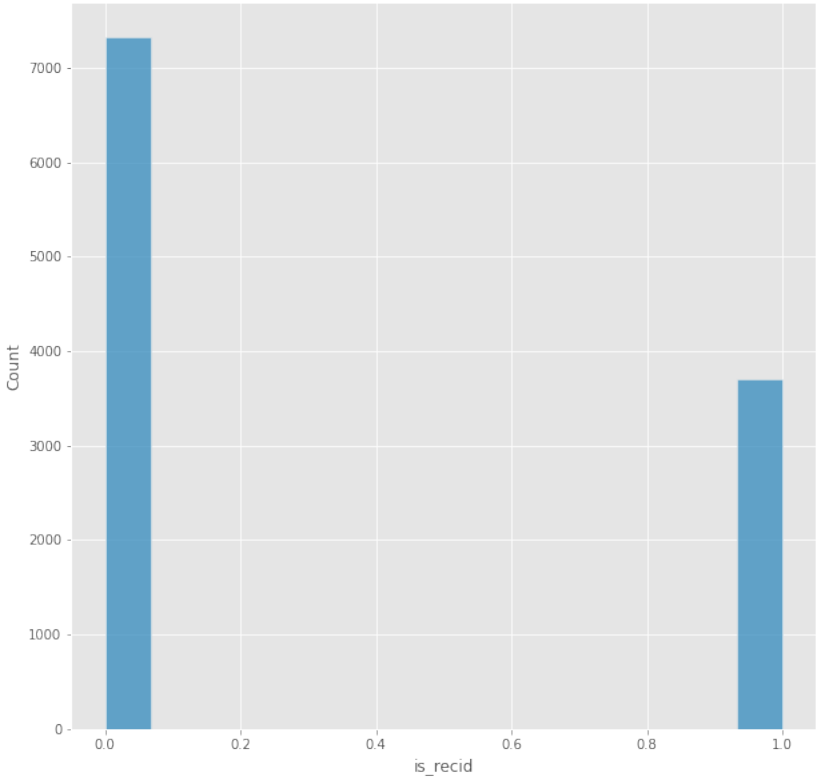

Figure 4.7: Plotting of is_recid Attribute

4.8 Level encoding

Level encoding means replace the categorical value with a numeric value between 0 and the number of classes minus 1. In our dataset, we have 2 attributes "sex" and "race" which is categorical data. So, we need to apply level encoding to make this categorical to numerical attribute because machine only understands 0/1. In this project, we apply two different approach for level encoding.

1. using scikit-learn library
2. using pandas library

4.8.1 Attribute "sex" level encoding

As we mentioned earlier, our attribute "sex" consists of categorical values (male and female). We covert these values into 0 and 1. 0 indicates female and 1 male. The figure-4.8 indicates the process of level encoding of "sex" attribute.

```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
```

```
sex_attrib = le.fit_transform(df['sex'])
df.drop(labels=['sex'], axis=1, inplace=True)
df['sex'] = sex_attrib
```

Figure 4.8: Attribute "sex" Level Encoding Process

4.8.2 Attribute "race" level encoding

To convert categorical to numerical for "race" attribute, we simply use pandas "get_dummy()" class. In above we applied scikit-learning level encoding process for "sex" attribute but for this we use different one. Because scikit-learn level encoding, encode class into (class-1) information. Our sex attribute holding only two information so that we can easily apply this.

But race consists of 6 different values. So, to hold the consistency of our data, we applied pandas encoding process. This figure-4.9 indicates the process of encoding for race attribute.

```
df = pd.get_dummies(data=df,columns=['race'])
```

Figure 4.9: Attribute "race" Encoding Process

After encoding, our dataset is looks like exactly of the following figure-4.10

race_African-American	race_Asian	race_Caucasian	race_Hispanic	race_Native American	race_Other
0	0	0	0	0	1
1	0	0	0	0	0
1	0	0	0	0	0
1	0	0	0	0	0
0	0	0	0	0	1
...	...	...	...	...	...
1	0	0	0	0	0
0	0	0	0	0	1
0	0	1	0	0	0
0	0	0	0	0	1
0	1	0	0	0	0

Figure 4.10: Dataset After Race Encoding

5 Modeling

5.1 Training and testing data

Before going to the modeling, we need to split our data into training and testing set. "**Training**" set is used to train our model to predict the result. In our project, we use 80% of data for training purpose. "**Testing**" set is used to predict the accuracy (of the result) after training the model. Basically, testing data is for test how accurate our model is. In our project, we use only 20% of data for testing purpose.

5.2 Model Selection

Model selection is one of the vital role in machine learning project. Choosing a good model is to increase the probability of prediction accuracy. Our target class is holding binary information.So we choose the binary classification model to predict our information:

1. Logistic Regression
2. Decision Trees
3. Naive Bayes
4. Ada-boost Classifier

5.3 Logistic Regression Model

Based on the model selection, our first model is logistic regression. The solvers We used is "saga" with "max_iter" of 1000000. Logistic regression by default uses "lbfgs" solver; but due to high number of attributes the default solver doesn't work very well. So we used "saga" instead of "lbfgs". After apply the logistic regression model, our result is:

Also, we calculated the four values (TP, TN, FP, and FN). The following figure-5.1 indicates the confusion matrix of the four values after applying the model.

Logistic Regression		
Accuracy: 0.71		
Metric	Score(0)	Score(1)
Score	0.81	0.40
Precision	0.72	0.68
Recall	0.93	0.28

Table 5.1: Logistic Regression Result

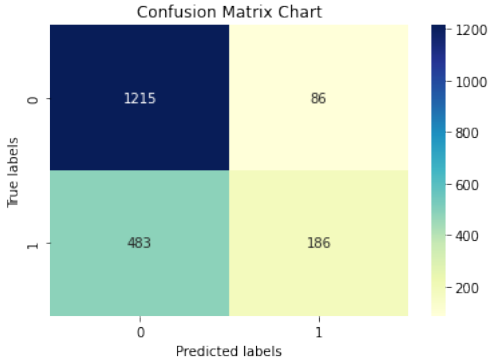

Figure 5.1: Confusion Matrix of logistic regression

5.4 Naive Bayes Model

The other classification model that we tried is the Naive Bayes model. The Naive Bayes classifier of "python's sklearn" library have three different classifier models namely "Gaussian","Multinomial" and "Bernoulli". After we applied these three Naive Bayes models and analyzing their models performance scores and also according the guidance we found in the "python's scikit-learn" documentation [3] we found that the "Bernoulli" model of Naive Bayes provides us better performance over the others as in the below table:

Naive Bayes (Bernoulli) model		
Accuracy: 0.67		
Metric	Score(0)	Score(1)
Score	0.79	0.28
Precision	0.69	0.58
Recall	0.93	0.18

Table 5.2: Naive Bayes Result

Also, we calculated the four values (TP, TN, FP, and FN). The following figure-5.2 indicates the confusion matrix of the four values after applying the model.

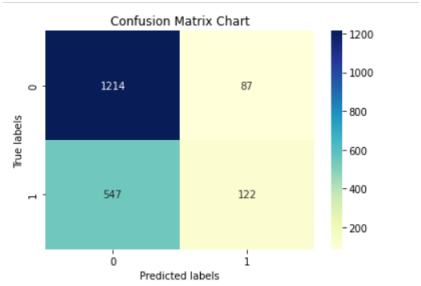

Figure 5.2: Confusion Matrix of Naive Bayes

From the confusion matrix, we see that the Naive Bayes classifier got the following results:

Out of the 1301 actual instances of '0', it predicted correctly 1214 of them; Out of the 669 actual instances of '1', it predicted correctly 122 of them. Note that the accuracy may be obtained from the confusion matrix, as the sum of the diagonal divided by the sum of all matrix entries:

5.5 Decision Tree

It is built iteratively by splitting the training examples by features, following a criterion (entropy and gini in our implementation). The results of the model trained on the dataset with entropy were the following:

Decision Tree (entropy)		
Accuracy: 0.61		
Metric	Score(0)	Score(1)
Score	0.70	0.44
Precision	0.71	0.43
Recall	0.69	0.45

Table 5.3: Decision Tree (Entropy) Result

The results of the model trained on the dataset with gini were the following:

Also, we calculated the four values (TP, TN, FP, and FN). The following figure-5.3 is indicate the confusion matrix after applying the model with entropy.

The following figure-5.4 is indicate the confusion matrix after applying the model with gini.

Decision Tree (gini)		
Accuracy: 0.49		
Metric	Score(0)	Score(1)
Score	0.58	0.37
Precision	0.65	0.32
Recall	0.52	0.45

Table 5.4: Decision Tree (Gini) Result

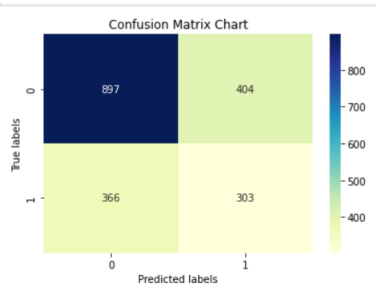

Figure 5.3: Confusion Matrix of decision tree (entropy)

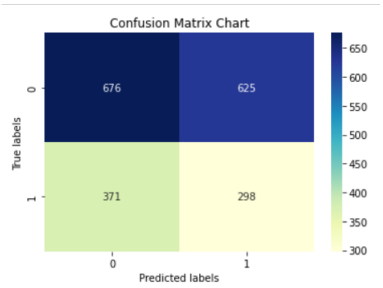

Figure 5.4: Confusion Matrix of decision tree (gini)

5.6 Ada-boost Classifier

The Ada-boost Classifier using decision tree as base classifier yields the following model performance results.

The confusion matrix indicating the four parameters(TP, TN, FP, and FN) of the ada-boost classifier is shown in the figure figure-5.5.

Ada-boost		
Accuracy: 0.71		
Metric	Score(0)	Score(1)
Score	0.80	0.47
Precision	0.73	0.61
Recall	0.87	0.38

Table 5.5: Performance score of Ada-boost

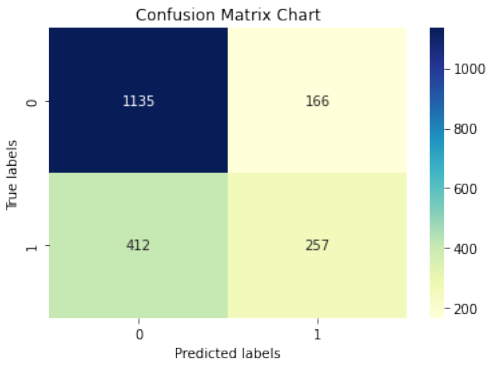

Figure 5.5: Confusion Matrix of ada-boost classifier

6 Comparison

Finally we tried to compare the performance of the above mentioned five models using one of the most widely used metrics for evaluation called receiver operating characteristic (ROC) curve.An ROC is one of the fundamental tools for diagnostic test evaluation and is created by plotting the true positive rate against the false positive rate at various threshold settings.The area under the ROC curve (AUC) is also commonly used to determine the predictability of a classifier. A higher AUC value represents the superiority of a classifier and vice versa [4].Figure 6.1 below shows the area under the curve with respect to the actual values of the five selected models and we as we can see from the figure the models that has better area under curve(AUC) are the logistic regression model and the adaboost ensemble classifier both having same are of "0.71".Since these models have same score in the ROC curve we made an additional comparison of their confusion matrix metrics like precision and recall.

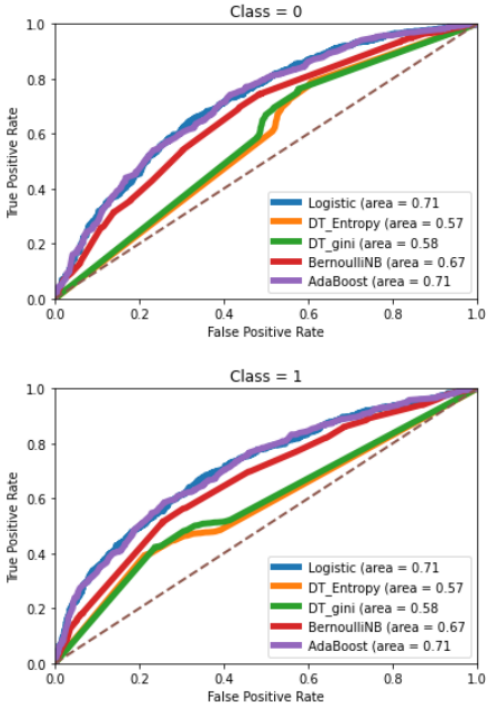

Figure 6.1: Comparison of ROC curves of different models

7 Conclusion

Based on the comparison on ROC and as we can see from figures 5.1, 5.5

Out of the 1301 actual instances of '0', Logistic regression predicted correctly 1215 of them but Ada-boost classifies 1135, Hence Logistic regression is better. Out of the 669 actual instances of '1', Logistic regression predicted correctly 186 of them and Ada-boost 257 of them. in this case Ada-boost predicts higher than the logistic regression.

As dataset for training increases in number the ability and performance of the model to predict expected to increase, thus, logistic regression is considered as best machine learning model for this work .

Bibliography

[1] *Compas analysis*, https://github.com/propublica/compas-analysis.

[2] *How we analyzed the compas recidivism algorithm*, https://www.propublica.org/article/how-we-analyzed-the-compas-recidivism-algorithm.

[3] *Naive bayes*, https://scikit-learn.org/stable/modules/naive_bayes.html.

[4] *Comparing different supervised machine learning algorithms*, https://bmcmedinformdecismak.biomedcentral.com/articles/10.1186/s12911-019-1004-8.