Supervised ML-Classification Credit Card default prediction

Mitesh Bhanushali

Points for Discussion

- Problem Statement
- Introduction
- Exploratory Data Analysis
- Handling class imbalance
- Multiple Models result
- Conclusion

Problem Statement

- This project is aimed at predicting the case of customers default payments in Taiwan. From the perspective of risk management, the result of predictive accuracy of the estimated probability of default will be more valuable than the binary result of classification - credible or not credible clients. We can use the K-S chart to evaluate which customers will default on their credit card payments. Credit card debt results when a client of a credit card company purchases an item or service through the card system. Debt accumulates and increases via interest and penalties when the consumer does not pay the company for the money they have spent.

Introduction

The given dataset consist of 30000 rows and 25 columns, the columns description is :
This research employed a binary variable, default payment (Yes = 1, No = 0), as the response variable. This study reviewed the literature and used the following 23 variables as explanatory variables:

1. X1: Amount of the given credit (NT dollar): it includes both the individual consumer credit and his/her family (supplementary) credit.
2. Gender (1 = male; 2 = female).
3. Education (1 = graduate school; 2 = university; 3 = high school; 4 = others).
4. Marital status (1 = married; 2 = single; 3 = others).
5. Age (year).

Introduction

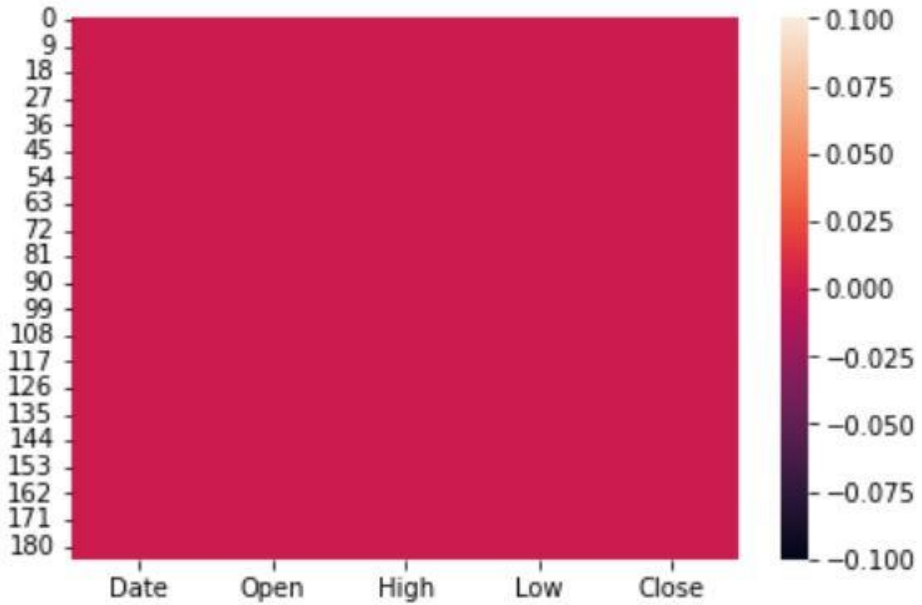
6. History of past payment. We tracked the past monthly payment records (from April to September, 2005) as follows: X6 = the repayment status in September, 2005; X7 = the repayment status in August, 2005; . . .;X11 = the repayment status in April, 2005. The measurement scale for the repayment status is: -1 = pay duly; 1 = payment delay for one month; 2 = payment delay for two months; . . .; 8 = payment delay for eight months; 9 = payment delay for nine months and above.
7. Amount of bill statement (NT dollar). X12 = amount of bill statement in September, 2005; X13 = amount of bill statement in August, 2005; . . .; X17 = amount of bill statement in April, 2005.
8. Amount of previous payment (NT dollar). X18 = amount paid in September, 2005; X19 = amount paid in August, 2005; . . .;X23 = amount paid in April, 2005.

Heatmap for showing the null value

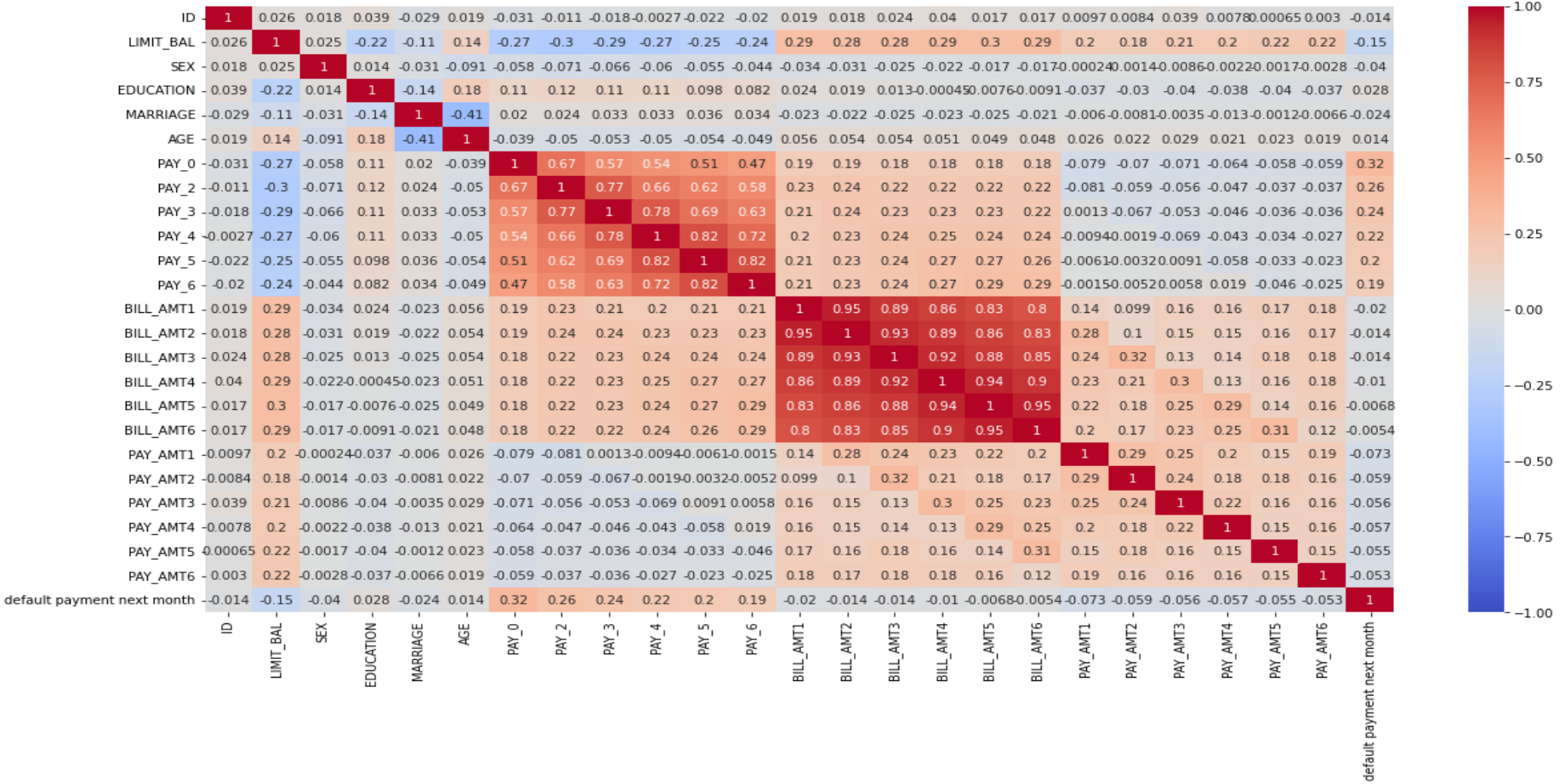
This graph shows that the given data does not contains any null values .

```
sns.heatmap(stock_data.isnull(),cbar=True)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f661ad42e10>
```

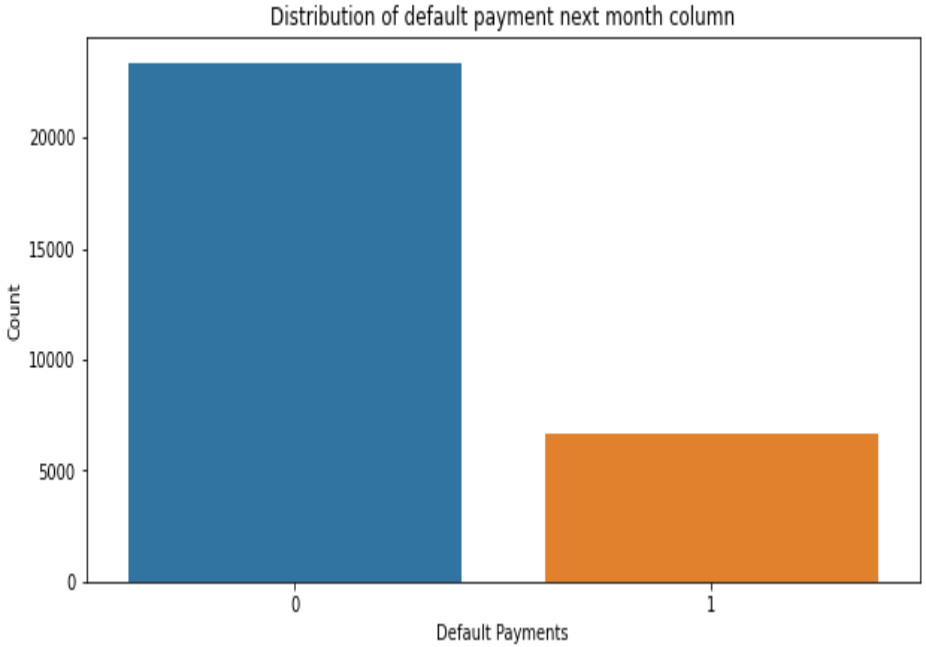


Correlation plot

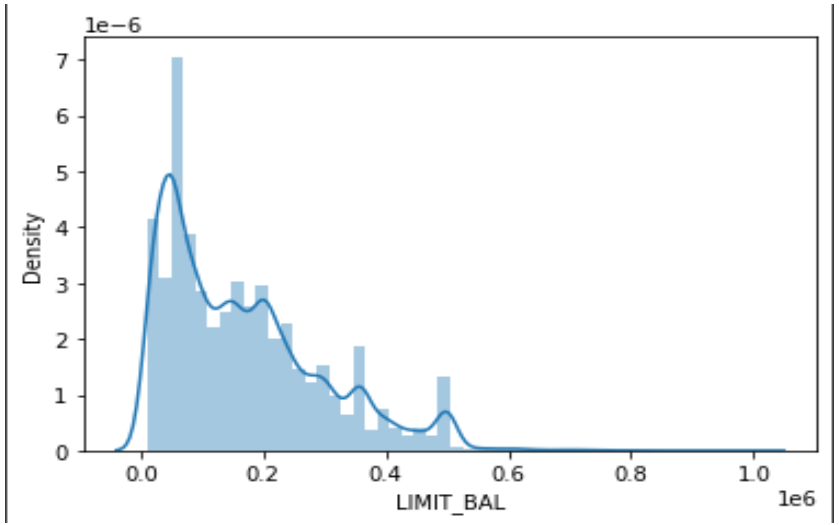


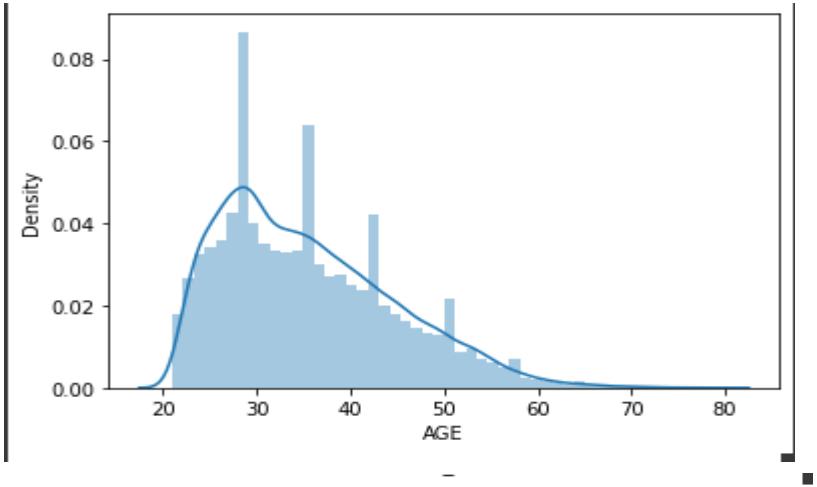
Checking target variable

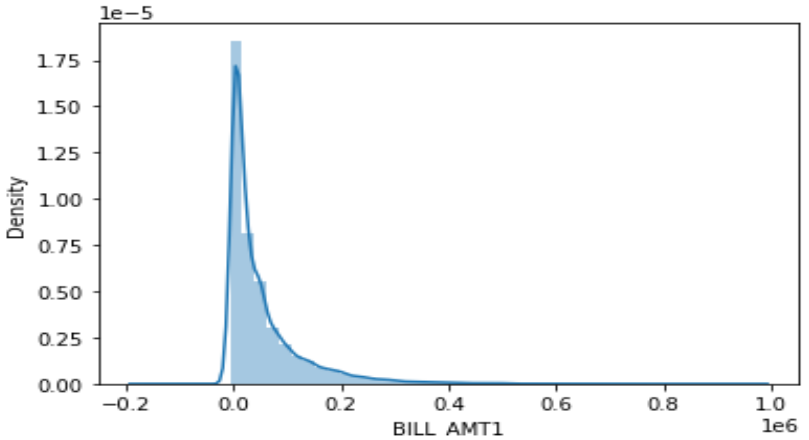
We can see class imbalance present in our dataset



Checking Distribution

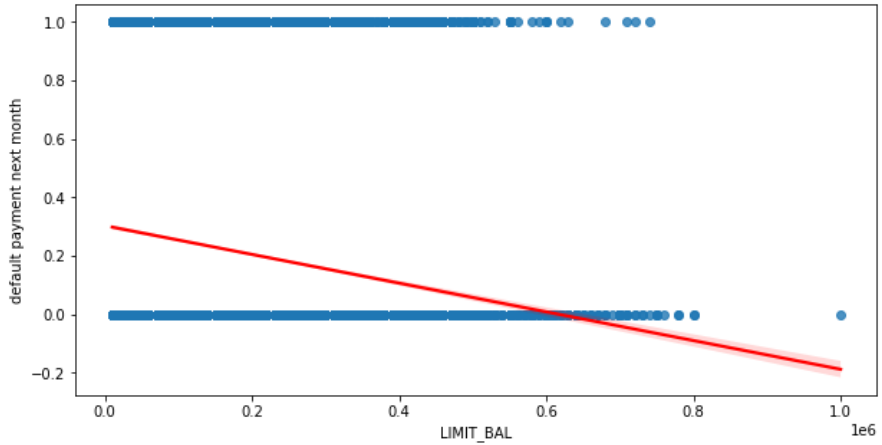


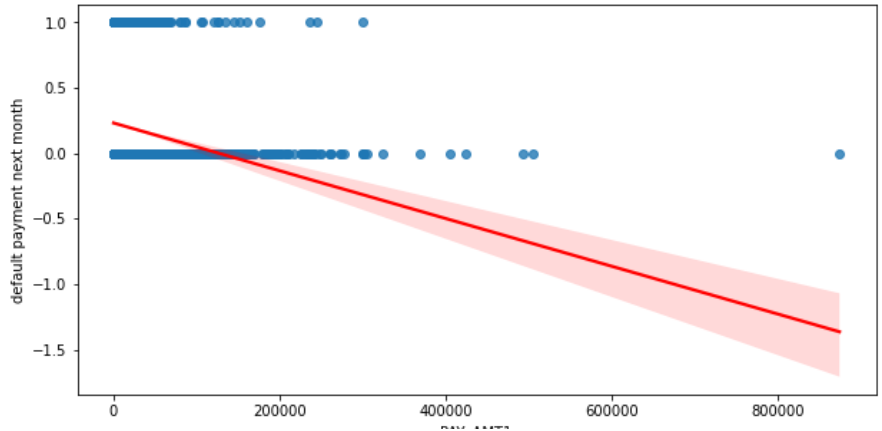




This plot shows that the limit balance, age, bill_amt are positively skewed.

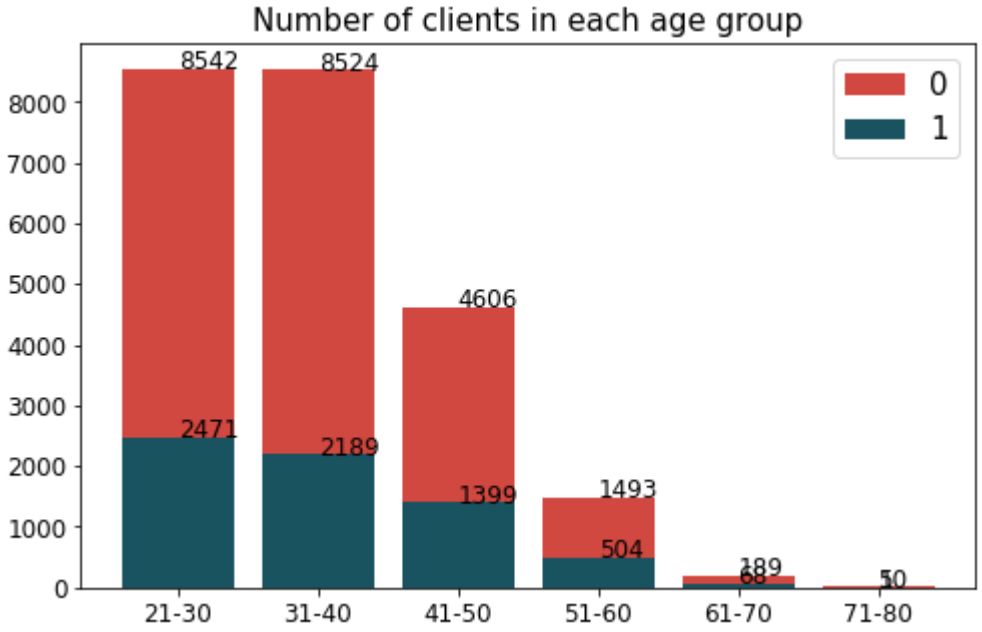
Regression Plots on features with target value.





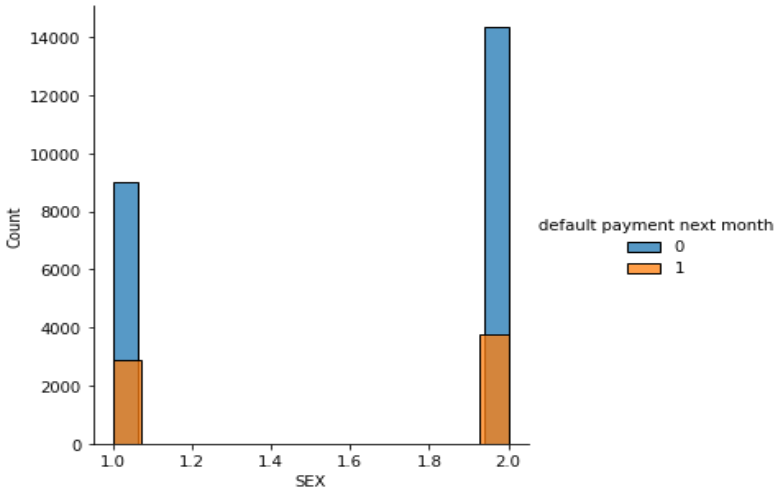
All the features are not linearly correlated to each other . As we see the Best Fit Line.

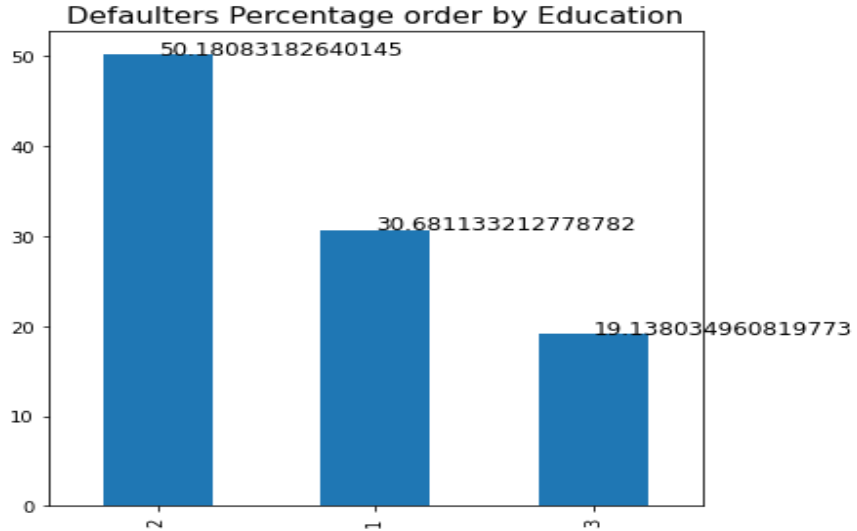
AGE group defaulters



Age group 21-40 have very high probability of defaulting. As the age increases the defaulters decreases.

Sex and Education group defaulters



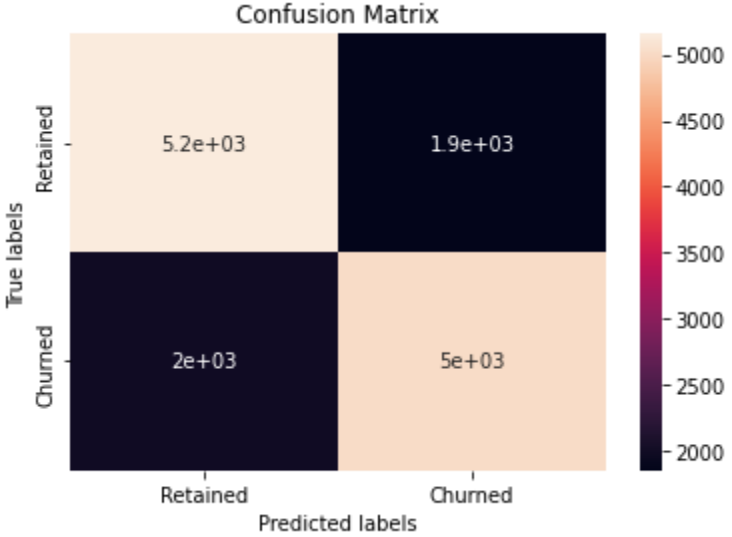


From above plot we can say that the if sex is 2 it has high probability of defaulter. University students tends to default more

SMOTE

- SMOTE (synthetic minority oversampling technique) is one of the most commonly used oversampling methods to solve the imbalance problem.
  It aims to balance class distribution by randomly increasing minority class examples by replicating them.
  SMOTE synthesises new minority instances between existing minority instances. It generates the **virtual training records by linear interpolation** for the minority class.
- The Original dataset had 30000 data points but after resampled dataset using SMOTE the data points increased to 46728.
- 0: 23364, 1: 23364 we can see that now both class are balance

Logistical model result



Confusion Matrix for test is :
[[5156 1854]
[1987 5022]]

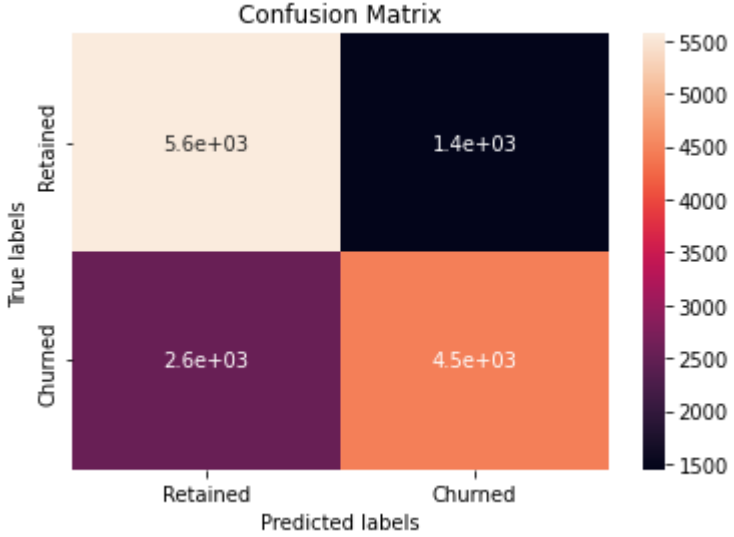
Logistical regression test prediction

5156 where class 0 and where predicted 0,1854 where class 0 but predicted 1,

1987 where class 1 but predicted as 0, 5022 where class 1 and predicted as 1

The roc auc on test data is 0.72

Decision Tree model result



Confusion Matrix for test is :
[5563 1447]
[2551 4458]

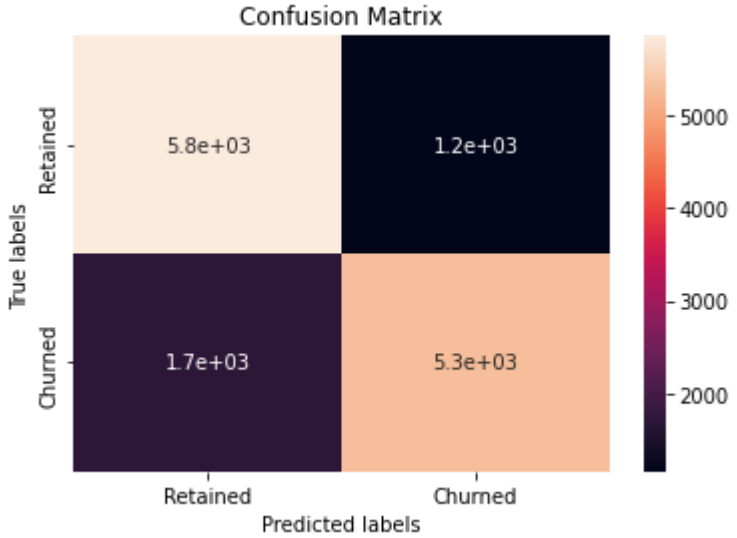
DT test prediction

5563 where class 0 and where predicted 0,1447 where class 0 but predicted 1,

2551 where class 1 but predicted as 0,4458 where class 1 and predicted as 1

The roc auc on test data is 0.71

Random Forest model result



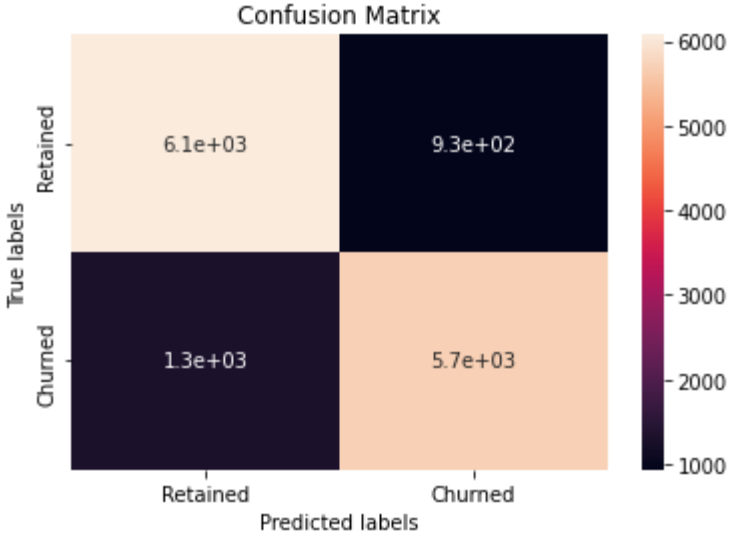
Confusion Matrix for test is :
[[5847 1163]
[1718 5291]]

Random Forest test prediction

5847 where class 0 and where predicted 0, 1163 where class 0 but predicted 1, 1718 where class 1 but predicted as 0,5291 where class 1 and predicted as 1

The roc auc on test data is 0.794

XGBoost model result



Confusion Matrix for test is :
[[6077 933]
[1295 5714]]

XGBoost test prediction

6077 where class 0 and where predicted 0, 933 where class 0 but predicted 1, 1295 where class 1 but predicted as 0, 5714 where class 1 and predicted as 1

The roc auc on test data is 0.84

All models results

The training and testing sets contain 70% and 30% data respectively. We have used multiple models to fit the data out of which XGBoost and Random Forest have performed great with AUC ROC score of 0.84 and 0.79 respectively.

Model	AUC ROC Train set	AUC ROC Test set
XGBoost	0.94	0.84
RF	0.83	0.79
KNN	0.80	0.76
DT	0.72	0.71
SVM	0.88	0.78
Logistic Regression	0.72	0.72

Conclusion

1. We have predicted the defaulters using multiple models in this project. We have used Logistical regression, Decision Tree, KNN, XGBoost, SVM. We have also used GridSearchCV to tune hyper parameters.

2. We have also seen the class imbalance so we did SMOTE to handle imbalance.

3. We did train test split and stratify the target variable.

4. We conclude that out of all models XGBoost performed well with roc auc score of 0.841

THANK YOU