Capstone Project

Credit Card Default Prediction

By
Sourav Chowdhury

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 ornot credible clients.

- To come up with a Machine Learning Model which best predicts the customer is going todefault or not.

Problem Faced

- Dataset containing outliers.

- Dataset was imbalanced.

- Applying Machine Learning Models, it was very much tedious to get the best algorithm which can perfectly gives the best prediction.

-Since Hyper Parameter Tuning was done on Algorithm, Thus it took more computational time to execute.

Tools Used

Programming Language - Python.

Data Analysis - Pandas

Data Visualisation - Matplotlib, Seaborn

ML Algorithm - Logistic Regression, Decision Tree, Random Forest, Gradient Boost, XG Boost, Support Vector Machines.

Reading Input Dataset

To load the dataset in the colab notebook, first we mounted the notebook with google drive. And then read the dataset using pandas built in function read_excel.

```
from google.colab import drive
drive.mount('/content/drive')
```

```
  dataset= pd.read_excel
('/content/drive/MyDrive/Supervised ML Classification (Credit Card Default Prediction)/default of
credit card clients.xls)
```

Data Pipeline

- **Data Processing - 1(Inspection):** In this part we have Checked the dataset with its shape size and content.
- **Data Processing - 2:** In this part we have done the Exploratory data analysis Checked for Outliers, Univariate and Multivariate analysis and Correlation.
- **Data Preparation :** Done Feature engineering on the data by assigning dummy values to categorical features(One hot encoding), Handling Class Imbalance, Data Transformation, Train Test Split.
- **Creating Model :** Finally, in this we have created and trained the models with iterative process so as to make ideal model with better Evaluation metrics.

Data Summary

Attribute ID	Attribute Name	Attribute Description
X1	Limit_Bal	Amount Of Given Credit
X2	Sex	Gender (1 = Male; 2 = Female)
X3	Education	Education (1 = Graduate School; 2 = University; 3 = High School; 4: Others)
X4	Marriage	Marital Status (1 = Married; 2 = Single; 3 = Others).
X5	Age	Age (Year)
X6-X11	Pay_1 to Pay_6	History of past payment (From April to September 2005): X6 = The repayment status in September 2005… X11 = The repayment status in April, 2005 History of past payment tracked via past monthly payment records (-1 = Payment on time; 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).
X12–X17	Bill_Amt1 to Bill_Amt6	Amount of bill statement (Dollar) X12 = amount of bill statement in September 2005… X17 = amount of bill statement in April 2005
X18–X23	Pay_Amt1 to Pay_Amt6	Amount of previous payment (Dollar) X18 = Amount paid in September 2005… X23 = Amount paid in April 2005
X24	default payment next month	Default= 1 and healthy= 0

Define Dependent Variable

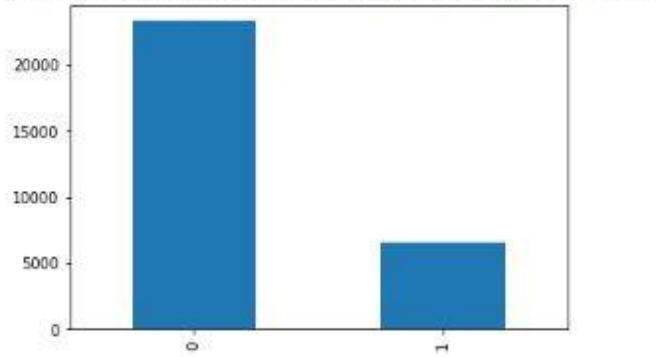
The Dependent Variable is whether the customer has

Defaulted on his credit card - 1

Not Defaulted - 0

Since Dataset was highly imbalanced as there were only 6636 customers out of 30000 who actually defaulted, So we needed to handle class Imbalance.

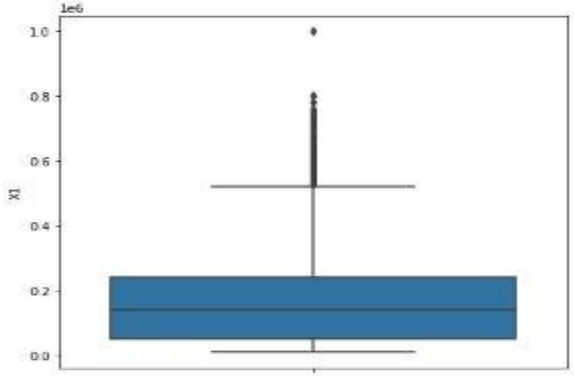
```
0    23364
1     6636
Name: Y, dtype: int64
<matplotlib.axes._subplots.AxesSubplot at 0x7f3efbcdd4d0>
```

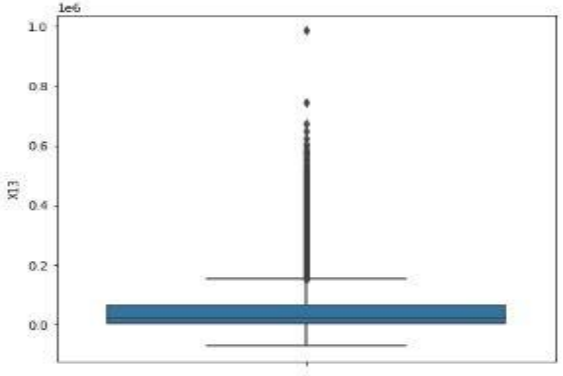


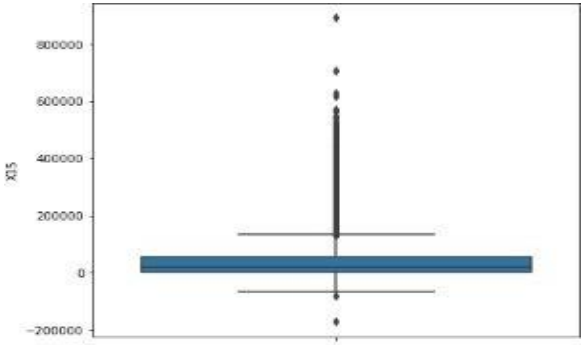
EDA

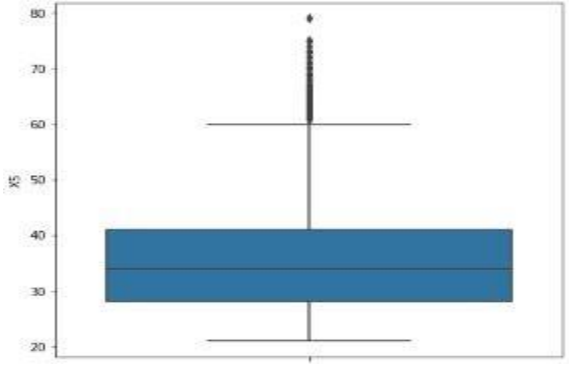
1. Univariate Analysis :

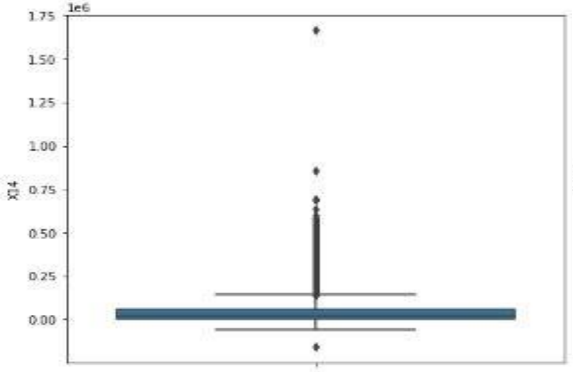
- ❖ Handling Outliers: Variables containing outliers were detected using Box Plot, Thus used percentile method to cap the outliers.

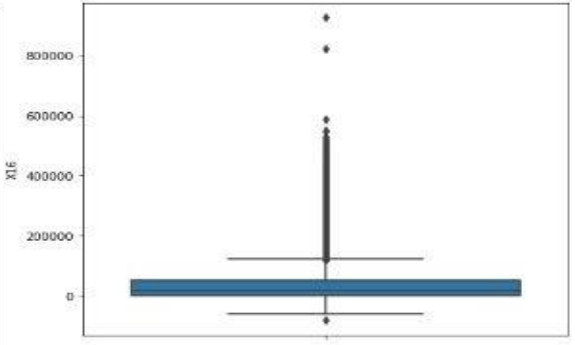






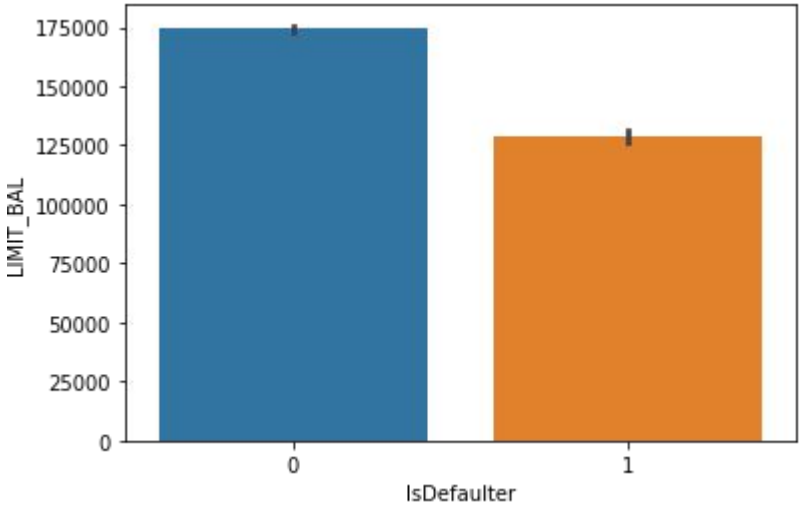






EDA

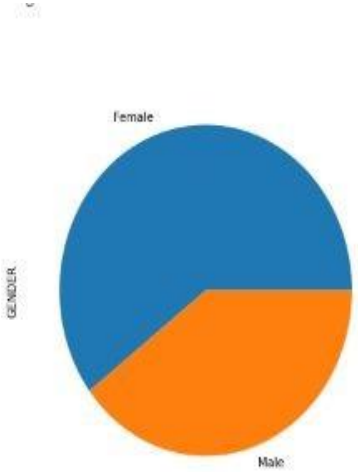
Analysis on Limit Balance:

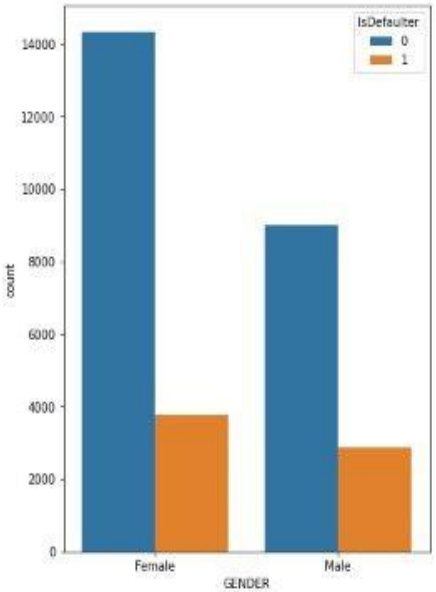


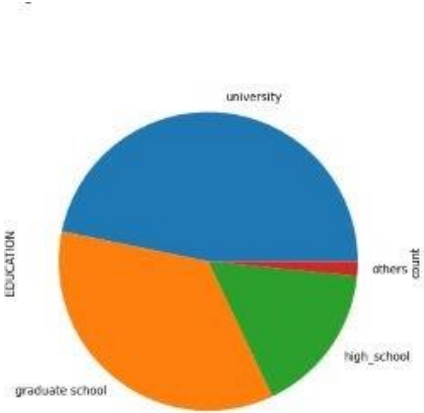
Thus for Defaulters the limit amount is less comparative to non defaulters

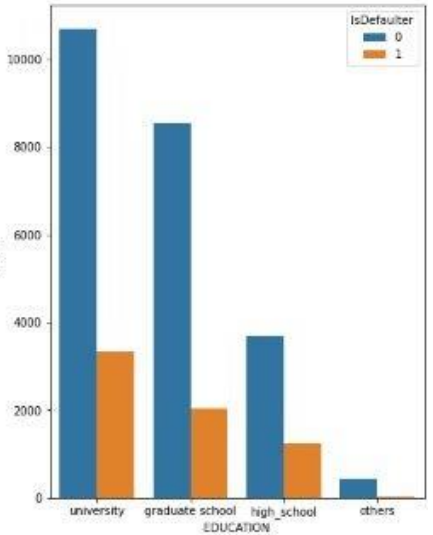
EDA

Analysis on Gender, Education:





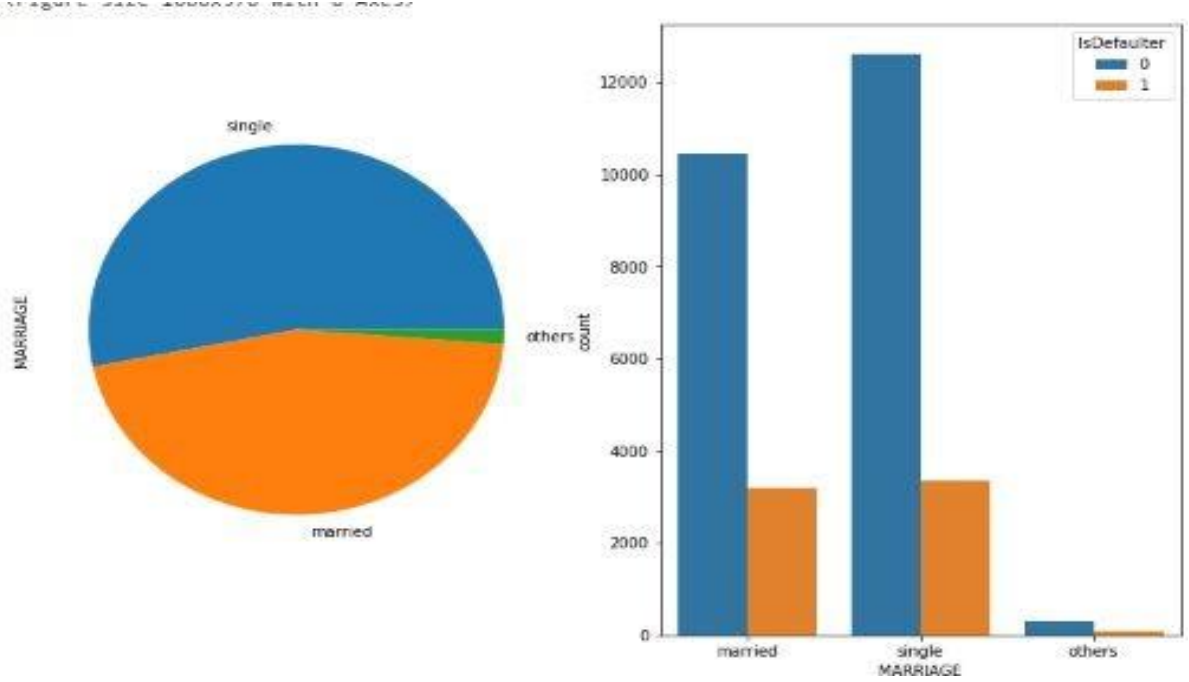




- Dataset have more female's credit card holder, so number of defaulter have high proportion of females.
- Number of defaulters have a higher possibility that he is graduated from school and university ie.The order follows as University>graduate school>High School>Others

EDA

Analysis on Marriage:

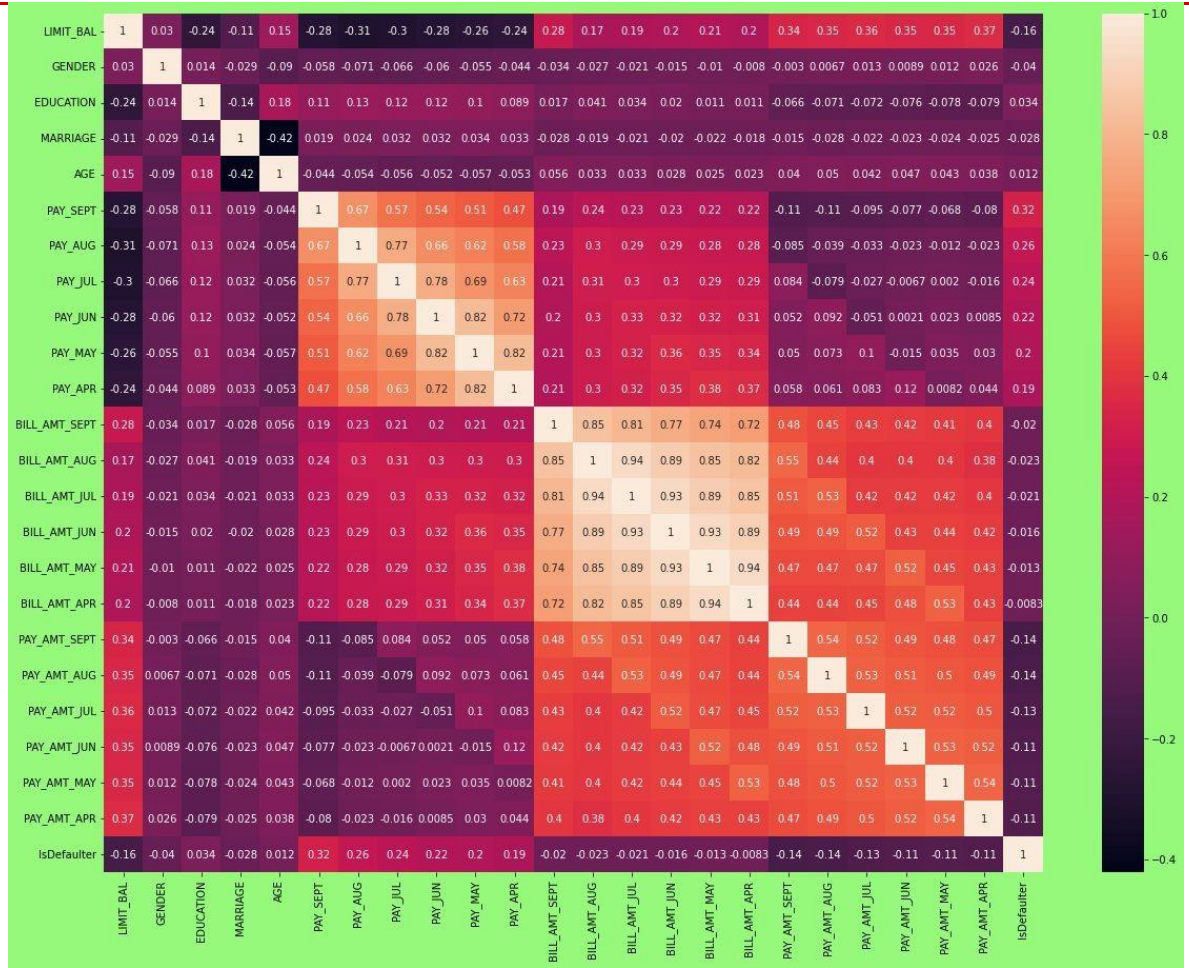


- Number of defaulters have a higher proportion of Singles.

EDA (Correlation)

```
LIMIT_BAL       -0.157
GENDER          -0.040
EDUCATION        0.034
MARRIAGE        -0.028
AGE              0.012
PAY_SEPT         0.325
PAY_AUG          0.264
PAY_JUL          0.235
PAY_JUN          0.217
PAY_MAY          0.204
PAY_APR          0.187
BILL_AMT_SEPT   -0.020
BILL_AMT_AUG    -0.023
BILL_AMT_JUL    -0.021
BILL_AMT_JUN    -0.016
BILL_AMT_MAY    -0.013
BILL_AMT_APR    -0.008
PAY_AMT_SEPT    -0.145
PAY_AMT_AUG     -0.140
PAY_AMT_JUL     -0.126
PAY_AMT_JUN     -0.114
PAY_AMT_MAY     -0.108
PAY_AMT_APR     -0.115
IsDefaulter      1.000
Name: IsDefaulter, dtype: float64
```

Since Payment history variables are Highly correlated with IsDefaulter.



Data Preparation

One hot encoding:

Creating dummies column for the given feature

```
dataset_copy = pd.get_dummies(dataset_copy, columns = ['EDUCATION', 'MARRIAGE'])
dataset_copy = pd.get_dummies(dataset_copy, columns=['PAY_SEPT', 'PAY_AUG', 'PAY_JUL', 'PAY_JUN',
'PAY_MAY', 'PAY_APR'])
# LABEL ENCODING FOR Gender
encoders_nums = {"GENDER":{"FEMALE": 0, "MALE": 1}}
dataset_copy = dataset_copy.replace(encoders_nums)
```

Shape after one hot encoding

```
(30000, 87)
```

Data Preparation

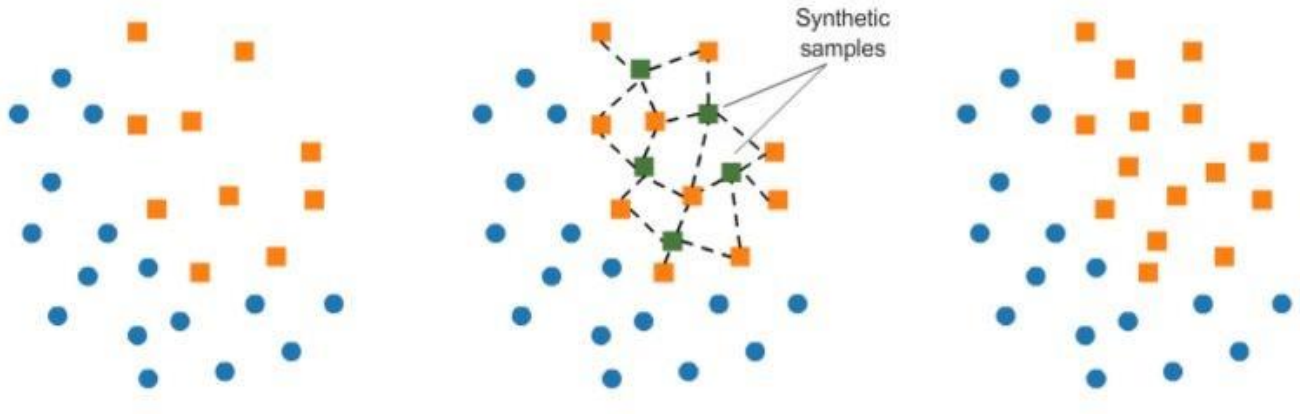
Handling Class Imbalance:

We have used SMOTE - Synthetic Minority Oversampling Technique

SMOTE works by randomly picking a point from the minority class and computing the k-nearest neighbors for this point. The synthetic points are added between the chosen point and its neighbors.

```
Original unbalanced dataset shape (30000, 87)

Resampled balanced dataset shape (46728,85)
```



Transformation of Data

- Data was unscaled the values have different ranges, thus to best fitin any machine learning model we need to scale the data.
- The goal is to enforce a level of consistency or uniformity to dataset.
- We have used Standardisation method to scale the data.

Splitting Data

- Data Splits into Train Test dataset
- Training dataset is used to make algorithm learn, and test is to check the performance of machine learning model on unknown data.
- We took 80% training data and 20% test data.
- Shape of training set `(37382, 86)` data
- Size of test data set (9346, 86)

Fitting Different Model

- Logistic Regression
- Decision Tree Classifier
- Random Forest Classifier
- Gradient Boosting Classifier
- Xg Boost Classifier
- Support Vector Machines

Cross Validation & Hyperparameter Tuning

- It is a Technique used to get best set of hyper parameters for best performance
- Cross validation makes set of combinations of small dataset of training sets and trains the model alternatively on each set measuring the performance gives the best set of hyperparameters for the model.
- Tuning the hyperparameter of respective algorithms is necessary for getting better accuracy and to avoid overfitting.

Model Comparison Summary

BaseLine Models :

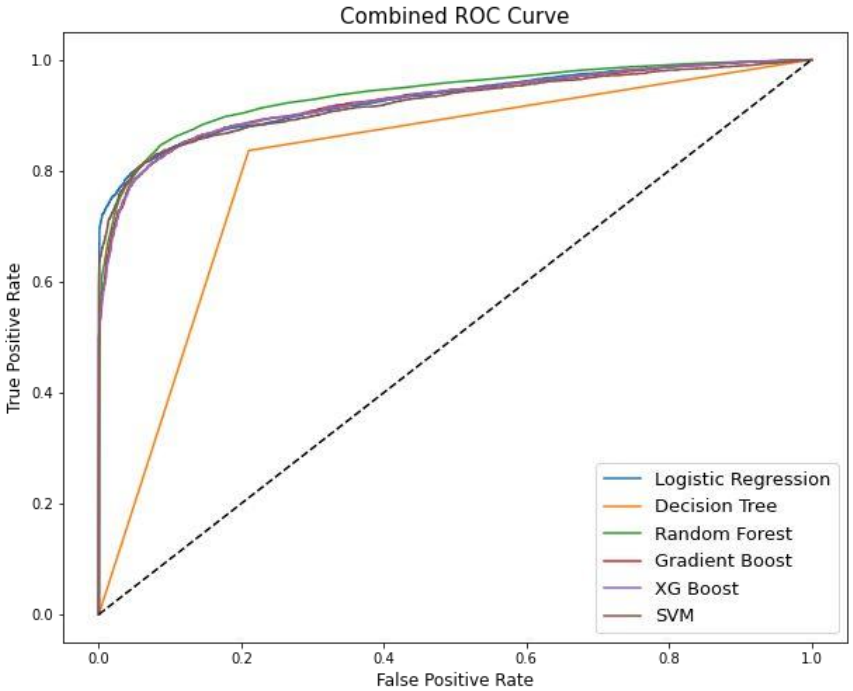
SL NO.	Classifier	Train Accuracy	Test Accuracy	Precision	Recall	F1 Score	ROC Score
1	Logistic Regression	0.875	0.875	0.803	0.939	0.866	0.883
2	Decision Tree	1.0	0.811	0.826	0.802	0.814	0.811
3	Random Forest	0.999	0.877	0.834	0.912	0.871	0.879
4	Gradient Boost	0.871	0.868	0.802	0.923	0.859	0.874
5	XG Boost	0.87	0.867	0.801	0.924	0.858	0.874
5	Support Vector Machines	0.879	0.876	0.805	0.938	0.866	0.884

Optimal Models:

SL NO.	Classifier	Train Accuracy	Test Accuracy	Precision	Recall	F1 Score	ROC Score
1	Logistic Regression	0.874	0.875	0.803	0.939	0.865	0.883
2	Decision Tree	0.859	0.839	0.79	0.875	0.83	0.842
2	Random Forest	0.857	0.849	0.806	0.881	0.842	0.851
2	Gradient Boost	0.959	0.88	0.834	0.92	0.875	0.884
2	XG Boost	0.995	0.881	0.836	0.918	0.875	0.884
2	Support Vector Machine	0.879	0.876	0.805	0.938	0.866	0.884

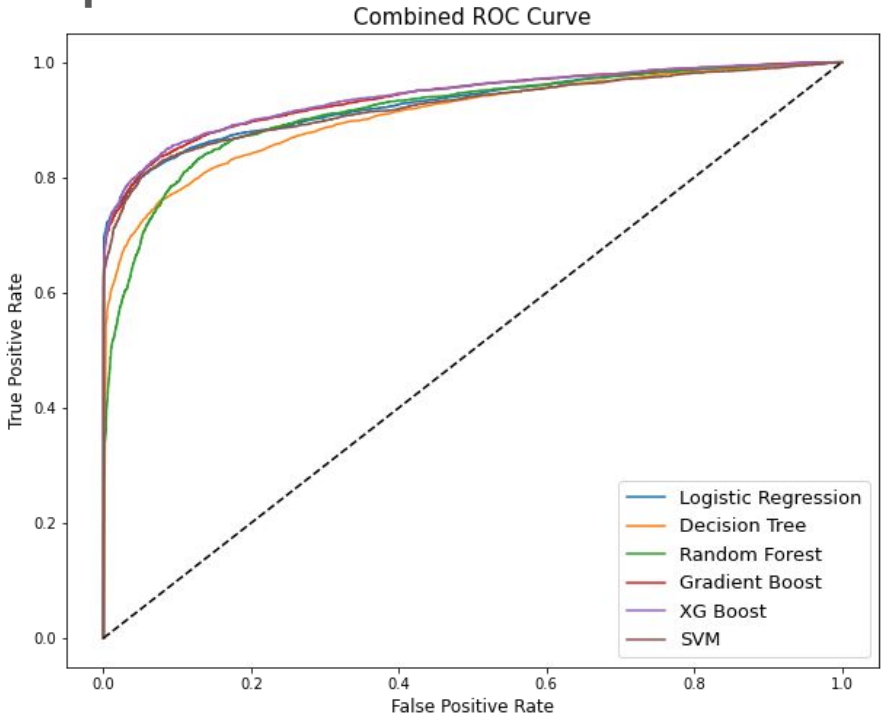
Model Comparison Summary

Baseline Models:



As per Baseline Model, the Random Forest gives best model.

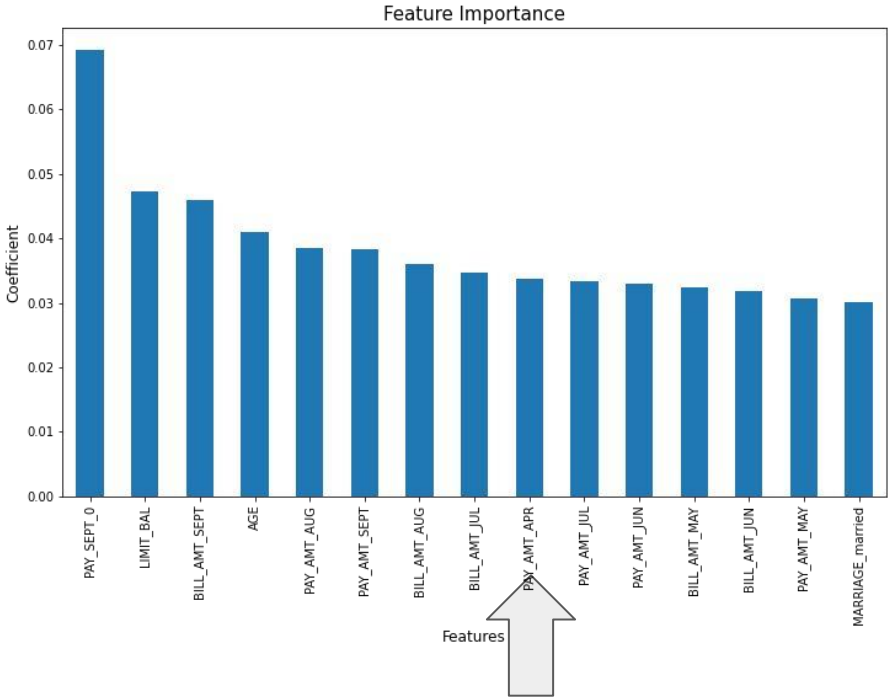
Optimal Models:



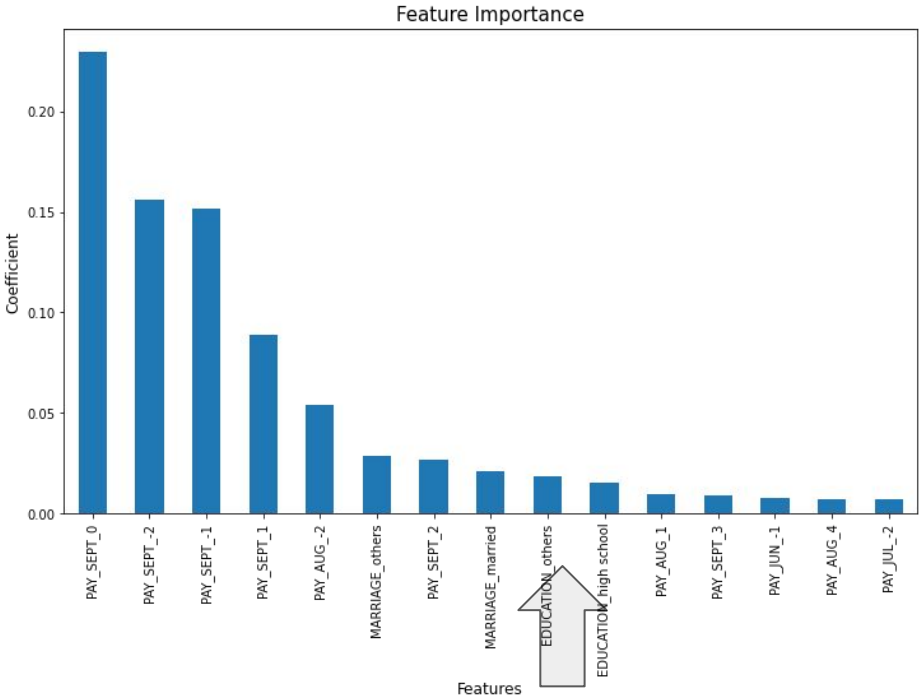
As per optimal Model accuracy and ROC curve the best model is XG Boost.

Model Validation & Selection (Feature Importance)

The Summary table shows that for Baseline Models Random Forest is giving best evaluation metrics and among Optimal models XG Boost is giving best results. Let's see Feature Importance from both of these.



Random Forest Classifier



XG Boost Classifier

Conclusion

- After Basic EDA,
  a. Limit amount was less for Credit Card Defaulters comparative to non defaulters.
  b. Dataset have more females credit card holder, so number of defaulter have high proportion of females.
  c. Credit Card No. of Defaulters as per education from top to bottom : University>graduate school>High School>Others
  d. Number of defaulters have a higher proportion of Singles.
- From all baseline models, Random Forest classifier shows highest test accuracy, F1 score and ROC score.
- Baseline model of Random forest and decision tree shows huge difference in train and test accuracy which shows overfitting.
- After cross validation and hyperparameter tuning, XG Boost shows highest test accuracy score of 88.10% and ROC Score is 0.884.
- This XG Boost with optimal model is best to predict the customer credit Card Default.