Capstone Project

Credit Card Default Prediction

Team Members

LAKSHMI KEERTHANA
TITO VARGHESE

Points to Discuss

- Problem Statement
- Data Summary
- Data Cleaning
- Exploratory Data Analysis
- Feature Engineering
- Model Training
- Evaluation Metrics from model
- Conclusion
- Challenges

Problem Statement

- To extract,observe, analyse and build a classification model based on dataset with the details of credit card holders of a bank in Taiwan for the period April to September, 2005.

- The features available include some basic customer demographics (gender, education, marital status and age), available credit line, their history of payment/default for the six months mentioned (Apr--Sep '05), their bill amounts and their payment amounts for that period and a binary target variable indicating default the following month.

- We perform some EDA to understand the data and clean the data, engineer relevant features, build predictive models to predict default and perform some statistical analyses to obtain a greater understanding of the features and their interactions. We finish with some case scenarios where the predictive model could be applied.

DATA SUMMARY

The dataset info gives us some crucial insights into our Features data type and Non Null Count:

```
#   Column     Non-Null Count  Dtype

--- ------     --------------  -----

 0   LIMIT_BAL  30000 non-null  int64

 1   SEX        30000 non-null  int64

 2   EDUCATION  30000 non-null  int64

 3   MARRIAGE   30000 non-null  int64

 4   AGE        30000 non-null  int64

 5   PAY_1      30000 non-null  int64

 6   PAY_2      30000 non-null  int64

 7   PAY_3      30000 non-null  int64

 8   PAY_4      30000 non-null  int64

 9   PAY_5      30000 non-null  int64

 10  PAY_6      30000 non-null  int64

 11  BILL_AMT1  30000 non-null  int64

 12  BILL_AMT2  30000 non-null  int64

 13  BILL_AMT3  30000 non-null  int64

 14  BILL_AMT4  30000 non-null  int64

 15  BILL_AMT5  30000 non-null  int64

 16  BILL_AMT6  30000 non-null  int64

 17  PAY_AMT1   30000 non-null  int64

 18  PAY_AMT2   30000 non-null  int64

 19  PAY_AMT3   30000 non-null  int64

 20  PAY_AMT4   30000 non-null  int64

 21  PAY_AMT5   30000 non-null  int64

 22  PAY_AMT6   30000 non-null  int64

 23  defaulter  30000 non-null  int64
```

The given dataset consist of 30000 rows and 24 columns

The feature 'defaulter' is our dependent/target feature

DATA CLEANING

Data Cleaning Summary

1. The given dataset does not contain any missing values
2. The given dataset had 35 duplicate rows and we have dropped the duplicate records.
3. The target variable consist of imbalance data with 77.87% Non defaulters(0 value) and 22.12% 1 defaulters (1 value).

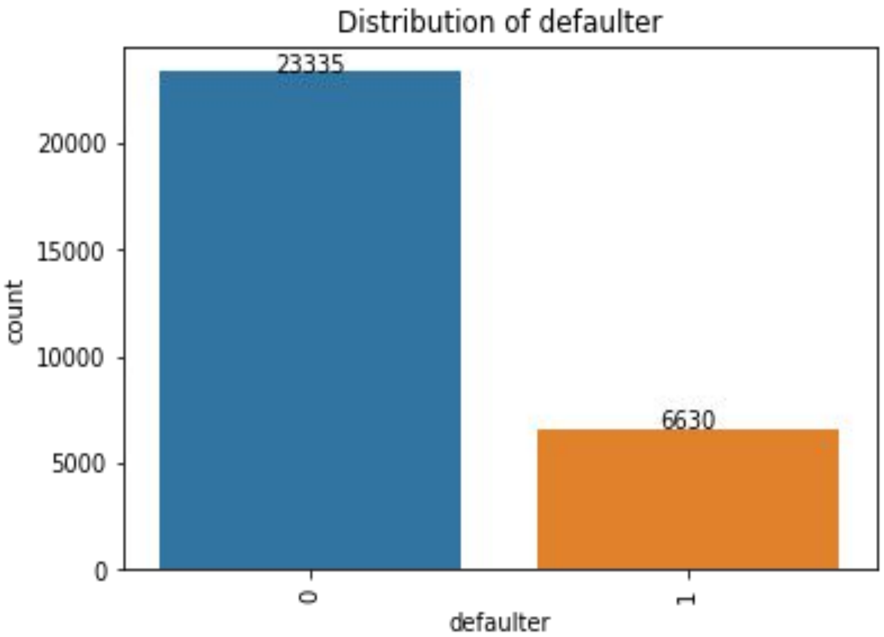
Renaming the Column name

1. Renamed the columns using the first record given in the dataset and dropped the first record.
2. Renamed our target variable to defaulter and PAY_0 column to PAY_1
3. Converted the datatypes of all columns from object to int datatype because all columns contains numerical values.
4. Drop the ID column from the dataset,since its not an influential feature in our modeling.

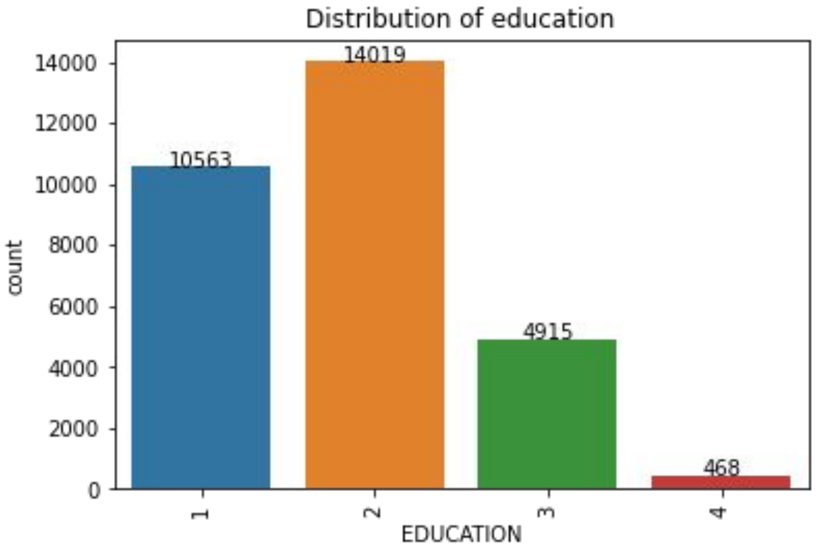
The Dataset Inspection Summary

1. The average credit card limit/consumer credit amount is 167484.32(NT Dollars)
2. The maximum number of credit card holders were females in Taiwan.
3. The given dataset consist of 30000 rows and 24 columns
4. The feature 'defaulter' is our dependent/target feature
5. The most number of credit card holders were having university degree education.
6. The most of the customers marriage status was Single, who carries a credit card.

EXPLORATORY DATA ANALYSIS

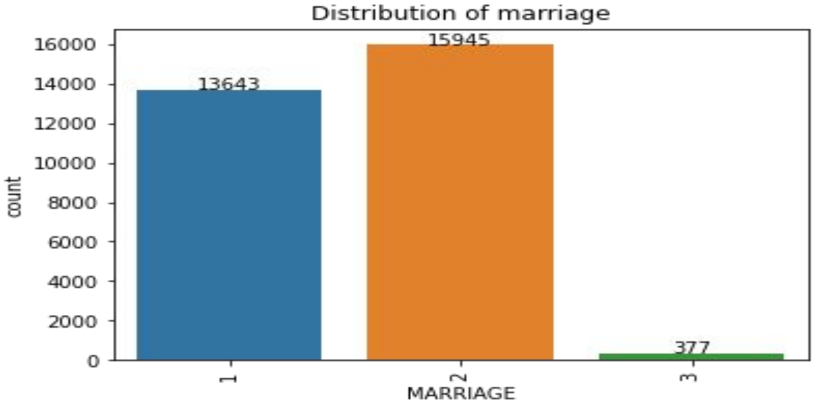


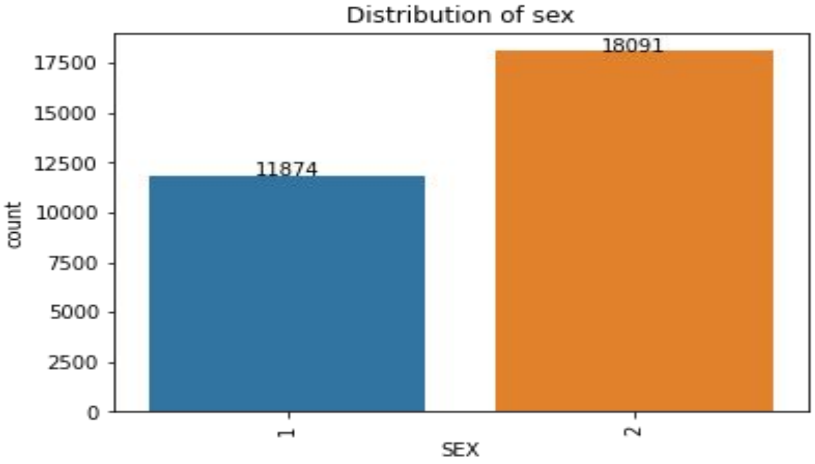
The target variable consist of imbalance data with 77.87% Non defaulters(0 value) and 22.12% 1 defaulters (1 value).

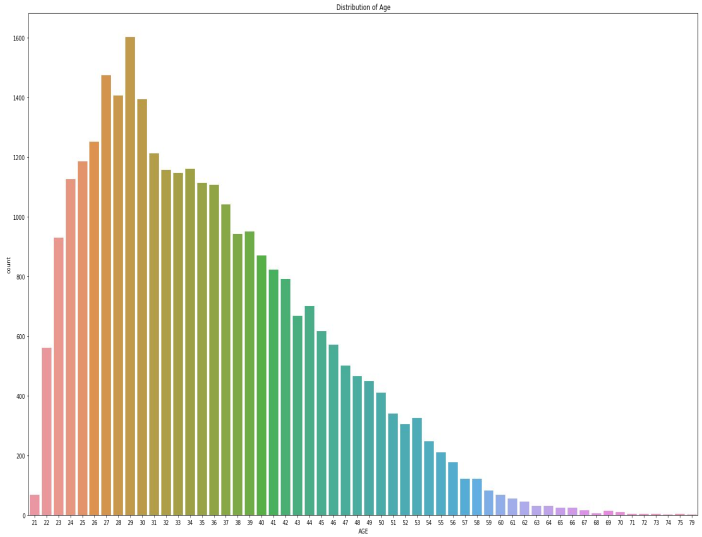


Most number of customers were holding a university degree as their educational qualification followed by graduates degree holders.

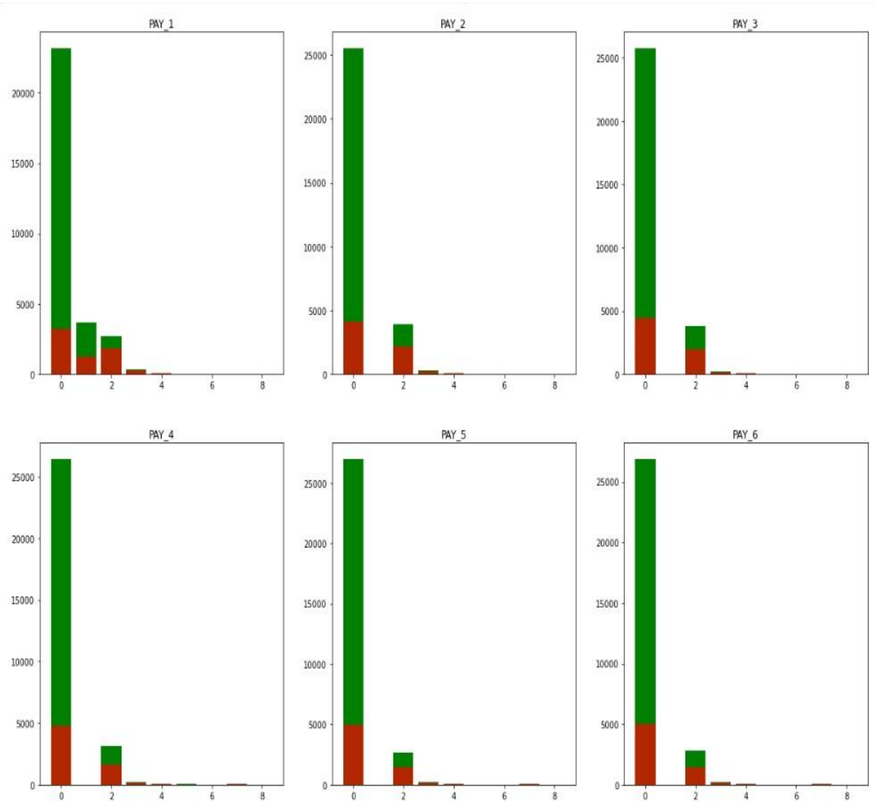
Univariate Analysis







- **We can conclude that the most number of credit card holders were not married(Single)**
- **The most number of credit card holders in Taiwan were females.**
- **The most frequent age of the credit card holder is between 25-34.**

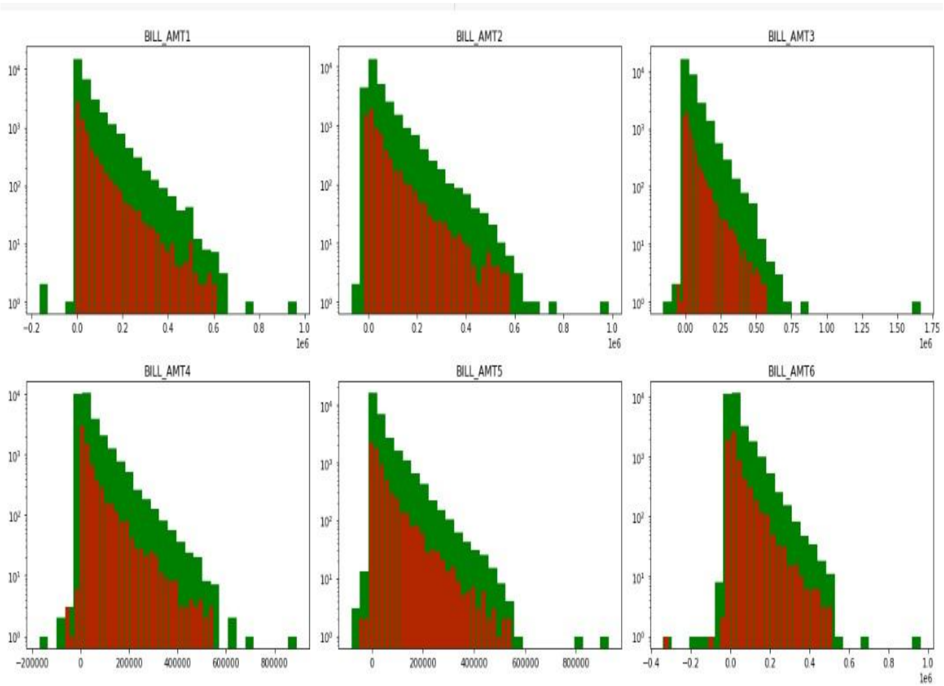


Repayment Status(Payment History)

The above figure shows the bar plot of payment history status for past six months starting from September to April , which show the count of defaulters and non-defaulter

The green bins shows the count of payment status of all the customers (both defaulters and non-defaulters). On the other hand, the red bins shows the count of payment status explicitly for the customers who were defaulters.

From the above graph, we can conclude that if the payment status is greater than 2 months,then there is a 90% chance of the customer to default the payment



Previous Amount Paid (PAY_AMT)

The above histogram shows the distribution of Bill amount generated for each month explicitly for defaulters

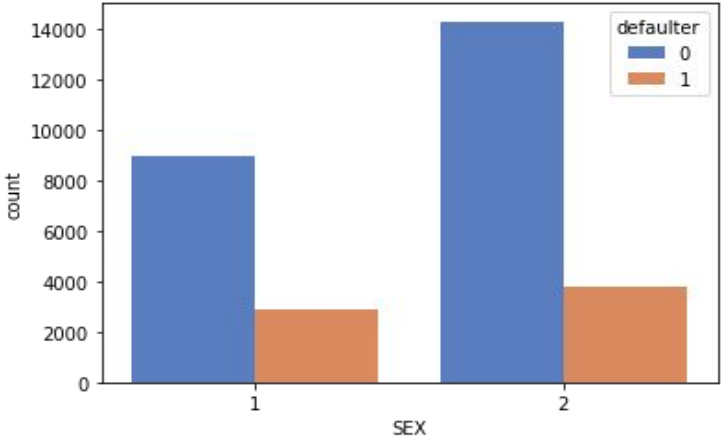
The green bins in the histograms shows the bill_amount for all the customers from September to April Month.

The red bins in the histogram tells the payment amount of customers who were actually a defaulter from September to April Month.

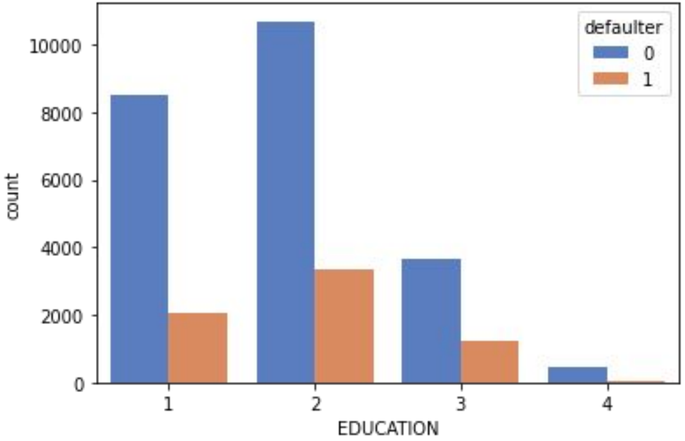
Hence payment amount features are more significant variables compared to the bill amount features.

BI-VARIATE ANALYSIS

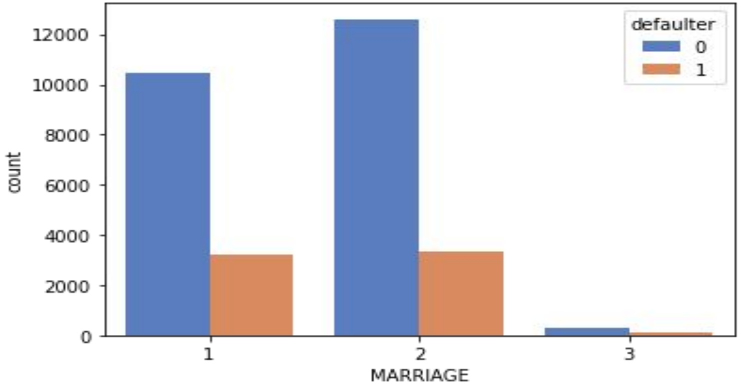
Defaulter vs Sex



Defaulter vs Education



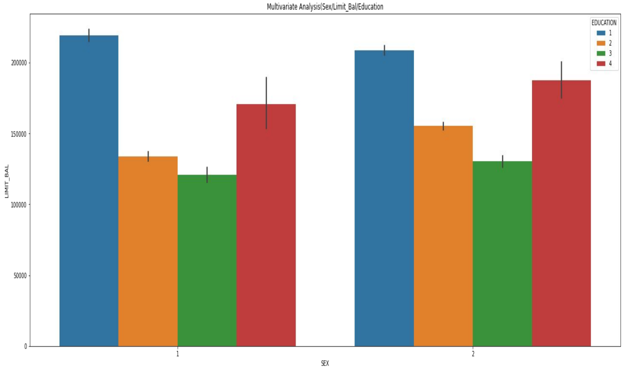
Defaulter vs Marriage



- It is evident from the count plot output that males have overall less default payment rate w.r.t females
- The credit card holders with a university degree were the customers with the highest number of default payment rate compared to other degree holders.
- It is evident from the plot that both the credit card holders who were singles and married used to do default in payments.

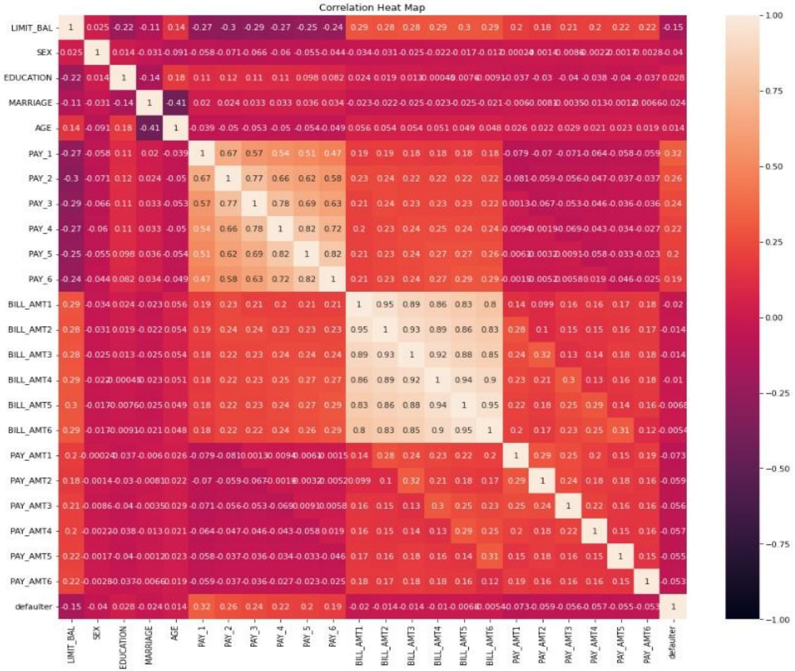
MULTIVARIATE ANALYSIS

Sex vs Credit limit vs Education



The above figure tells us that the highest LIMIT_BAL/credit limit amount is given to the graduate education credit card holders in both the sex.

HEAT MAP



We can see that next month default prediction is dependent on payment repayment status of past six months of all the features given to us. But there is multicollinearity between the Payment Repayment Status features.

FEATURE ENGINEERING

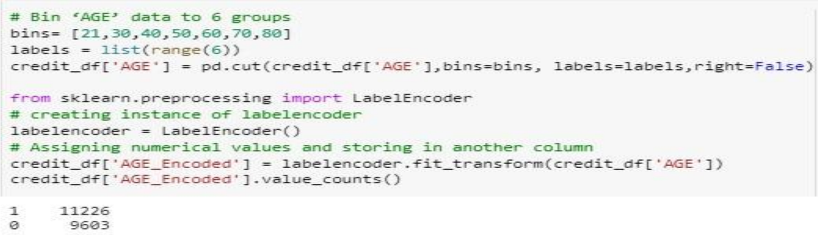
1.Replaced 0 class/category in marriage column into class/category 3='others' because 0 class is not defined in the marriage variable.

2.Similarly, we replaced few undefined classes like 0, 5 and 6 in education column into class 4='others' using a function name education.

3.Replaced negative values in payment history columns i.e (PAY_1,PAY_2...PAY_6) into class 0 -pay duly on time.

4.Two new Features were derived from the existing independent features,because it will help to train our model more effectively

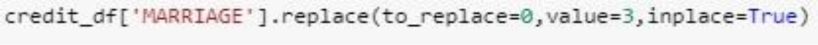
5.We have also implemented binning on one feature i.e AGE column in order to better train our model efficiently.
6.Lastly performed One-Hot Encoding on few features like AGE,SEX,MARRIAGE and PAY_0-PAY_6 to improve the model evaluation



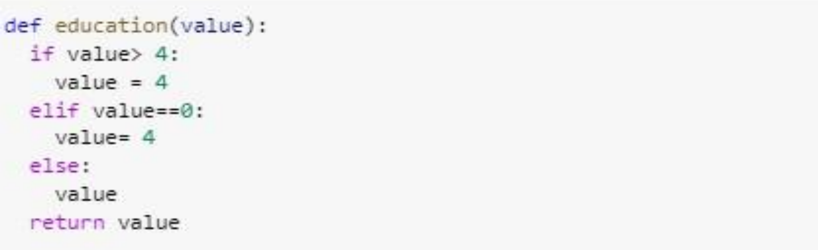
```
# Bin ‘AGE’ data to 6 groups
bins= [21,30,40,50,60,70,80]
labels = list(range(6))
credit_df['AGE'] = pd.cut(credit_df['AGE'],bins=bins, labels=labels,right=False)

from sklearn.preprocessing import LabelEncoder
# creating instance of labelencoder
labelencoder = LabelEncoder()
# Assigning numerical values and storing in another column
credit_df['AGE_Encoded'] = labelencoder.fit_transform(credit_df['AGE'])
credit_df['AGE_Encoded'].value_counts()
```

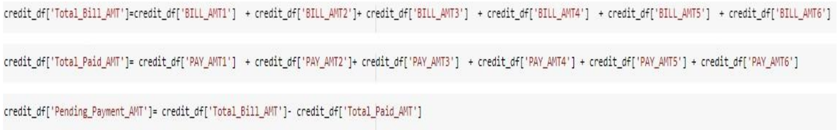
```
1    11226
0     9603
```



```
credit_df['MARRIAGE'].replace(to_replace=0,value=3,inplace=True)
```



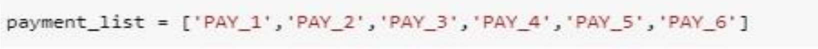
```
def education(value):
  if value> 4:
    value = 4
  elif value==0:
    value= 4
  else:
    value
  return value
```



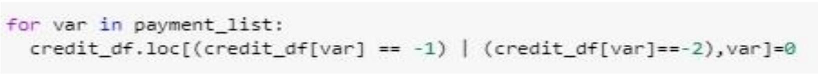
```
credit_df['Total_Bill_AMT']=credit_df['BILL_AMT1']  + credit_df['BILL_AMT2']+ credit_df['BILL_AMT3']  + credit_df['BILL_AMT4']  + credit_df['BILL_AMT5']  + credit_df['BILL_AMT6']
```

```
credit_df['Total_Paid_AMT']= credit_df['PAY_AMT1']  + credit_df['PAY_AMT2']+ credit_df['PAY_AMT3']  + credit_df['PAY_AMT4'] + credit_df['PAY_AMT5'] + credit_df['PAY_AMT6']
```

```
credit_df['Pending_Payment_AMT']= credit_df['Total_Bill_AMT']- credit_df['Total_Paid_AMT']
```

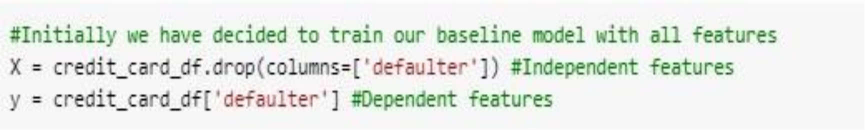


```
payment_list = ['PAY_1','PAY_2','PAY_3','PAY_4','PAY_5','PAY_6']
```

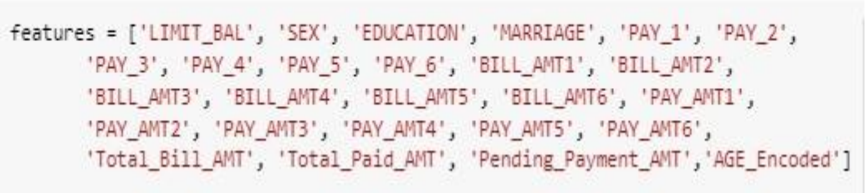


```
for var in payment_list:
  credit_df.loc[(credit_df[var] == -1) | (credit_df[var]==-2),var]=0
```

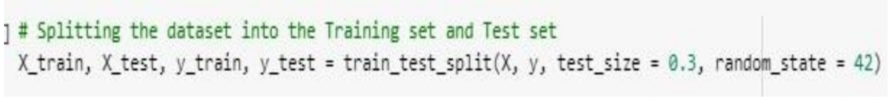
Model Formulation



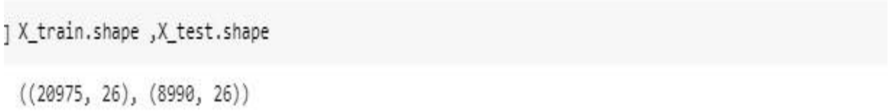
```
#Initially we have decided to train our baseline model with all features
X = credit_card_df.drop(columns=['defaulter']) #Independent features
y = credit_card_df['defaulter'] #Dependent features
```



```
features = ['LIMIT_BAL', 'SEX', 'EDUCATION', 'MARRIAGE', 'PAY_1', 'PAY_2',
       'PAY_3', 'PAY_4', 'PAY_5', 'PAY_6', 'BILL_AMT1', 'BILL_AMT2',
       'BILL_AMT3', 'BILL_AMT4', 'BILL_AMT5', 'BILL_AMT6', 'PAY_AMT1',
       'PAY_AMT2', 'PAY_AMT3', 'PAY_AMT4', 'PAY_AMT5', 'PAY_AMT6',
       'Total_Bill_AMT', 'Total_Paid_AMT', 'Pending_Payment_AMT','AGE_Encoded']
```

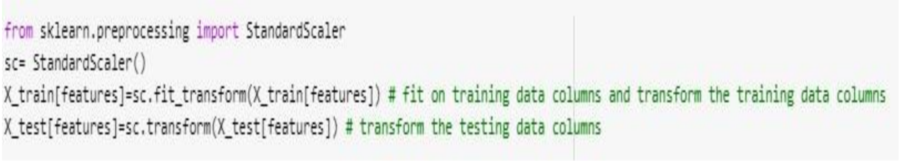


```
] # Splitting the dataset into the Training set and Test set
  X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3, random_state = 42)
```



```
] X_train.shape ,X_test.shape

  ((20975, 26), (8990, 26))
```



```
from sklearn.preprocessing import StandardScaler
sc= StandardScaler()
X_train[features]=sc.fit_transform(X_train[features]) # fit on training data columns and transform the training data columns
X_test[features]=sc.transform(X_test[features]) # transform the testing data columns
```

Initially we handled the imbalance target variable using S.M.O.T.E

The next step for model building is to split the dataset into train and test data

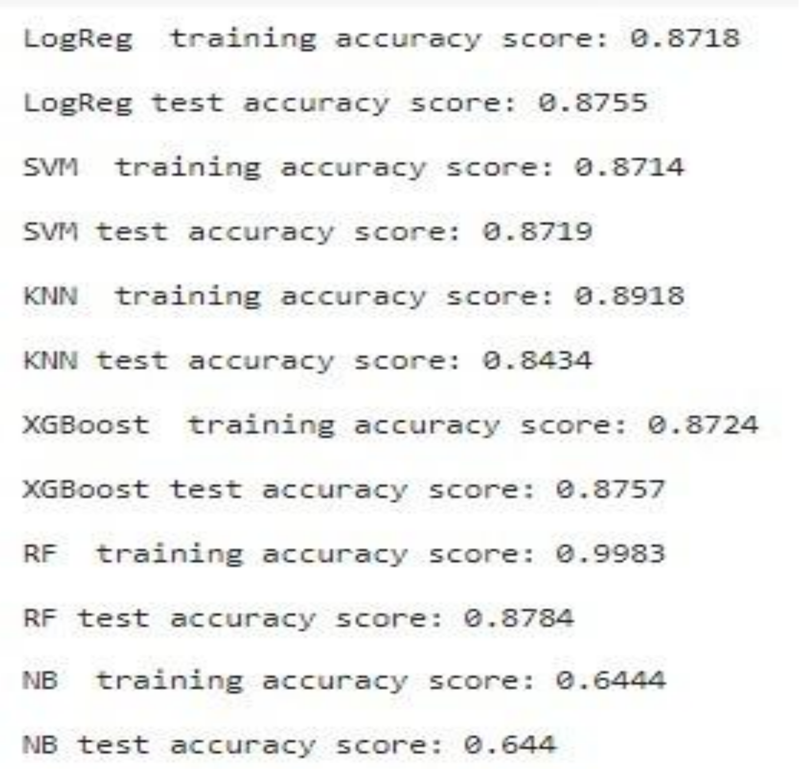
Then, we have performed Standard Scaling technique on the training set using fit.transform and thereafter transformed the test data

After splitting and scaling the dataset, we build a function to train the model using multiple supervised learning algorithms and classify the different models on the basis of their test - train accuracy.

Different Machine Learning Models Used

1. Logistic Regression
2. Gaussian Naive Bayes
3. Support Vector Machines
4. K-Nearest Neighbour
5. XGBoost
6. Random Forest

Baseline Model with Default parameters



```
LogReg  training accuracy score: 0.8718

LogReg test accuracy score: 0.8755

SVM  training accuracy score: 0.8714

SVM test accuracy score: 0.8719

KNN  training accuracy score: 0.8918

KNN test accuracy score: 0.8434

XGBoost  training accuracy score: 0.8724

XGBoost test accuracy score: 0.8757

RF  training accuracy score: 0.9983

RF test accuracy score: 0.8784

NB  training accuracy score: 0.6444

NB test accuracy score: 0.644
```

We can clearly classify our models based on above test accuracy score and the best baseline model is the XGBoost.

Random Forest and KNN models were overfitting using default parameters

Naive Bayes model gave us the least train and test accuracy comparatively.

Logistic Regression

Logistic Regression utilizes the power of regression to do classification One of the main reasons for the model’s success is its power of explainability.

After hyperparameter tuning, from this model we get the results as below:

- The accuracy on test data is **0.867**
- The precision on the test data is **0.902**
- The recall on the test data is **0.821**
- The f1 score on the test data is **0.860**
- The ROC score on the test data is **0.867**

Parameters:

{'C': 0.001,

'class_weight': {0: 10, 1: 15},

'penalty': 'l2'}

Gaussian Naive Bayes

Naive Bayes applies what is known as a posterior probability using Bayes Theorem to do the categorization on the unstructured data.

From this model we get the results as below:

- The accuracy on test data is **0.641**
- The precision on the test data is **0.851**
- The recall on the test data is **0.344**
- The f1 score on the test data is **0.490**
- The ROC score on the test data is **0.642**

Support Vector Machine

SVM algorithm creates the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category.

After hyperparameter tuning, from this model we get the results as below:

- The accuracy on test data is **0.875**
- The precision on the test data is **0.937**
- The recall on the test data is **0.805**
- The f1 score on the test data is **0.865**
- The ROC score on the test data is **0.875**

Parameters:

{'C': 100,

'gamma': 0.01,

'kernel': 'rbf'}

K Nearest Neighbor

K-Nearest Neighbor (KNN) algorithm predicts based on the specified number (k) of the nearest neighboring data points. Here, the pre-processing of the data is significant as it impacts the distance measurements directly.

After hyperparameter tuning, from this model we get the results as below:

- The accuracy on test data is **0.840**
- The precision on the test data is **0.889**
- The recall on the test data is **0.778**
- The f1 score on the test data is **0.830**
- The ROC score on the test data is **0.840**

Parameters:

{'n_neighbors': 15,

'weights': 'uniform'}

Random Forest

A Random Forest is a reliable ensemble of multiple Decision Trees (or CARTs); though more popular for classification, than regression applications

After hyperparameter tuning, from this model we get the results as below:

- The accuracy on test data is **0.868**
- The precision on the test data is **0.9168**
- The recall on the test data is **0.810**
- The f1 score on the test data is **0.860**
- The ROC score on the test data is **0.868**

Parameters:

(max_depth=10,

max_features=0.4,

min_samples_leaf=5,

n_estimators=800)

XGBoost

XGBoost is an implementation of gradient boosted decision trees designed for speed and performance.XGBoost works with large, complicated datasets. XGBoost is an ensemble modelling technique.

After hyperparameter tuning, from this model we get the results as below:

- The accuracy on test data is **0.874**
- The precision on the test data is **0.918**
- The recall on the test data is **0.822**
- The f1 score on the test data is **0.867**
- The ROC score on the test data is **0.874**

Parameters:

(gamma=1,

max_depth=10,

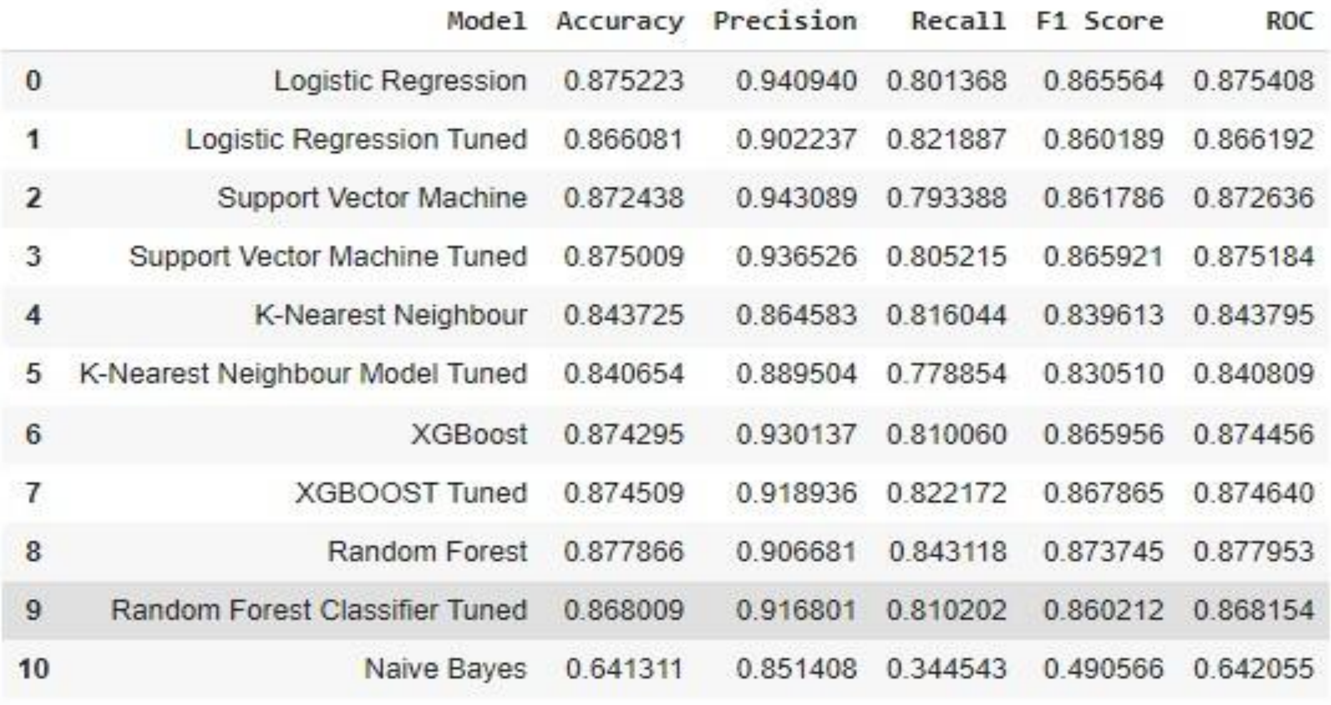
max_features='auto',

min_child_weight=20,

n_estimators=23,

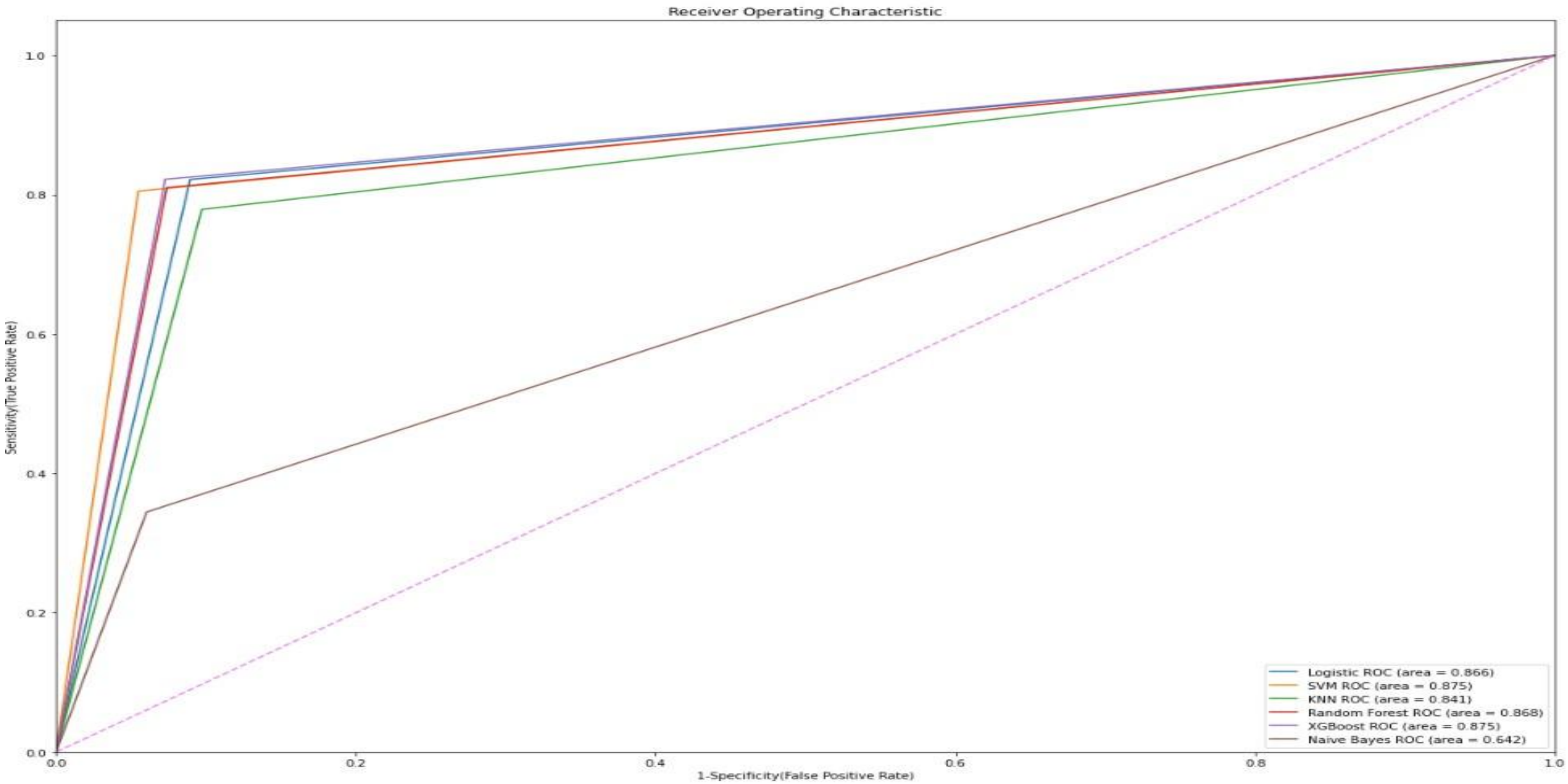
reg_alpha=0.1)

Performance Metrics Summary



	Model	Accuracy	Precision	Recall	F1 Score	ROC
0	Logistic Regression	0.875223	0.940940	0.801368	0.865564	0.875408
1	Logistic Regression Tuned	0.866081	0.902237	0.821887	0.860189	0.866192
2	Support Vector Machine	0.872438	0.943089	0.793388	0.861786	0.872636
3	Support Vector Machine Tuned	0.875009	0.936526	0.805215	0.865921	0.875184
4	K-Nearest Neighbour	0.843725	0.864583	0.816044	0.839613	0.843795
5	K-Nearest Neighbour Model Tuned	0.840654	0.889504	0.778854	0.830510	0.840809
6	XGBoost	0.874295	0.930137	0.810060	0.865956	0.874456
7	XGBOOST Tuned	0.874509	0.918936	0.822172	0.867865	0.874640
8	Random Forest	0.877866	0.906681	0.843118	0.873745	0.877953
9	Random Forest Classifier Tuned	0.868009	0.916801	0.810202	0.860212	0.868154
10	Naive Bayes	0.641311	0.851408	0.344543	0.490566	0.642055

ROC-AUC Curve



Conclusion

1. The maximum number of credit card holders in Taiwan were females and the average credit card limit provided by the credit card company to their respective customers was 167484.32(NT Dollars).

2. The most number of credit card holders were having university degree education and the most of the customers marriage status was Single, who carries a credit card in Taiwan.

3. The highest proportion of credit card holders were youth in the age of 29,thus we can conclude that mostly credit cards were popular among youths of taiwan than the older people.

4. The Correlation between features and target variable tells us the level of education and financial stability of the customers had high impact on the default rate.

5. The data also conveys us that the best indicator of delinquency is the behavior of the customer, which has been predominantly seen in the past couple of months payment repayment status .The heat map shows us the high correlation of payment repayment status with the target variable.

Conclusion

6. Comparatively after hyperparameter tuning the XGBoost Model comes out to be the best model in terms of its AUC_ROC score(0.875) and Recall score(0.82) and we can predict with 87.45% accuracy, whether a customer is likely to default next month.

7. The Second best model was the Support Vector Machine with a AUC_ROC score of 0.875 and a Recall score of 0.805 and we can predict with 87.5% accuracy, whether a customer is likely to default next month.

8. But it would be worth using Logistic Regression model for production since we do not just need a reliable model with good ROC_AUC Score but also a model that is quick and less complex.

9. Except Naive Bayes model,all the models have got really good ROC_AUC scores with a probability of 0.85 on an average.

10.The Random Forest and KNN models were really overfitting with default parameters and we handle the overfit in both these model by fine tuning the model.

11.Demographics: we see that being Female, More educated, Single and between 30-40 years old means a customer is more likely to make payments on time.

Challenges

- Extracting new features from existing features were a bit tedious job to do.
- Handling Imbalance data in Target features
- There were few undefined data records present in the dataset and few duplicate records
- The Hyper Parameter tuning using GridSearchCv was really time consuming task and required lots of patience until it get executed
- Selecting different parameter for hyperparameter tuning and finding the best parameters has to follow a trial n error technique,which is again a challenging job.