Credit Card Default Prediction

Bhargava Gaggainpali

DSC680 - Applied Data Science

bgaggainpali/bgaggainpali_DSC680 (github.com)

Business problem & Hypothesis

In financial industry, banks are playing important role in challenging times like now, with COVID pandemic across the globe. People are losing jobs and financial institutions are facing more delinquency rate on credit card loans. The increase in delinquency rate will result in significant financial loss to commercial banks. It is very critical for lending institutions like banks to have a prediction model to be able to predict customers for credit card default.

I have selected the topic, as I was interested in knowing the variables which influence the credit card default key factors. As I explore more about the domain, I understand that it's not same set of rules which is being used across domain and each different banks and credit unions are based on different credit score calculation structure when approving credit cards, but the factors which influence the default are same.

Solution Method

I see this problem as a classification issue, where we should try to understand and able to predict the customers, who have high Credit Card default chances. Planning to use supervised machine learning algorithm to work on the classification problem to be trained with algorithms like:

1. Logistic Regression
2. Decision Tree
3. Random Forest

Start with loading data into a data frame and then understand the data, then perform Exploratory Data Analysis (EDA) on the data set. EDA involves doing Univariate and

Bivariate Analysis, identify missing values and outliers and fill the gaps with appropriate values. In the next step, building model with starting from logistic regression and observe the accuracy of the model. When the accuracy of the of the model is not high, then planning to use Decision Tree and Random Forest to achieve higher accuracy.

Technical approach involves understanding the data by drawing multiple charts to observe the target variable with respect to each of the variable. Build a heat map to understand the relationship between variables. Build model using the different algorithms and observe the accuracy of the model, evaluate the accuracy of the model by building confusion matrix.

Data

I have identified UCI_Credit_Card.csv as source for my work, below is the Kaggle link. There are 30,000 observations in the dataset, each row in the dataset represents a credit card client. Given is the list of variables in the dataset.

Source File: https://www.kaggle.com/ainslie/credit-card-default-prediction-analysis

Variable	**Description**
ID	Credit Card ID - Sequence Number
LIMIT_BAL	Credit Limit
SEX	1 = male, 2 = female
EDUCATION	1 = graduate school, 2 = university, 3 = high school
MARRIAGE	1 = married, 2 = single, 3 = others
AGE	Customer Age
PAY_0	Repayment status September 2005

PAY_2	Repayment status August 2005
PAY_3	Repayment status July 2005
PAY_4	Repayment status June 2005
PAY_5	Repayment status May 2005
PAY_6	Repayment status April 2005
BILL_AMT1	Bill Amount September 2005
BILL_AMT2	Bill Amount August 2005
BILL_AMT3	Bill Amount July 2005
BILL_AMT4	Bill Amount June 2005
BILL_AMT5	Bill Amount May 2005
BILL_AMT6	Bill Amount April 2005
PAY_AMT1	Payment Amount September 2005
PAY_AMT2	Payment Amount August 2005
PAY_AMT3	Payment Amount July 2005
PAY_AMT4	Payment Amount June 2005
PAY_AMT5	Payment Amount May 2005
PAY_AMT6	Payment Amount April 2005
default.payment.next.month	1 = default, 0 = On time payment

Initial Observations

```
#Step 3:  Look at the sample data by taking first 5 rows
print(data.head(5))
```

```
   ID  LIMIT_BAL  SEX  EDUCATION  MARRIAGE  AGE  PAY_0  PAY_2  PAY_3  PAY_4  \
0   1    20000.0    2          2         1   24      2      2     -1     -1
1   2   120000.0    2          2         2   26     -1      2      0      0
2   3    90000.0    2          2         2   34      0      0      0      0
3   4    50000.0    2          2         1   37      0      0      0      0
4   5    50000.0    1          2         1   57     -1      0     -1      0

   ...  BILL_AMT4  BILL_AMT5  BILL_AMT6  PAY_AMT1  PAY_AMT2  PAY_AMT3  \
0  ...        0.0        0.0        0.0       0.0     689.0       0.0
1  ...     3272.0     3455.0     3261.0       0.0    1000.0    1000.0
2  ...    14331.0    14948.0    15549.0    1518.0    1500.0    1000.0
3  ...    28314.0    28959.0    29547.0    2000.0    2019.0    1200.0
4  ...    20940.0    19146.0    19131.0    2000.0   36681.0   10000.0

   PAY_AMT4  PAY_AMT5  PAY_AMT6  default.payment.next.month
0       0.0       0.0       0.0                           1
1    1000.0       0.0    2000.0                           1
2    1000.0    1000.0    5000.0                           0
3    1100.0    1069.0    1000.0                           0
4    9000.0     689.0     679.0                           0

[5 rows x 25 columns]
```

<u>Categorical Features:</u> Based on the data, below are categorical variables.

SEX

EDUCATION

MARRIAGE

default.payment.next.month 1 = default, 0 = On time payment

<u>Ordinal Features:</u> Based on the data with inherent hierarchy, below are ordinal variables.

AGE

PAY_0, PAY_2, PAY_3, PAY_4, PAY_5 & PAY_6

Numerical Features: Based on the numerical data, below are numerical variables.

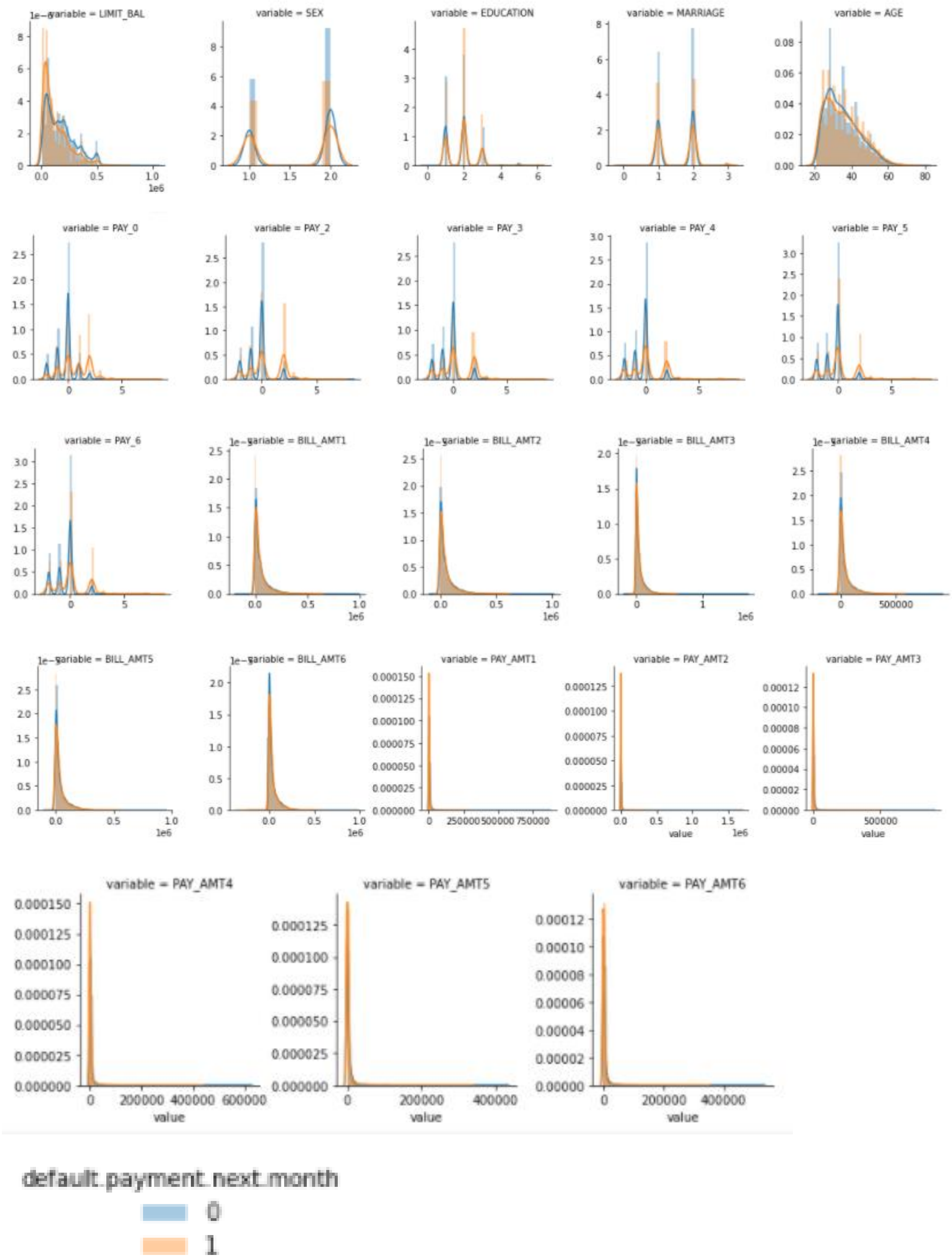
BILL_AMT1, BILL_AMT2, BILL_AMT3, BILL_AMT4, BILL_AMT5 & BILL_AMT6

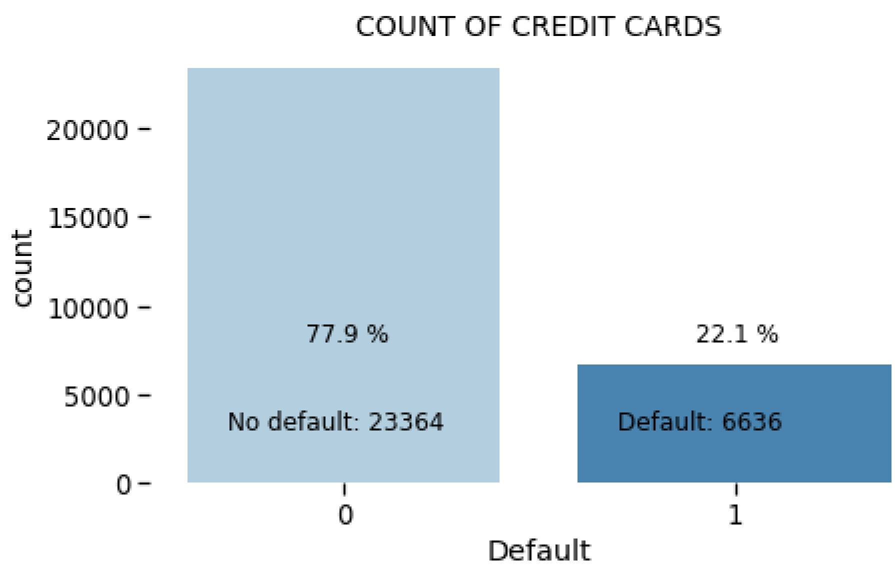
PAY_AMT1, PAY_AMT2, PAY_AMT3, PAY_AMT4, PAY_AMT5 & PAY_AMT6

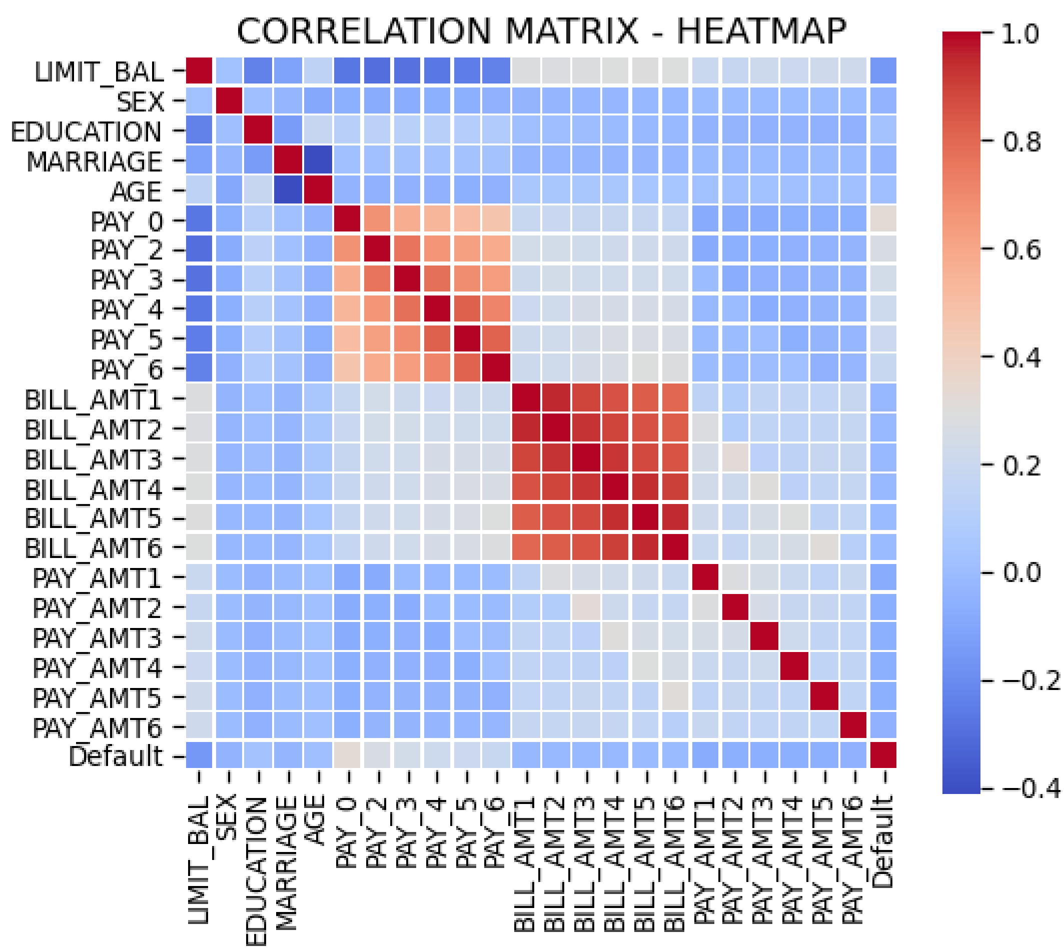
Exploratory Data Analysis

After initial analysis of looking at the dataset values and the basic stats, I had to change my focus on considering many factors. Initially was under the impression that, Credit Card Default depends on Limit_Balance, Education, Marriage, Pay months and limited factors. I saw surprising stats when I used visualizations to give clear idea on how each factor has its effect on the Credit Card Default. I had to increase my research questions to explore and include more variables, than initially prepared. Its based on the initial analysis using visualization.

```
Describe Data
                 ID        LIMIT_BAL           SEX     EDUCATION      MARRIAGE  \
count  30000.000000     30000.000000  30000.000000  30000.000000  30000.000000
mean   15000.500000    167484.322667      1.603733      1.853133      1.551867
std     8660.398374    129747.661567      0.489129      0.790349      0.521970
min        1.000000     10000.000000      1.000000      0.000000      0.000000
25%     7500.750000     50000.000000      1.000000      1.000000      1.000000
50%    15000.500000    140000.000000      2.000000      2.000000      2.000000
75%    22500.250000    240000.000000      2.000000      2.000000      2.000000
max    30000.000000   1000000.000000      2.000000      6.000000      3.000000

                AGE         PAY_0         PAY_2         PAY_3         PAY_4  \
count  30000.000000  30000.000000  30000.000000  30000.000000  30000.000000
mean      35.485500     -0.016700     -0.133767     -0.166200     -0.220667
std        9.217904      1.123802      1.197186      1.196868      1.169139
min       21.000000     -2.000000     -2.000000     -2.000000     -2.000000
25%       28.000000     -1.000000     -1.000000     -1.000000     -1.000000
50%       34.000000      0.000000      0.000000      0.000000      0.000000
75%       41.000000      0.000000      0.000000      0.000000      0.000000
max       79.000000      8.000000      8.000000      8.000000      8.000000

           PAY_AMT2      PAY_AMT3       PAY_AMT4       PAY_AMT5  \
count  3.000000e+04   30000.00000   30000.000000   30000.000000
mean   5.921163e+03    5225.68150    4826.076867    4799.387633
std    2.304087e+04   17606.96147   15666.159744   15278.305679
min    0.000000e+00       0.00000       0.000000       0.000000
25%    8.330000e+02     390.00000     296.000000     252.500000
50%    2.009000e+03    1800.00000    1500.000000    1500.000000
75%    5.000000e+03    4505.00000    4013.250000    4031.500000
max    1.684259e+06  896040.00000  621000.000000  426529.000000

            PAY_AMT6  default.payment.next.month
count   30000.000000                30000.000000
mean     5215.502567                    0.221200
std     17777.465775                    0.415062
min         0.000000                    0.000000
25%       117.750000                    0.000000
50%      1500.000000                    0.000000
75%      4000.000000                    0.000000
max    528666.000000                    1.000000
```


variable = LIMIT_BAL
variable = SEX
variable = EDUCATION
variable = MARRIAGE
variable = AGE
variable = PAY_0
variable = PAY_2
variable = PAY_3
variable = PAY_4
variable = PAY_5
variable = PAY_6
variable = BILL_AMT1
variable = BILL_AMT2
variable = BILL_AMT3
variable = BILL_AMT4
variable = BILL_AMT5
variable = BILL_AMT6
variable = PAY_AMT1
variable = PAY_AMT2
variable = PAY_AMT3
variable = PAY_AMT4
variable = PAY_AMT5
variable = PAY_AMT6
value
default.payment.next.month
0
1





Few observations based on the above plots.

1. Customers with low LIMIT_BAL have higher Default rate.
2. Default rate low among Females(Sex=2).
3. Customers with highly educated are less like to default (EDUCATION=1 or 2).
4. Customers with Marital status single are less like to default (MARRIAGE=2).
5. People in the age group 30-40 years are less likely to default.

Data Preparation

Applied MinMax Scaler to scale the numeric data variables.

```
# applying MinMax Scaler to numerical variables
scaler=MinMaxScaler()
scaler.fit(df_Nums)
# Transform Scaled data
df_Nums=scaler.transform(df_Nums)
# Convert the data to DataFrame
df_Nums = pd.DataFrame(df_Nums)
df_Nums.columns = ['AGE','PAY_0','BILL_AMT1','PAY_AMT1']
df_Nums.head()
```

	AGE	PAY_0	BILL_AMT1	PAY_AMT1
0	0.051724	0.4	0.149982	0.000000
1	0.086207	0.1	0.148892	0.000000
2	0.224138	0.2	0.172392	0.001738
3	0.275862	0.2	0.188100	0.002290
4	0.620690	0.1	0.154144	0.002290

Applied One Hot Encoding to convert Categorical data to Numerical data variables.

```
# convert the Categorical data to Numerical data
df_Catg = df_Catg.replace({'SEX': {1: 'male', 2: 'female'}})
df_Catg = df_Catg.replace({'EDUCATION': {1: 'graduate school', 2: 'university' ,3:'high school', 4:'others'}})
df_Catg = df_Catg.replace({'MARRIAGE': {1: 'married', 2: 'single', 3:'others'}})
# One Hot Encoding
df_Catg = pd.get_dummies(df_Catg)

# check the data
df_Catg.head()
```

	SEX_female	SEX_male	EDUCATION_graduate school	EDUCATION_high school	EDUCATION_others	EDUCATION_university	MARRIAGE_married	MARRIAGE_others	MARR
0	1	0	0	0	0	1	1	0	
1	1	0	0	0	0	1	0	0	
2	1	0	0	0	0	1	0	0	
3	1	0	0	0	0	1	1	0	
4	0	1	0	0	0	1	1	0	

Applied PCA to reduce the number of Dimensions or Variables.

```
# redusing the number of companents to 4 using PCA
pca=PCA(n_components=4)
pca.fit(df_final_data)
# Transform the data after applying PCA
df_final_data_PCA = pca.transform(df_final_data)
print('Number of elements in the data frame after applying PCA ')
df_final_data_PCA.shape
```

```
Number of elements in the data frame after applying PCA

(30000, 4)
```

```
# Display the input data which is converted to 4 components using PCA
df_final_data_PCA = pd.DataFrame(df_final_data_PCA)
df_final_data_PCA.columns = ['PCA_Comp_1','PCA_Comp_2','PCA_Comp_3','PCA_Comp_4']
df_final_data_PCA.head()
```

	PCA_Comp_1	PCA_Comp_2	PCA_Comp_3	PCA_Comp_4
0	-1.038423	-0.268148	-0.312136	-0.248259
1	0.172907	-0.710107	-0.812939	-0.026458
2	0.160931	-0.700891	-0.807798	-0.017851
3	-1.056963	-0.256821	-0.295211	-0.242089
4	-0.630666	1.097638	-0.363249	-0.246180

Model Development

As part of the current project, four models were developed after data preparation steps. Data is split in the ratio of 70:30 for train and test, i.e. 70% of the data is fed to the model to understand the patterns and remembering the outcome, later 30% of the data is used to validate the prediction results.

```
No. of samples in training set:  21000
No. of samples in validation set: 9000


No. of default and not-defaultes in the training set:
0    16396
1     4604
Name: Default, dtype: int64


No. of default and not-defaulted in the validation set:
0    6968
1    2032
Name: Default, dtype: int64
```

Below are four models

- ***Logistic Regression***

```
#--------------
# Logistic Regression
#--------------
from sklearn.linear_model import LogisticRegression
classifier2 = LogisticRegression()
classifier2.fit( X_train, y_train )
y_pred = classifier2.predict( X_val )

cm = confusion_matrix( y_val, y_pred )
print("Accuracy on Test Set for LogReg = %.2f" % ((cm[0,0] + cm[1,1] )/len(X_val)))
scoresLR = cross_val_score( classifier2, X_train, y_train, cv=10)
print("Mean LogReg CrossVal Accuracy on Train Set %.2f, with std=%.2f" % (scoresLR.mean(), scoresLR.std() ))
```

```
Accuracy on Test Set for LogReg = 0.80
```

- ***Kernel SVM Model***

```
# kernel SVM  Model

from sklearn.svm import SVC
classifier_svm = SVC(kernel="rbf")
classifier_svm.fit( X_train, y_train )
y_pred = classifier_svm.predict( X_val )

cm = confusion_matrix( y_val, y_pred )
print("Accuracy on Test Set for kernel-SVM = %.2f" % ((cm[0,0] + cm[1,1] )/len(X_val)))
scoresSVC = cross_val_score( classifier_svm, X_train, y_train, cv=10)
print("Mean kernel-SVM CrossVal Accuracy on Train Set %.2f, with std=%.2f" % (scoresSVC.mean(), scoresSVC.std() ))
```

```
Accuracy on Test Set for kernel-SVM = 0.78
Mean kernel-SVM CrossVal Accuracy on Train Set 0.79, with std=0.00
```

- ***Naïve Bayes***

```
#--------------
# Naive Bayes
#--------------
from sklearn.naive_bayes import GaussianNB
classifier3 = GaussianNB()
classifier3.fit( X_train, y_train )
y_pred = classifier3.predict( X_val )
cm = confusion_matrix( y_val, y_pred )
print("Accuracy on Test Set for NBClassifier = %.2f" % ((cm[0,0] + cm[1,1] )/len(X_val)))
scoresNB = cross_val_score( classifier3, X_train, y_train, cv=10)
print("Mean NaiveBayes CrossVal Accuracy on Train Set %.2f, with std=%.2f" % (scoresNB.mean(), scoresNB.std() ))
```

```
Accuracy on Test Set for NBClassifier = 0.75
Mean NaiveBayes CrossVal Accuracy on Train Set 0.75, with std=0.02
```

- ***KNeighborsClassifier***

```
#--------------
# K-NEIGHBOURS
#--------------
from sklearn.neighbors import KNeighborsClassifier
classifier4 = KNeighborsClassifier(n_neighbors=5)
classifier4.fit( X_train, y_train )
y_pred = classifier4.predict( X_val )
cm = confusion_matrix( y_val, y_pred )
print("Accuracy on Test Set for KNeighborsClassifier = %.2f" % ((cm[0,0] + cm[1,1] )/len(X_val)))
scoresKN = cross_val_score( classifier3, X_train, y_train, cv=10)
print("Mean KN CrossVal Accuracy on Train Set Set %.2f, with std=%.2f" % (scoresKN.mean(), scoresKN.std() ))
```

```
Accuracy on Test Set for KNeighborsClassifier = 0.79
Mean KN CrossVal Accuracy on Train Set Set 0.75, with std=0.02
```

Testing and Evaluation

After completing Model building using different algorithms, evaluate the accuracy of the model by building confusion matrix. As part of this project confusion matrix is built for each of the models, below is confusion matrix built on KNeighborsClassifier Model.

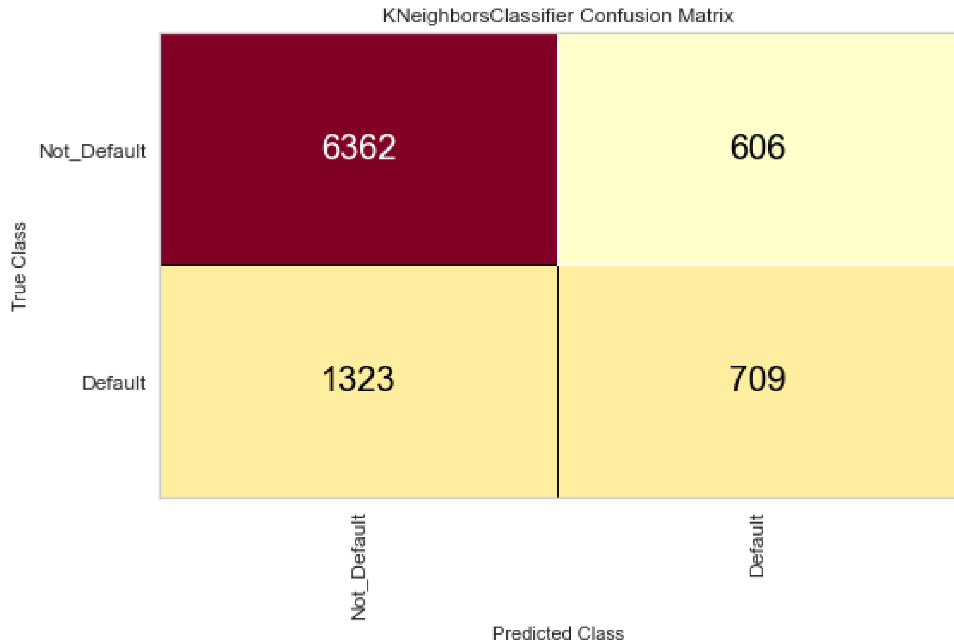


Chart shows the good precision and recall values and f1score 0.868 for not-Default cases indicates that model is performing as expected.

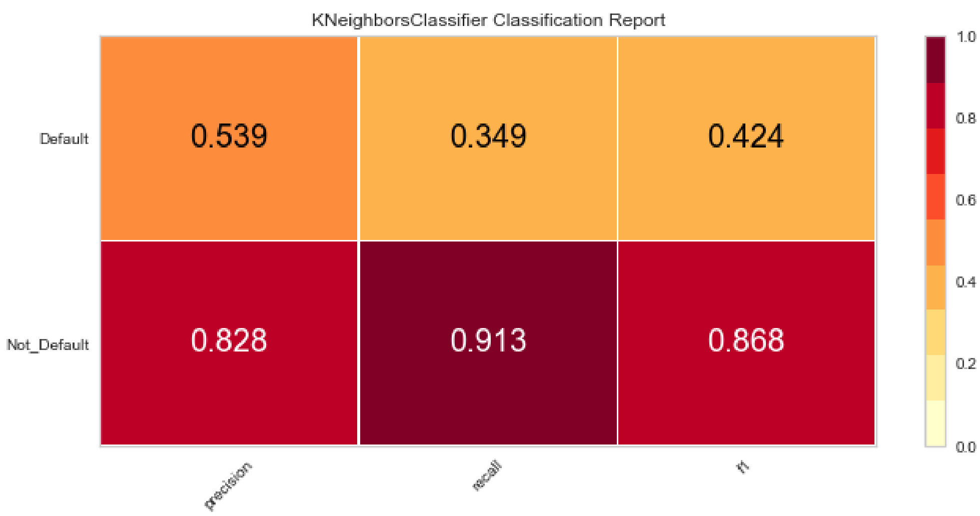
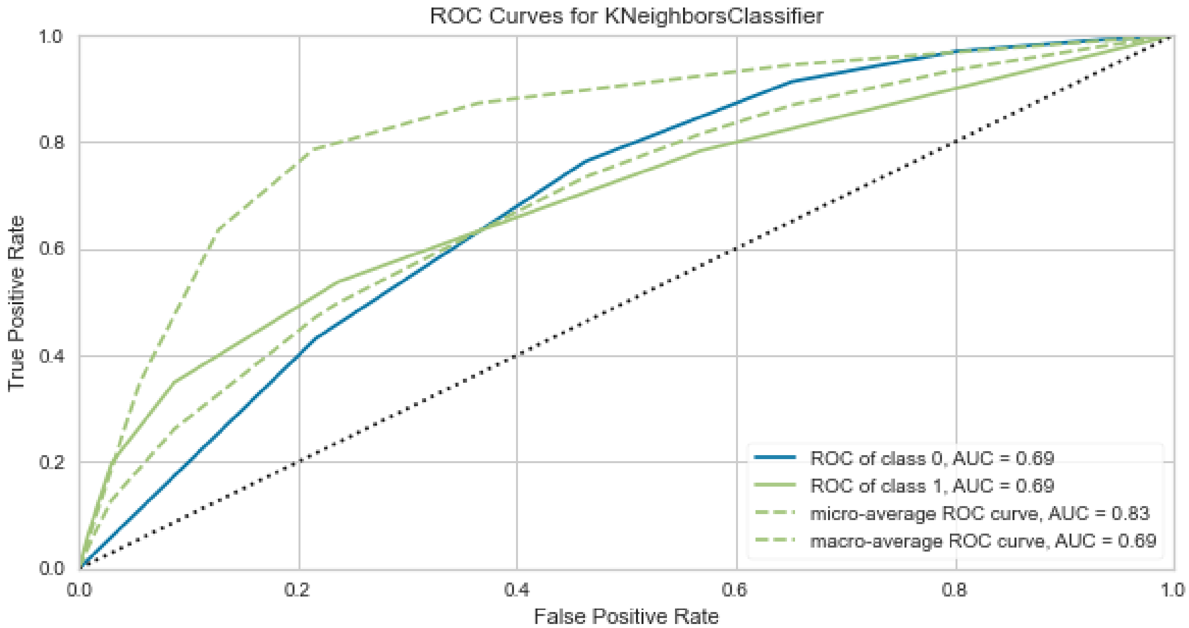
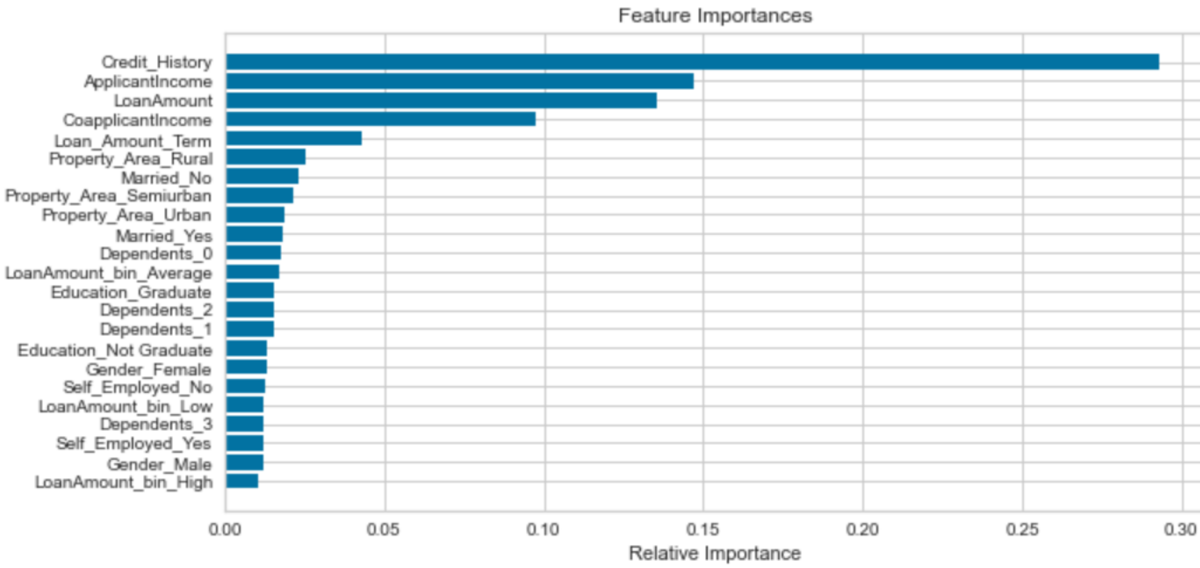


Chart shows the AUC (Area Under Curve) is at a value of 0.69 indicates that model is performing as expected.



Conclusion

Below chart shows that Bill Amount is most important factor in deciding the Credit Card default followed by Payment Amount and Age.



Future Analysis Questions

I would like to continue my analysis and try to explore further to find answers to the given questions.

1. With the COVID situation in place, are these features still valid to predict the Credit Card Default.
2. As with the different type of defaults like Credit Card, Car Loan, Home Loan, would these features stay in common across industry.
3. Will the feature importance change with respect to geographic location?

Reference:

1. LATOYA IRBY - February 10, 2020 - What You Can Do About Credit Card Default

What You Can Do About Credit Card Default (thebalance.com)

2. Jenny Wang - Jun 24, 2020 - Will You Be Able to Make Your Credit Card Payment?

Will You Be Able to Make Your Credit Card Payment? | by Jenny Wang | Towards Data Science

3. Marcos Dominguez - Feb 26,2021- Predicting Credit Card Defaults with Machine Learning

Predicting Credit Card Defaults with Machine Learning | by Marcos Dominguez | The Startup | Feb, 2021 | Medium

4. Yashna Sayjadah, Ibrahim Abaker Targio Hashem, Faiz Alotaibi, Khairl Azhar Kasmiran - October 2018 - Credit Card Default Prediction using Machine Learning Techniques

(PDF) Credit Card Default Prediction using Machine Learning Techniques (researchgate.net)

5. Bank Rate – 2021 - Credit card default: How it happens, what to do about it

Credit Card Default: What to Do About It | Bankrate.com

6. Equifax – 2021 - What Happens If I Default on a Loan or Credit Card Debt?

Process & Potential Effects of Defaulting on a Loan | Equifax