Classification Analysis on Credit Card Default Prediction

Kanike Lakshmi Narayana
Data science Trainee at
AlmaBetter

Abstract

In recent years, credit card issuers in Taiwan faced the cash and credit card debt crisis and delinquency is expected to peak in the third quarter of 2006. To increase market share, card-issuing banks in Taiwan over-issued cash and credit cards to unqualified applicants. At the same time, most cardholders, irrespective of their repayment ability, overused credit cards for consumption and accumulated heavy credit and cash–card debts. The crisis caused a blow to consumer finance confidence and it is a big challenge for both banks and cardholders.

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.

Introduction

Credit card companies make money by collecting fees. Out of the various fees, interest charges are the primary source of revenue. When credit card users fail to pay off their bills at the end of the month, the bank is allowed to charge interest on the borrowed amount. Other fees, such as annual fees and late fees, also contribute, though to a lesser extent. Another major source of income for credit card companies are fees collected from merchants who accept card payments.

Through the fees they get to collect, banks make a profit on their credit card business.

Machine Learning is concerned with computer programs that automatically improve their performance through experience. In machine learning, we have supervised learning, unsupervised learning and reinforcement learning. Again the supervised learning is further divided into regression and classification. In this project, we are going to look after the supervised learning classification model.

The Classification model is used when we have to predict something from the discrete-valued output.

Objective

The main objective is to build a model that can predict the customers who are going to default their credit card due for the next month.

Dataset Peeping

Owing to the size of the dataset, extensive cleaning of the dataset is not needed. The following steps are performed for the analysis purpose:

- In the dataset, the columns are not properly named, the names are recorded in the first row of the sheet.
- The columns are renamed and the row containing the names of the columns is dropped and are made the dataset to the actual size.
- Dataset was clean and does not contain any NaN values.

Data Description

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:

- X1: Amount of the given credit (NT dollar): it includes both the individual consumer credit and his/her family (supplementary) credit.
- X2: Gender (1 = male; 2 = female).
- X3: Education (1 = graduate school; 2 = university; 3 = high school; 4 = others).
- X4: Marital status (1 = married; 2 = single; 3 = others).
- X5: Age (year).
- X6 - X11: 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.
- X12-X17: 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.
- X18-X23: Amount of previous payment (NT dollar). X18 = amount paid in September, 2005; X19 = amount paid in August, 2005; . . .; X23 = amount paid in April, 2005.

Challenges Faced

The following are the challenges faced in the data analysis:

- Renaming the columns to the actual names.
- Converting the data type of features.

Approach

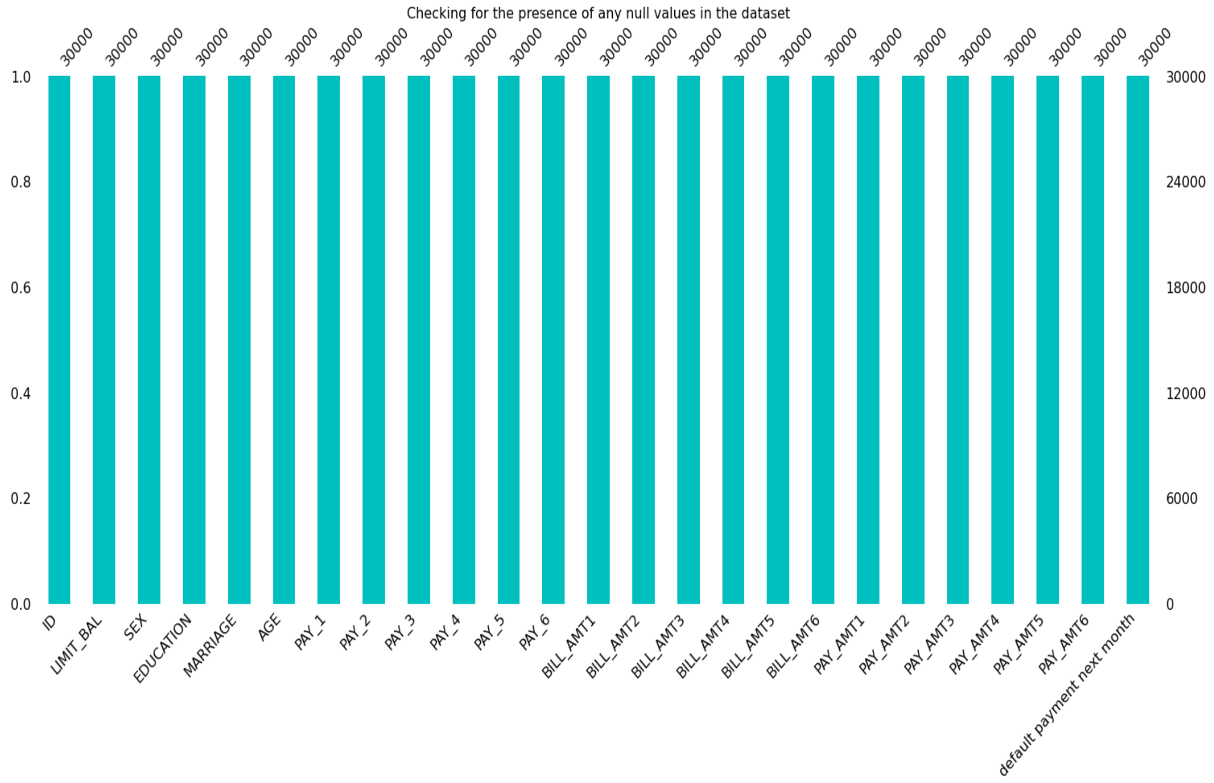
As the problem statement says the main objective is to predict the defaulters of the credit card due and the dependent variable is discrete data. I have used the supervised learning classification analysis Logistic Regression, Random Forest Classifier, K Near Neighbour, Naive Bayes Classifier to train the model to predict the defaulters.

Tools Used

The whole project was done using python, in google collaboratory. Following libraries were used for analysing the data and visualizing:

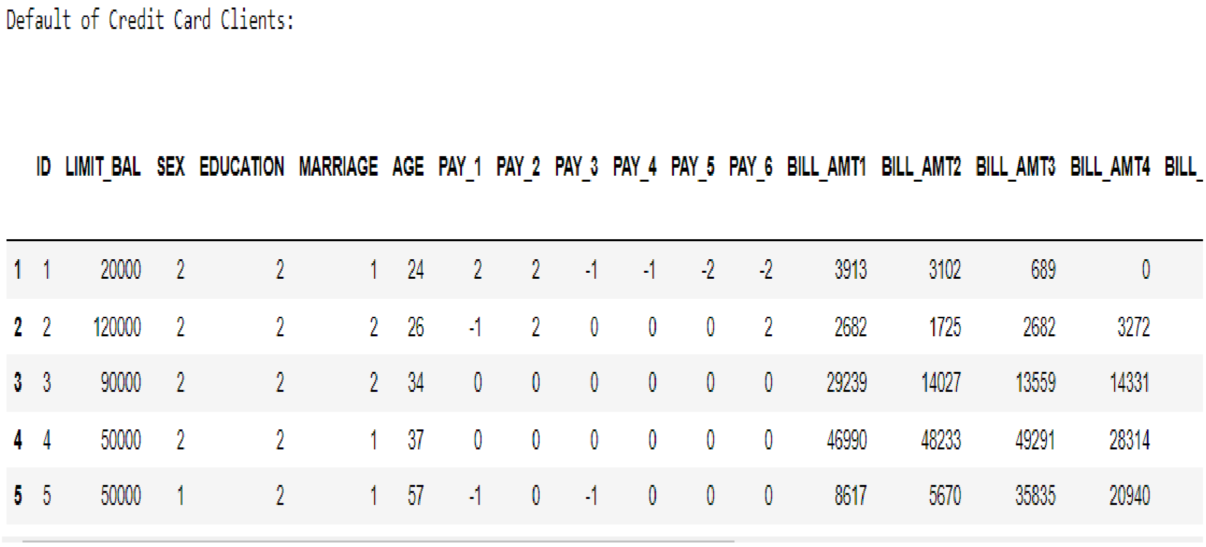
- Pandas: Extensively used to load and wrangle with the dataset.
- Matplotlib: Used for visualization.
- Seaborn: Used for visualization.
- Warnings: For filtering and ignoring warnings.
- Numpy: For some math operations in predictions.
- Sklearn: For analysis and prediction.
- Stats models: For outliers influence.

Visualizing the presence of NaN values



The above figure shows that there are no NaN (Not a Number) values in the given dataset.

Pandas DataFrame



Default of Credit Card Clients:

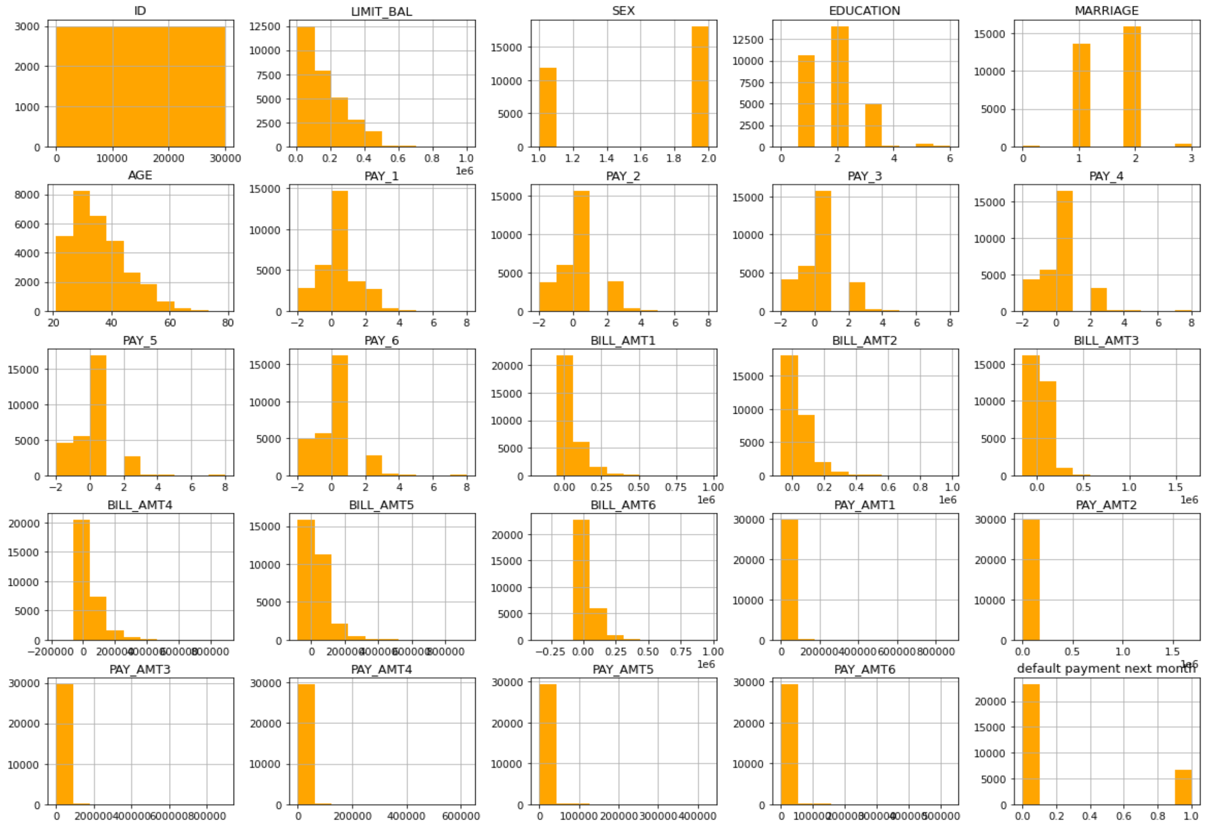
	ID	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	PAY_1	PAY_2	PAY_3	PAY_4	PAY_5	PAY_6	BILL_AMT1	BILL_AMT2	BILL_AMT3	BILL_AMT4	BILL_
1	1	20000	2	2	1	24	2	2	-1	-1	-2	-2	3913	3102	689	0	
2	2	120000	2	2	2	26	-1	2	0	0	0	2	2682	1725	2682	3272	
3	3	90000	2	2	2	34	0	0	0	0	0	0	29239	14027	13559	14331	
4	4	50000	2	2	1	37	0	0	0	0	0	0	46990	48233	49291	28314	
5	5	50000	1	2	1	57	-1	0	-1	0	0	0	8617	5670	35835	20940	

The table shows the dataset in the form of Pandas DataFrame.
The dataset has 8760 rows and 14 columns wholly the shape of (8760, 14).
It contains the following columns:

- Id
- Limit Balance
- Sex
- Education
- Marriage
- Age
- Pay_1 - repayment status of September
- Pay_2 - repayment status of August
- Pay_3 - repayment status of July
- Pay_4 - repayment status of June
- Pay_5 - repayment status of May
- Pay_6 - repayment status of April
- Bill_Amt1 - repayment amount of September
- Bill_Amt2 - repayment amount of August
- Bill_Amt3 - repayment amount of July
- Bill_Amt4 - repayment amount of June
- Bill_Amt5 - repayment amount of May
- Bill_Amt6 - repayment amount of April
- Pay_Amt1 - the amount paid on September
- Pay_Amt2 - the amount paid on August
- Pay_Amt3 - the amount paid on July
- Pay_Amt4 - the amount paid on June
- Pay_Amt5 - the amount paid on May
- Pay_Amt6 - the amount paid on April

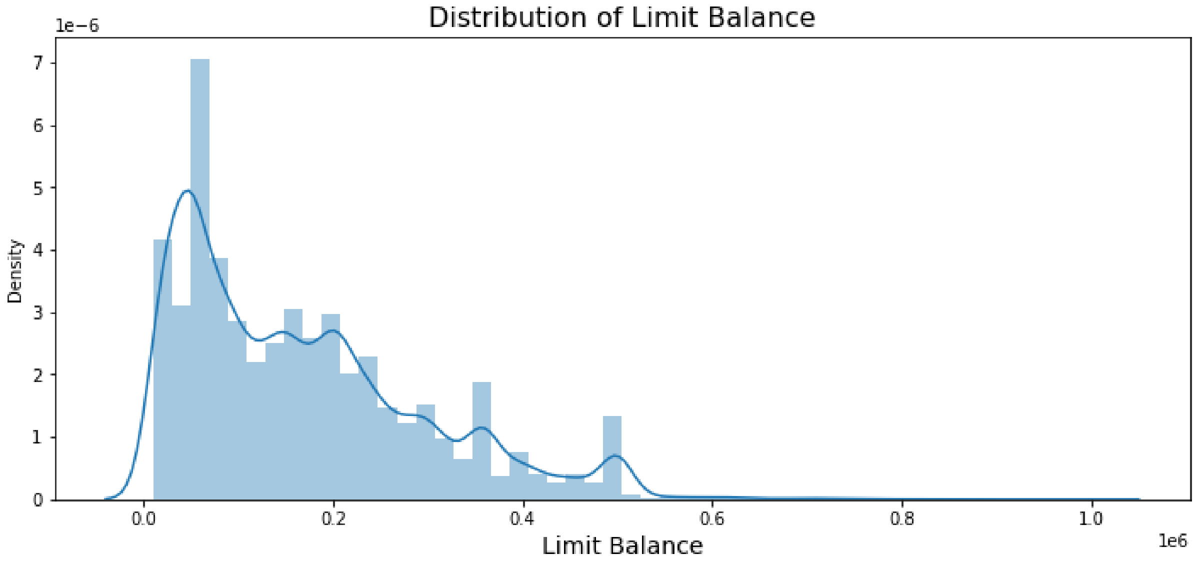
- Default payment next month

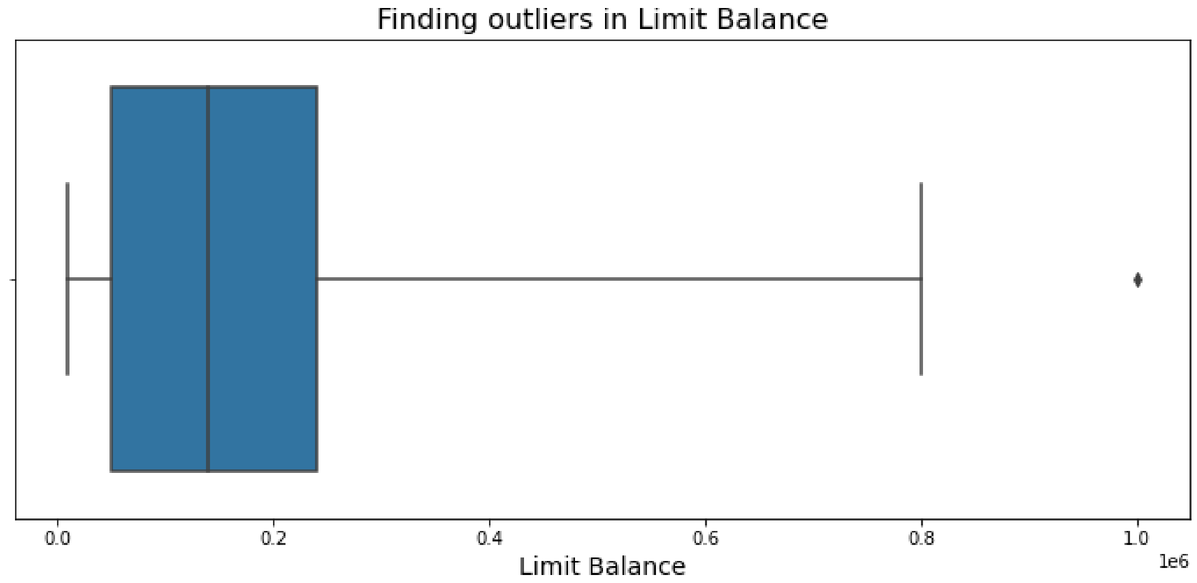
Histogram representation of the data



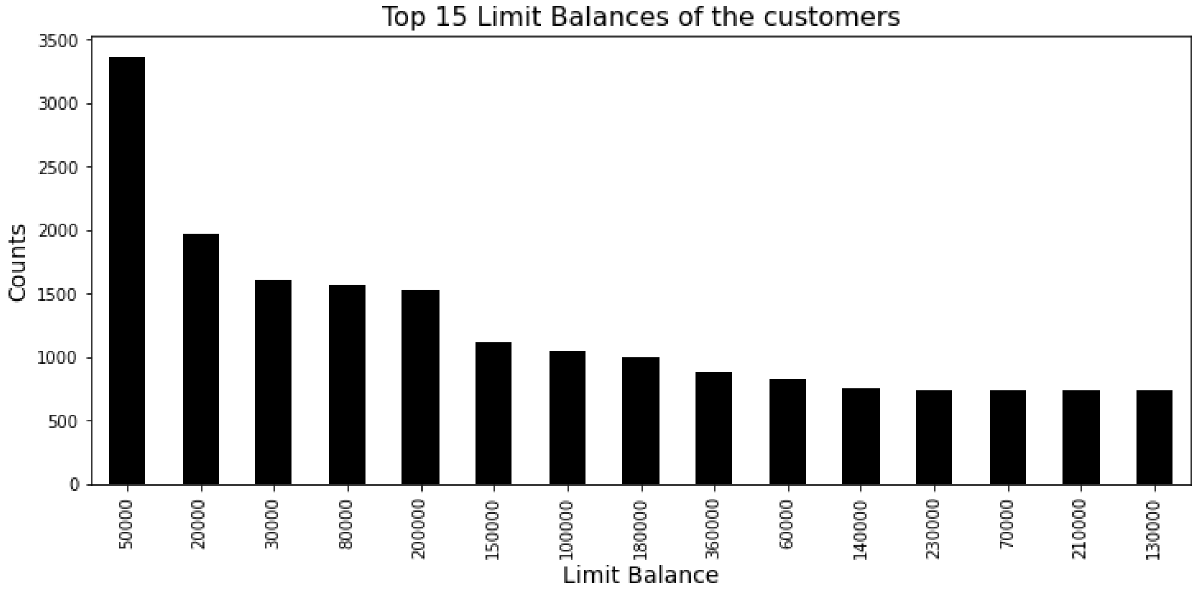
The above figure depicts the distribution of all the columns of the dataset in a separate histogram and shows the density of such columns.

Distribution of Limit Balance

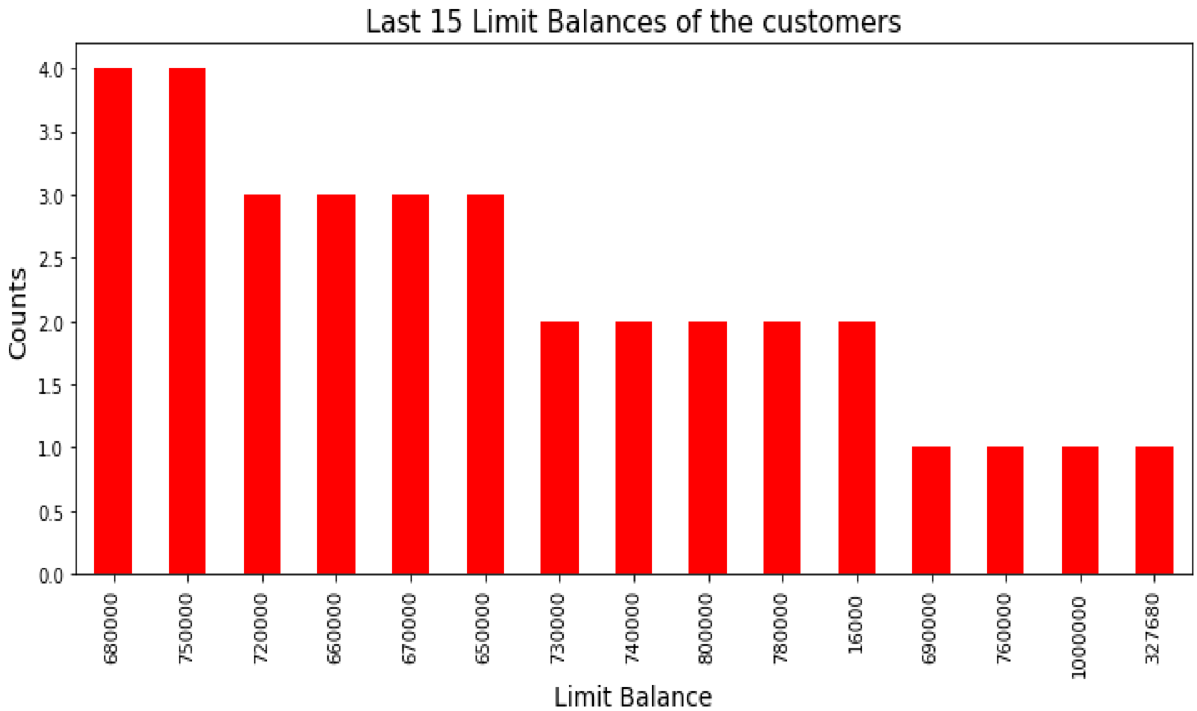




The boxplot shows that there are no outliers in the Limit Balance feature.

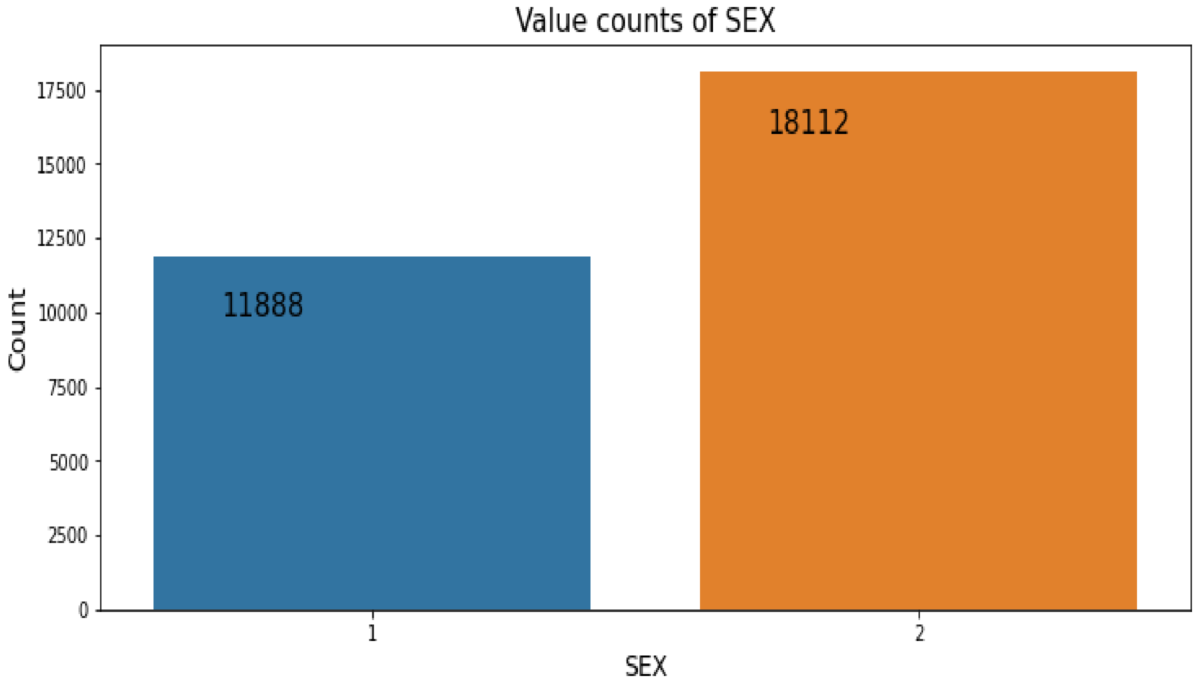


From the above figure, we can say that these listed limit balance customers are the priority customers who are having high credit limit balances.

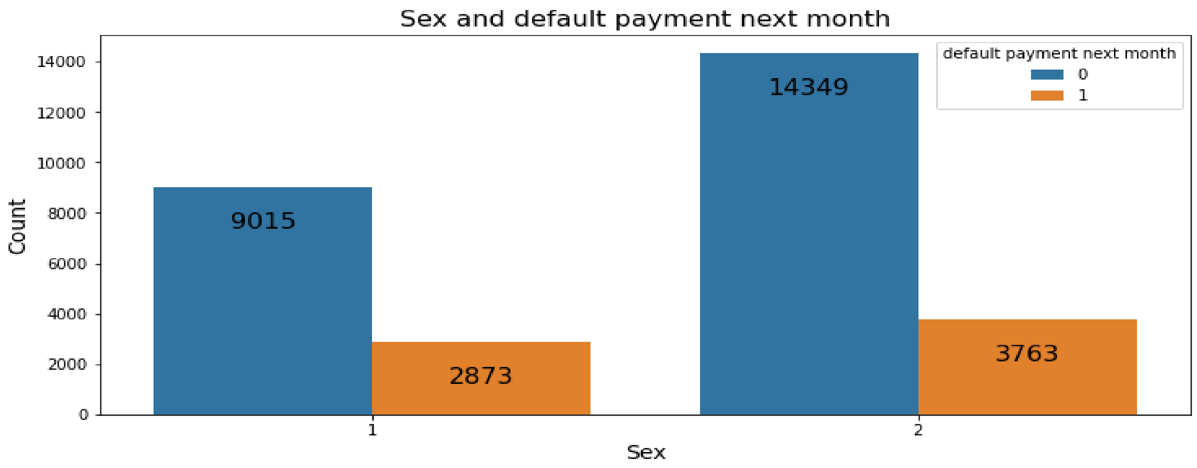


The above figures show the 15 low credit limit balance customers of the company.

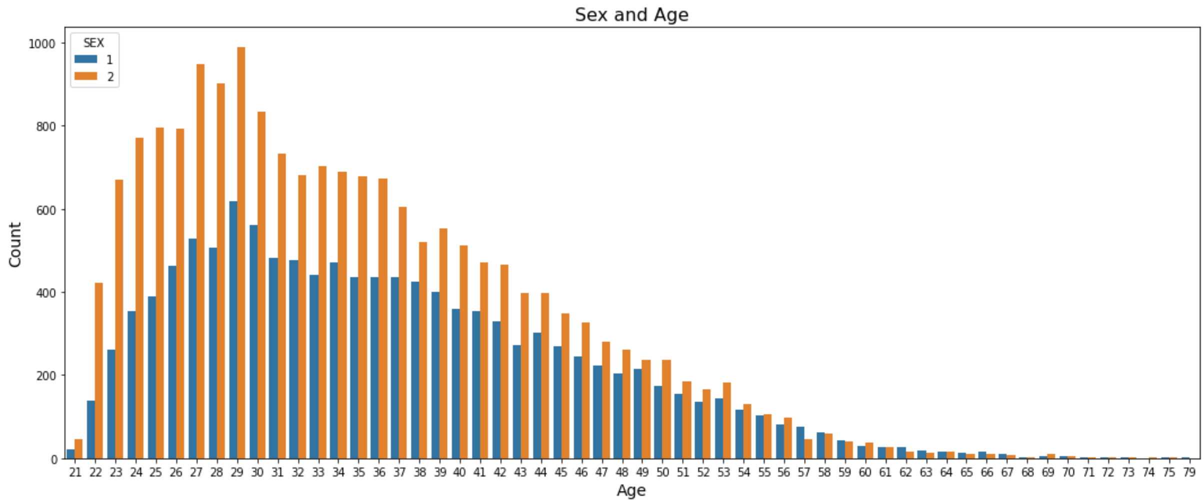
Analysis of Sex



The dataset is containing 40:60 as male and female sex ratio in customers.

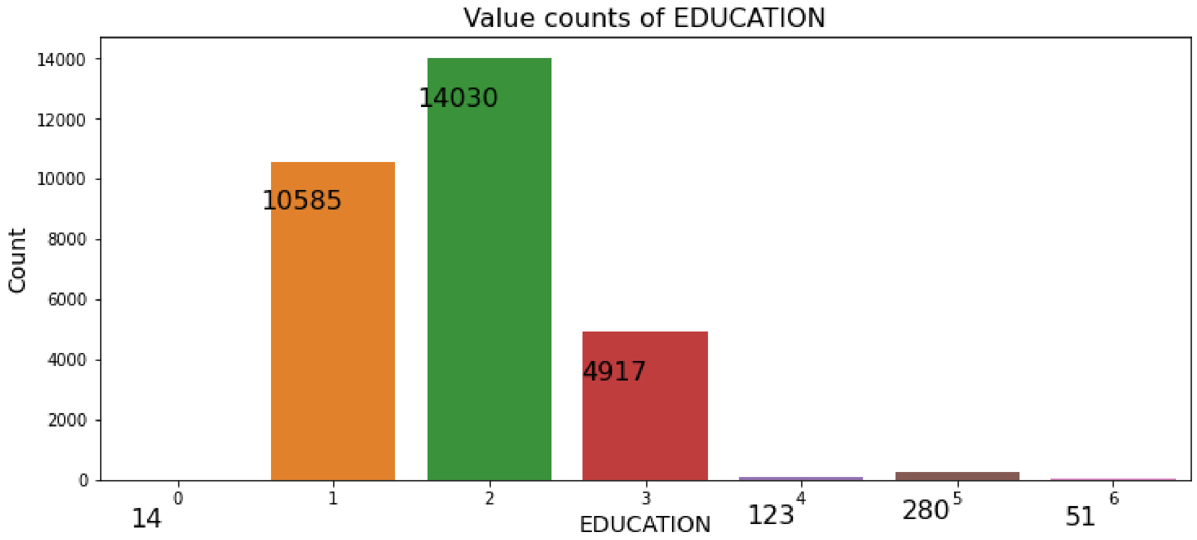


It clearly shows that the female customers are more in number as well as in default the credit due also.

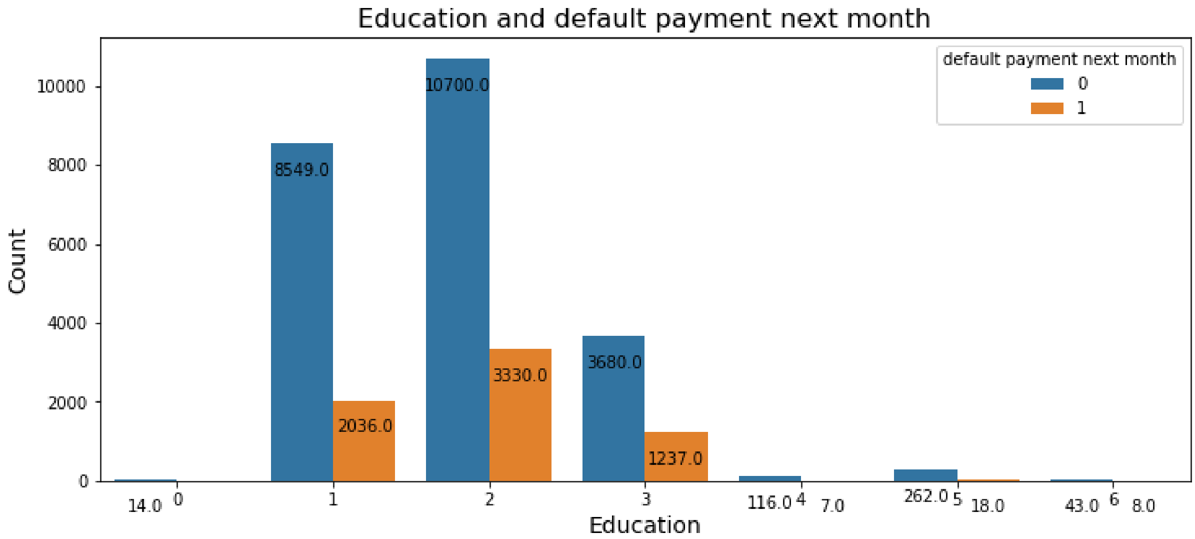


We have more customers in the 25 to 30 age group in both male and female.

Analysis of Education

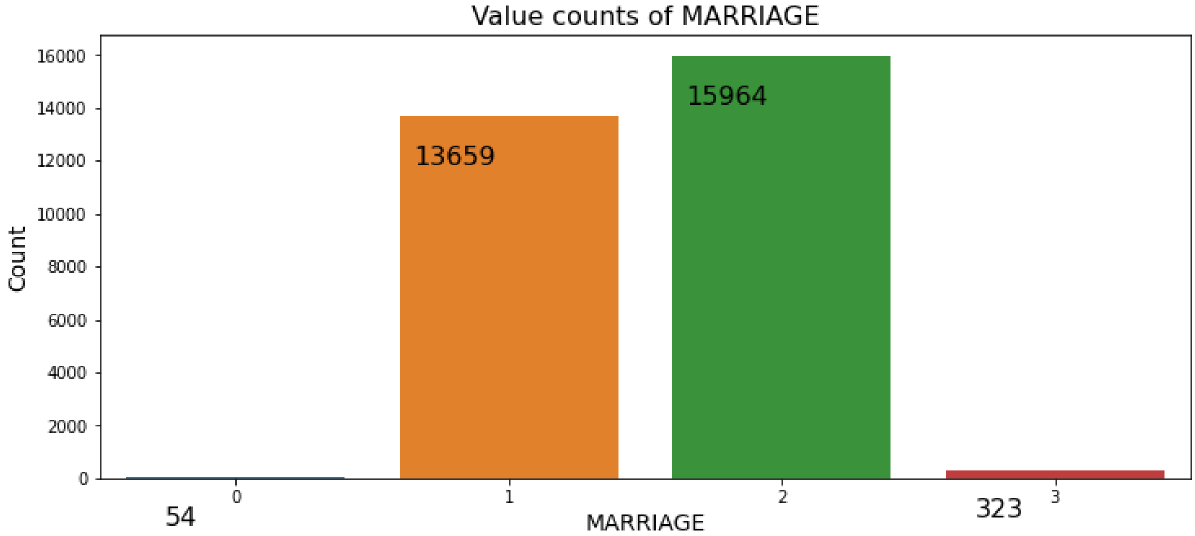


In the above figure 1 means graduate, 2 means university, 3 means high school, and the remaining all refers to others. We have more customers from the university.

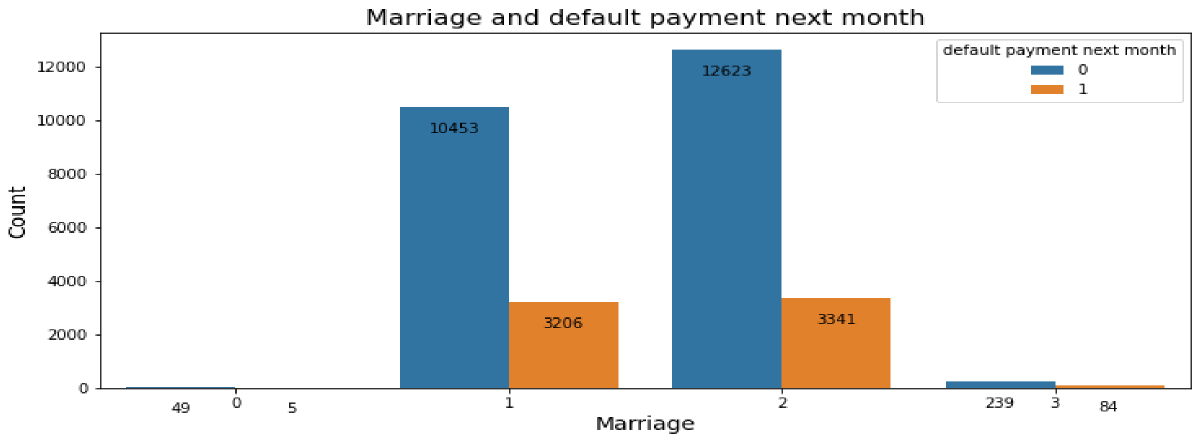


Most of the defaulters are from the university and secondly lies in graduate customers.

Analysis of Marriage

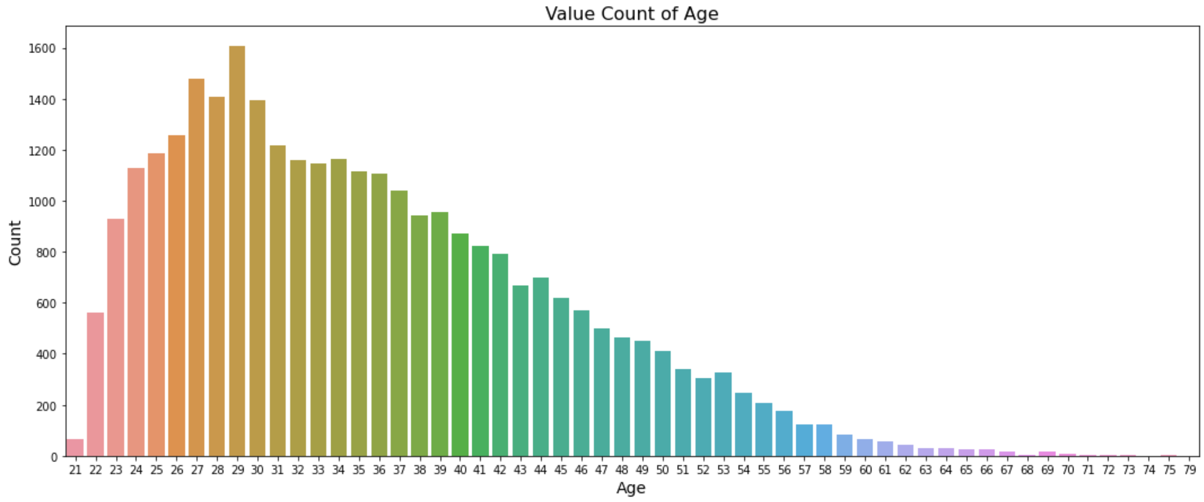


The above data shows 1 refers to married 2 refers to single and remaining refers to others. Most of the customers are single.

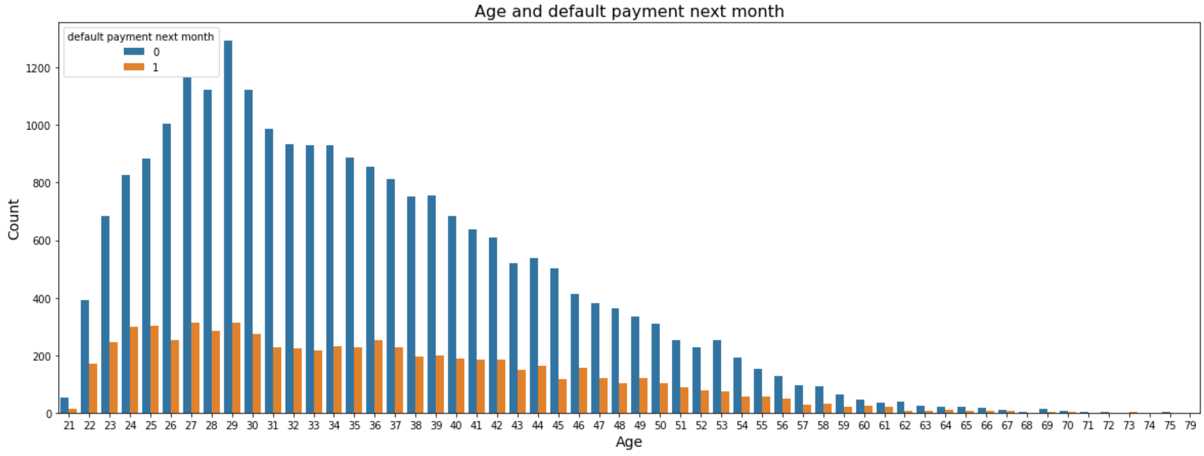


Defaulters of bill due are from both of the categories married and singles are mostly equal in number.

Analysis of Age

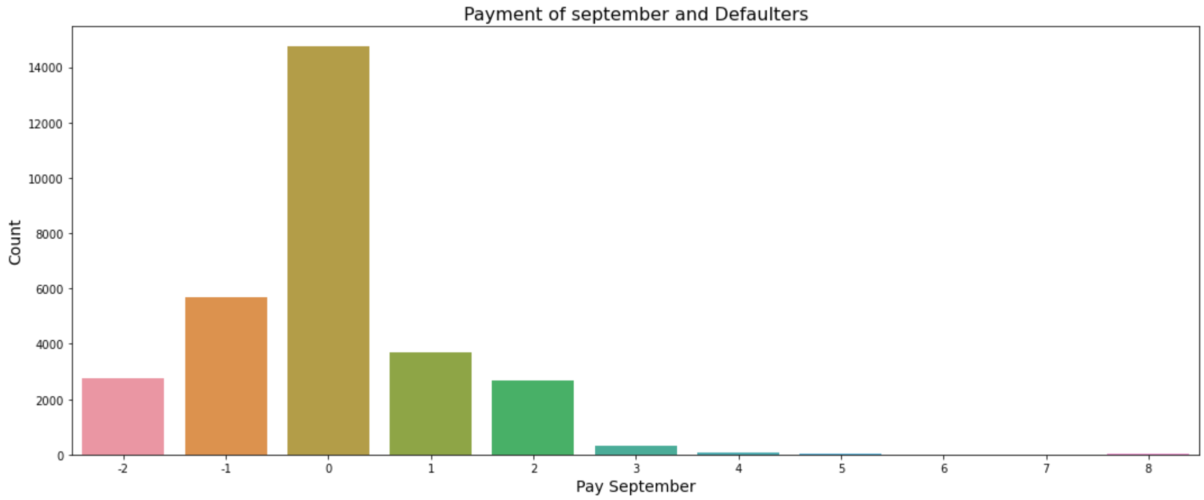


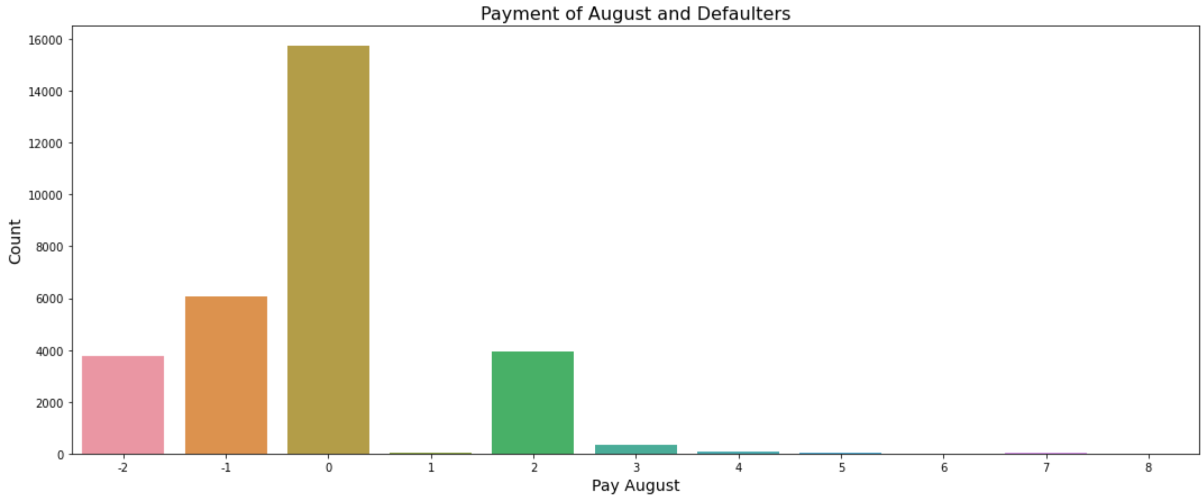
From the above figure we can say that most of the customers are from the age of 29. And we have more customers from the age group of 26 to 30.

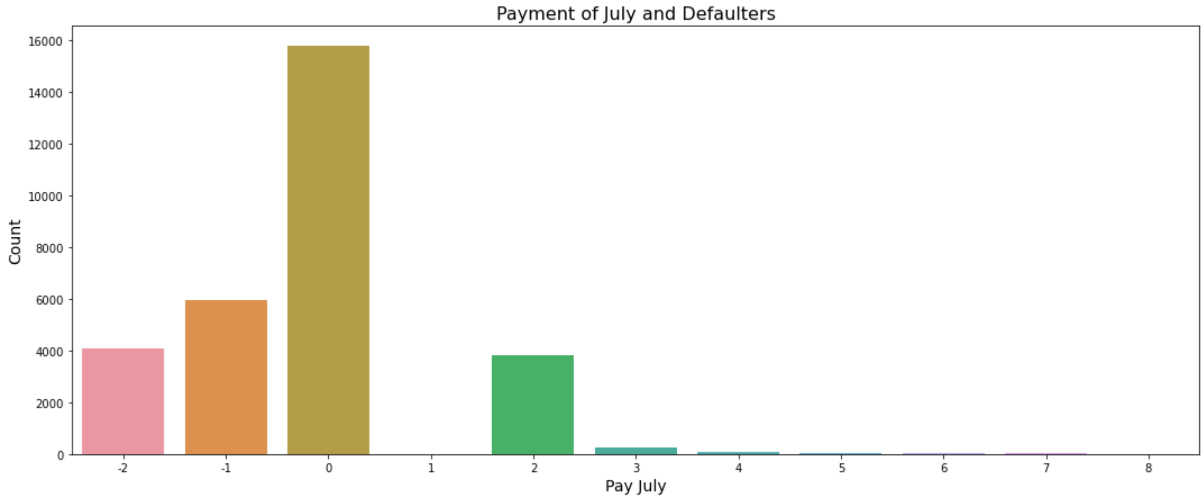


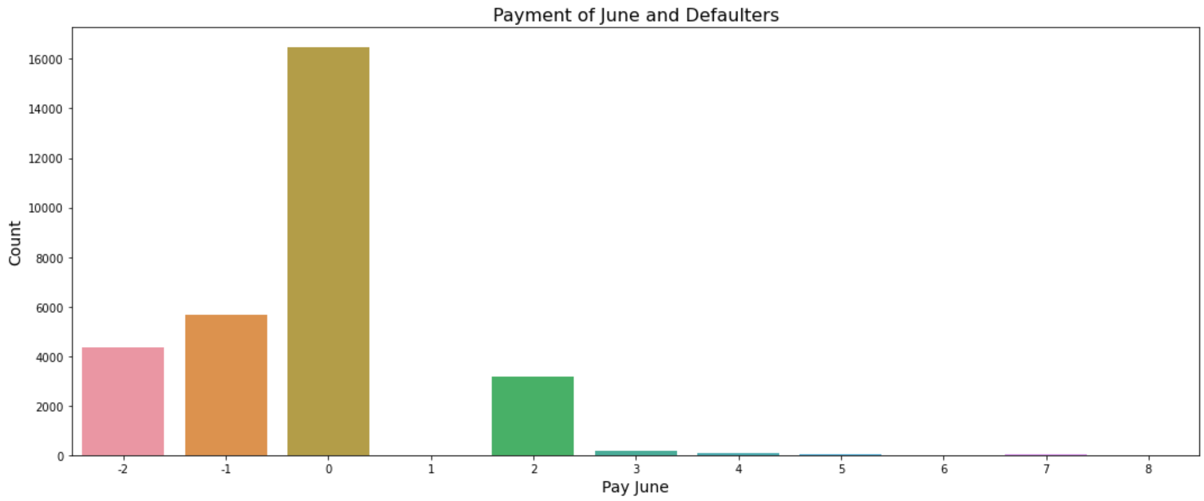
When it comes to the defaulters from all age groups it is nearly equal in number with slight differences.

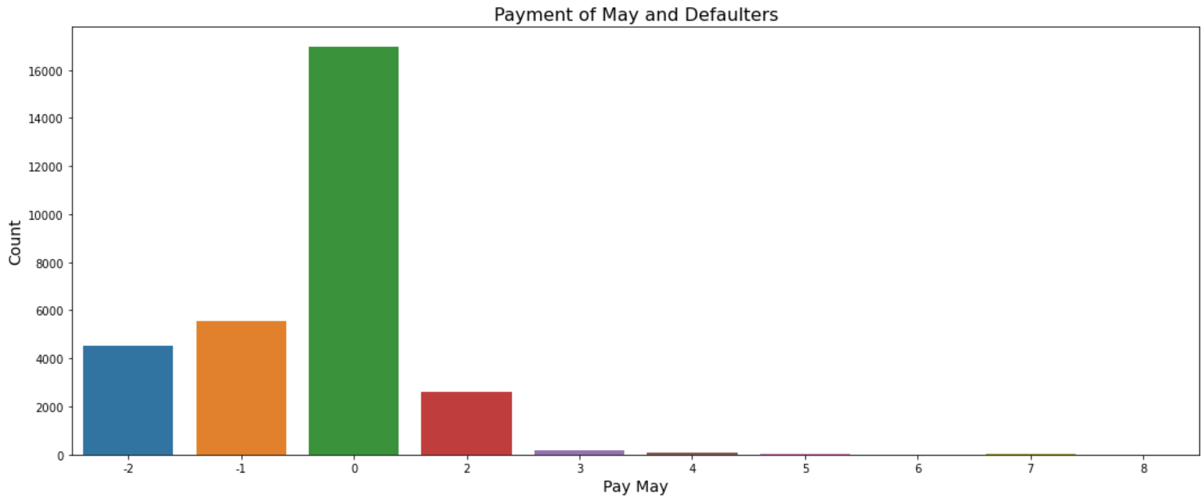
Analysis of History of Past payment

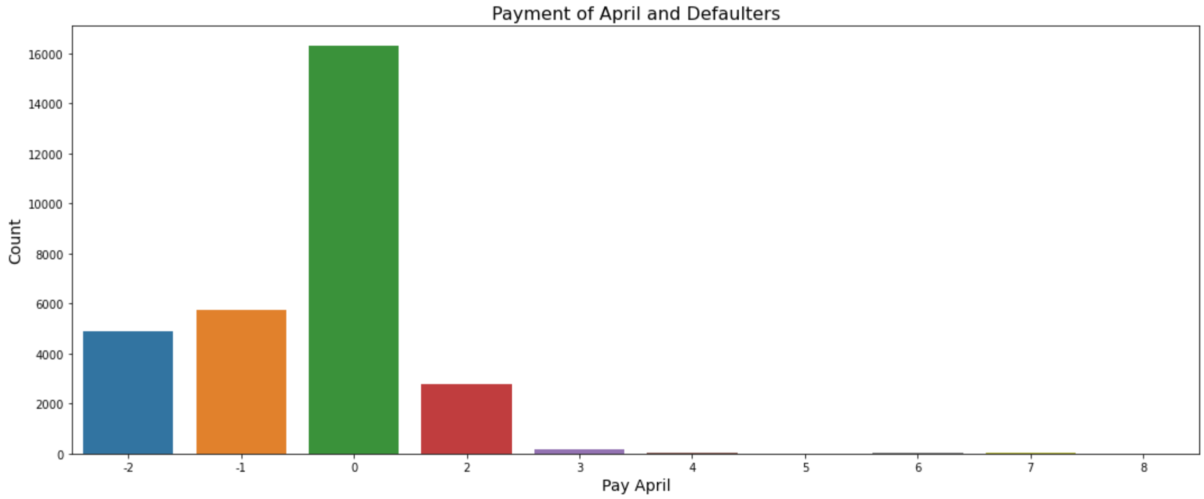






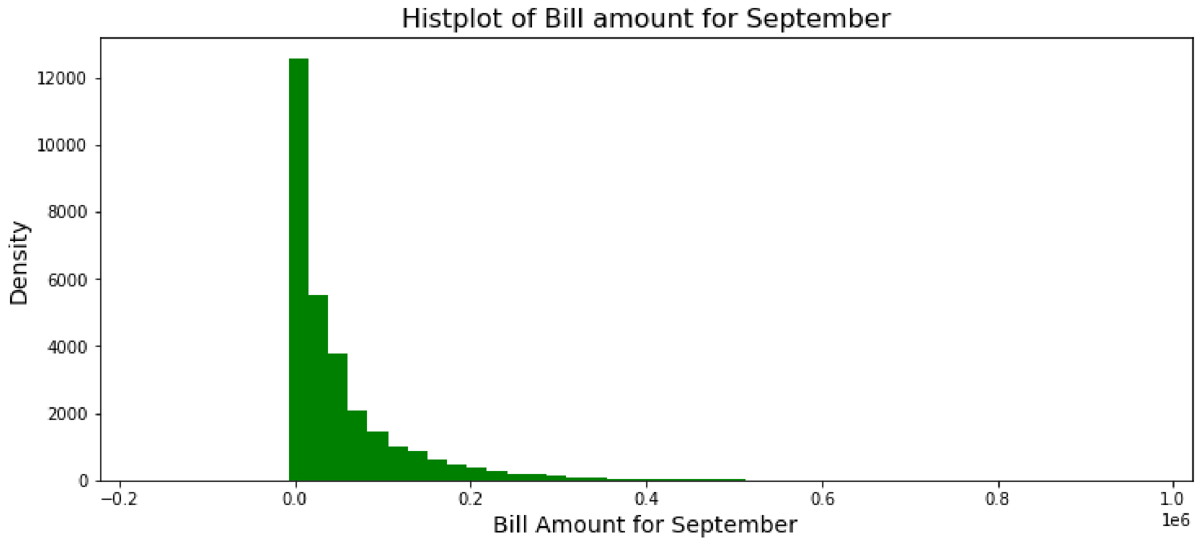


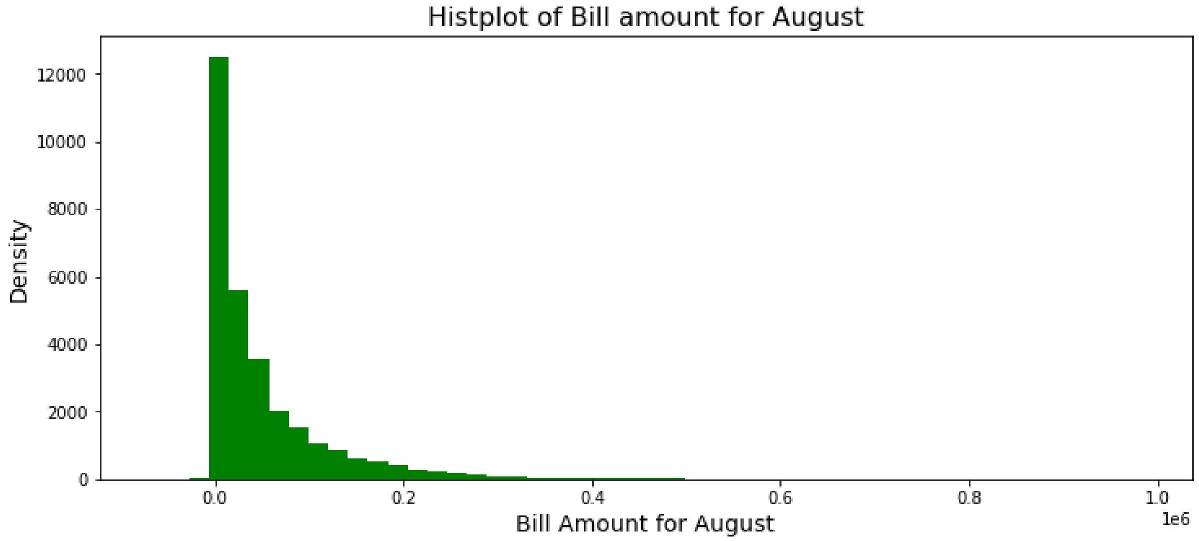


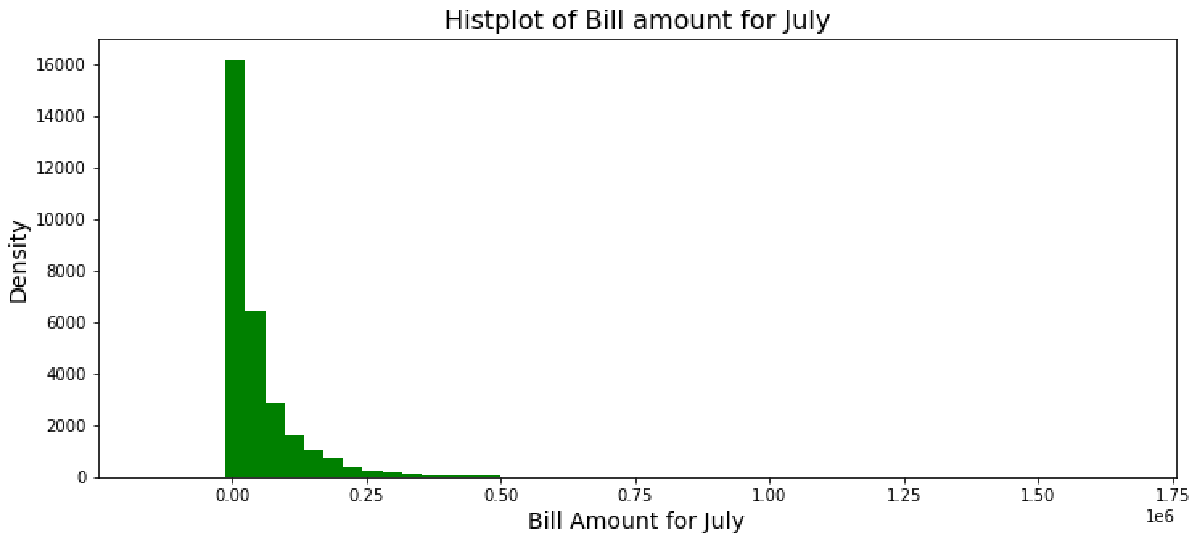


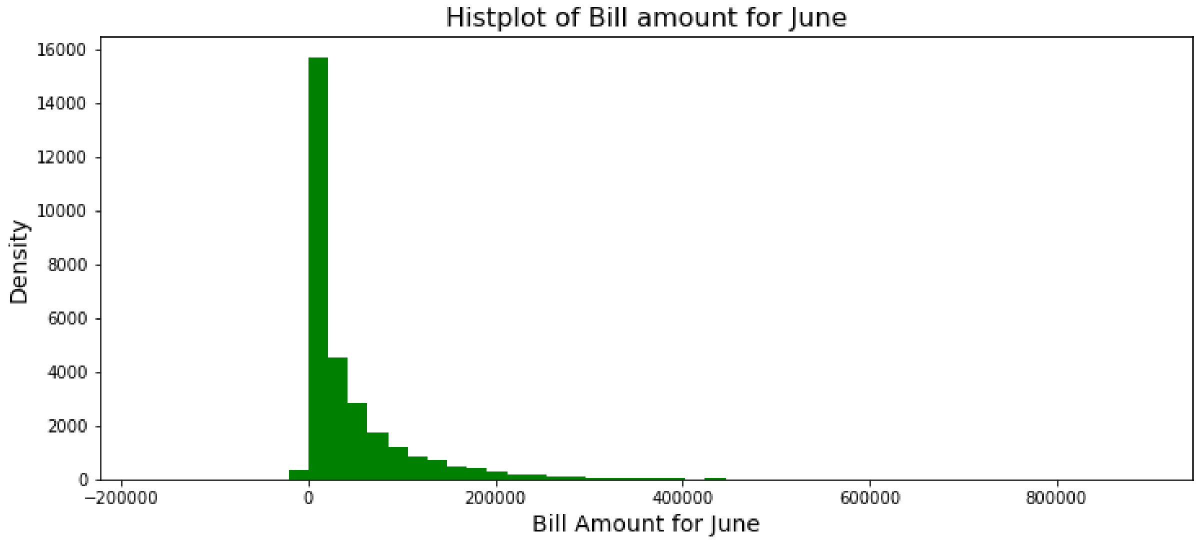
The numerical values in all the above analyses refer to -1=pay duly, 1=payment delay for one month, 2=payment delay for two months, 8=payment delay for eight months, 9=payment delay for nine months and above. All the above figures show the payment of the credit card due.

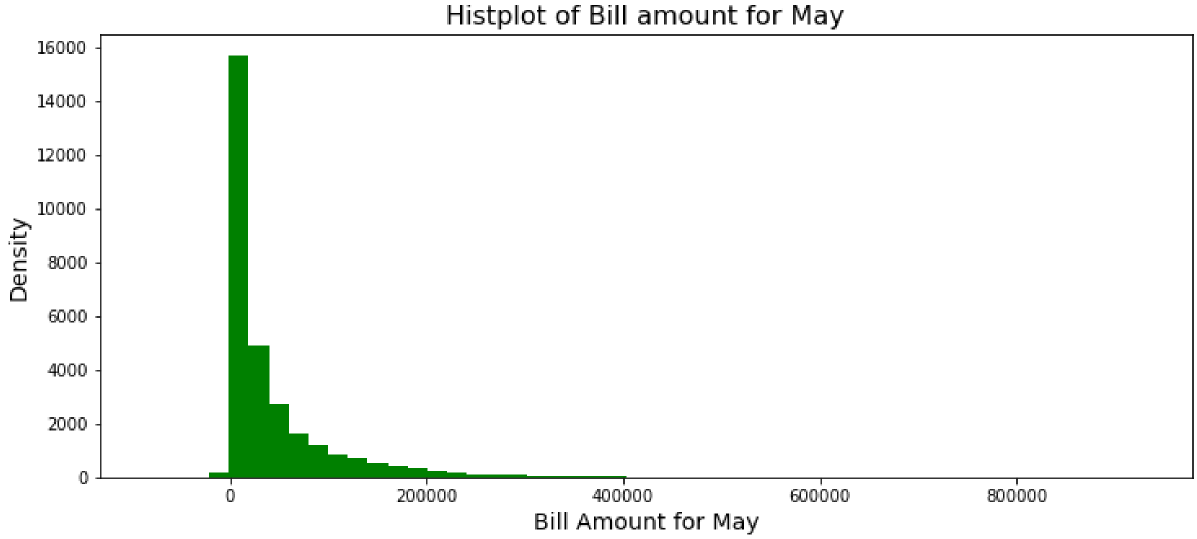
Analysis of Bill Amount

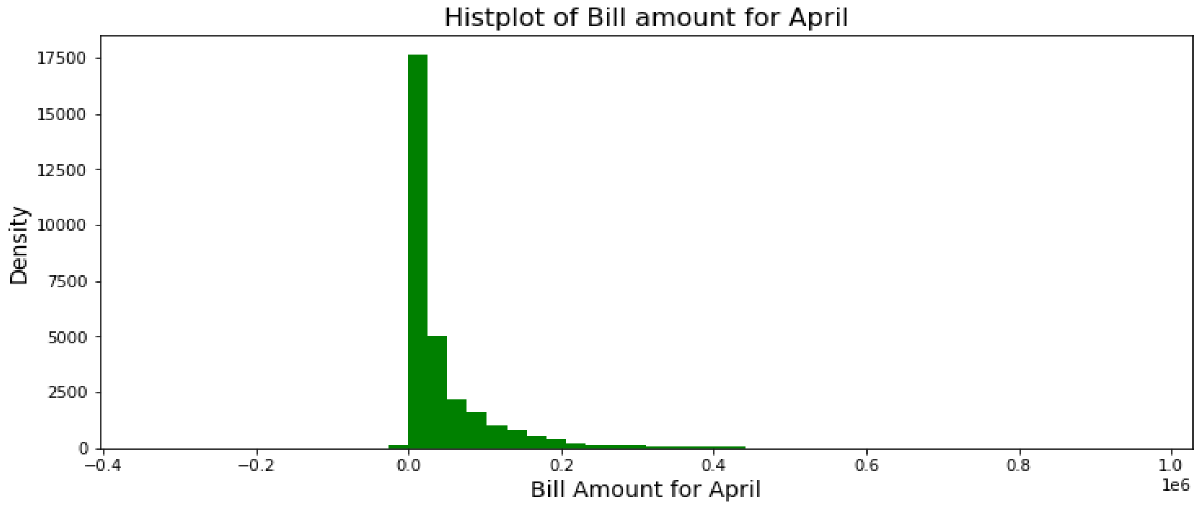






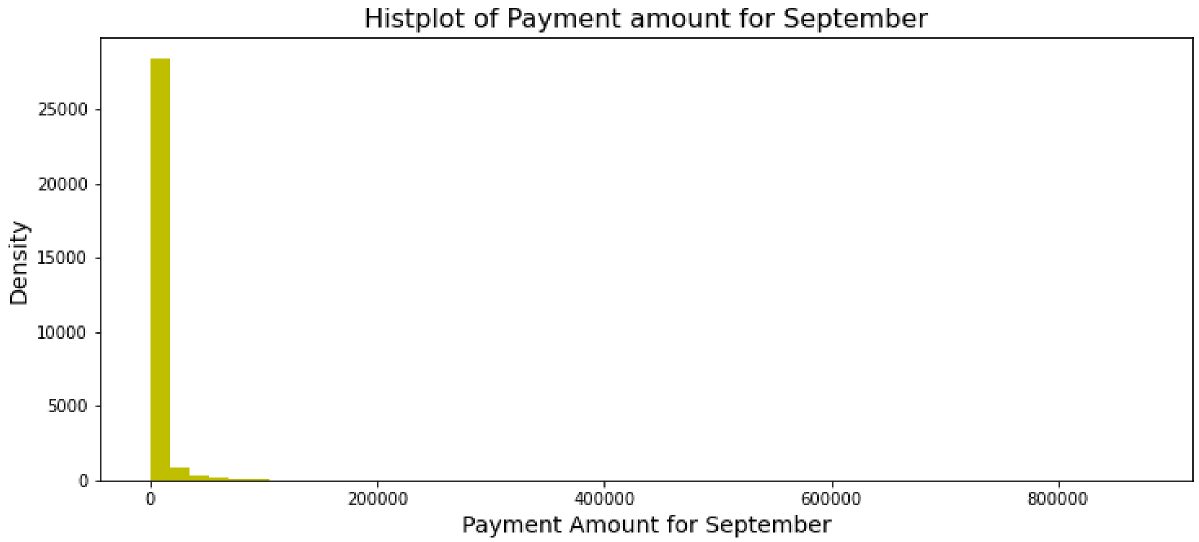


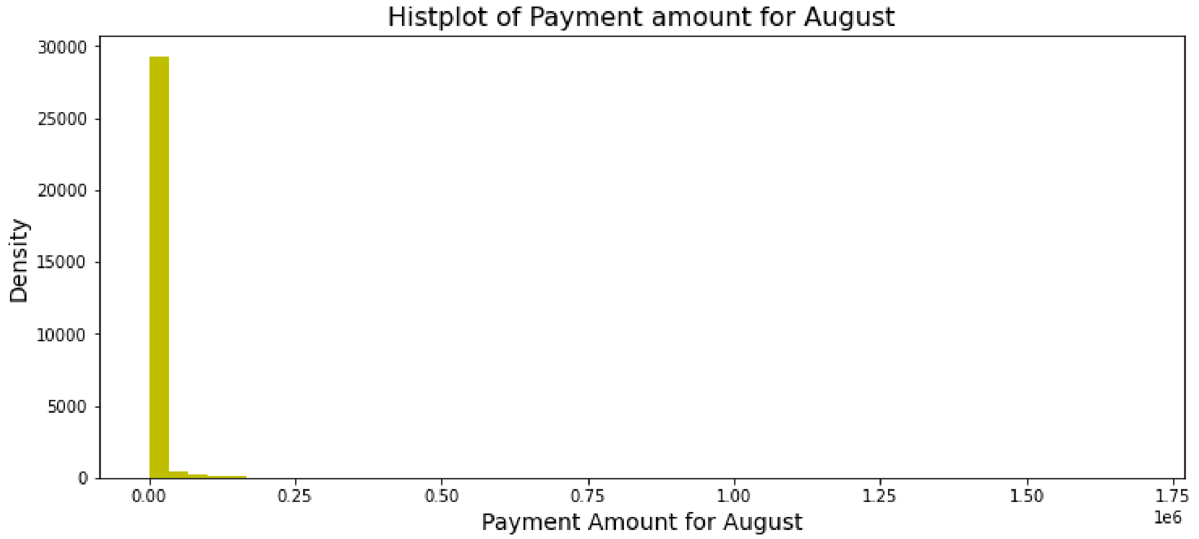


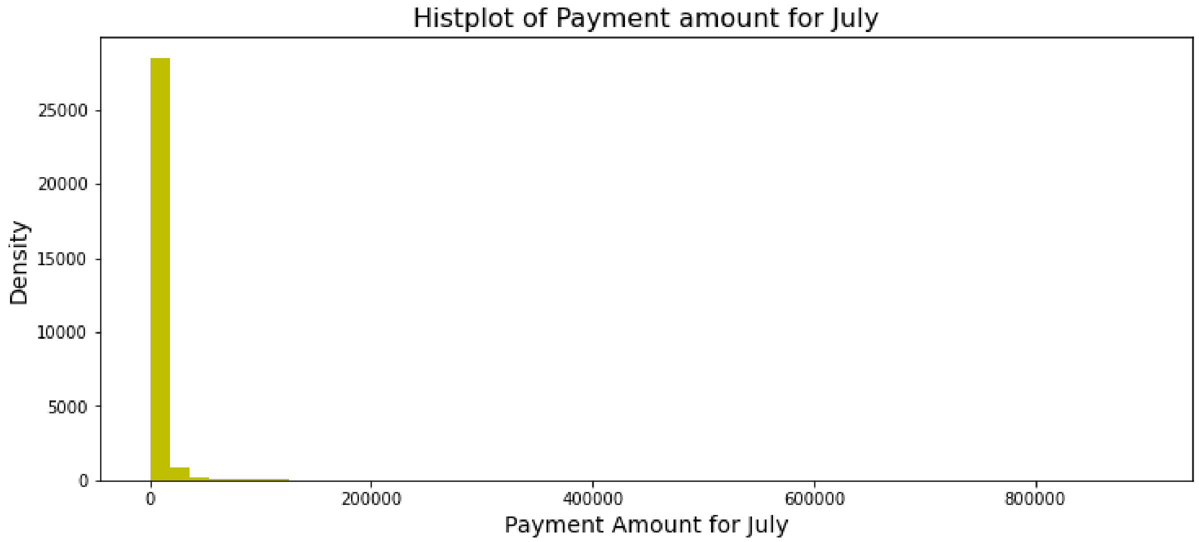


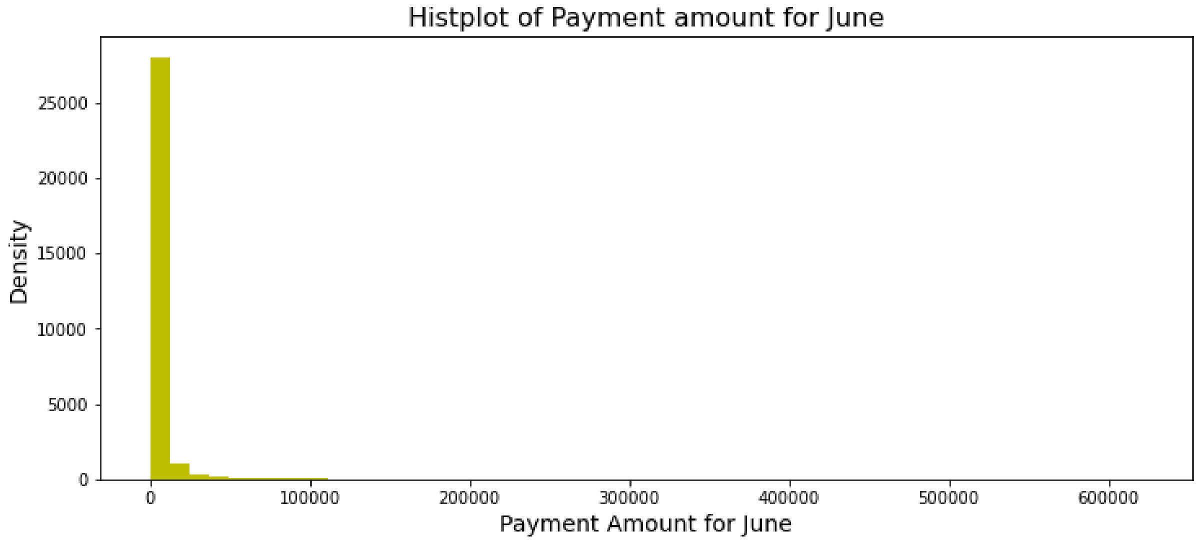
All the above figures show the density distribution of the number of bill statements for respective months. In some figures, the tail has extended towards the left and it is assumed that the respective monthly statement includes advance payments.

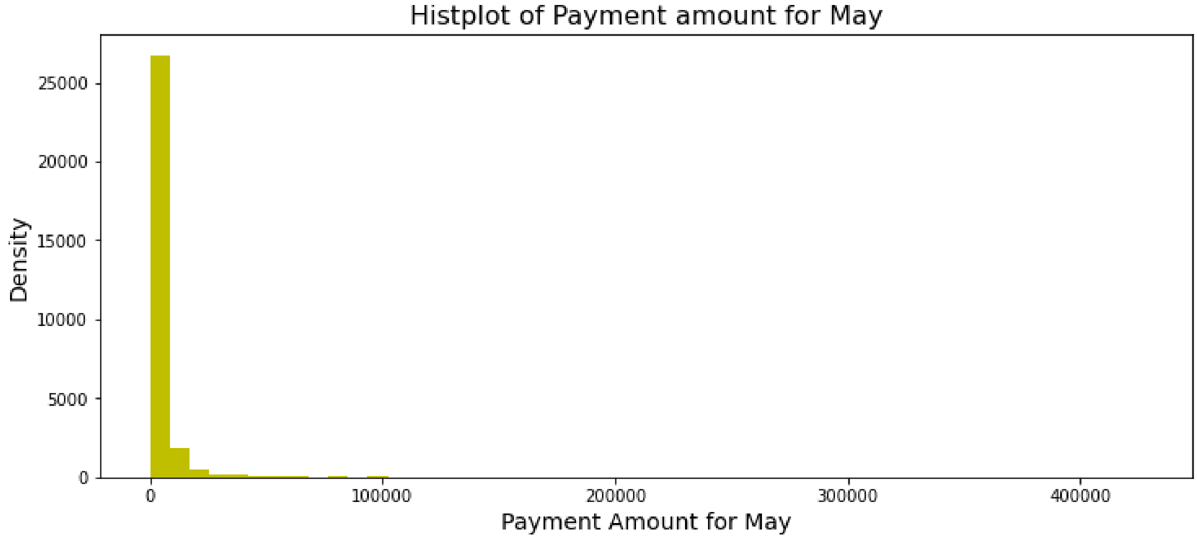
Analysis of Payment Amount

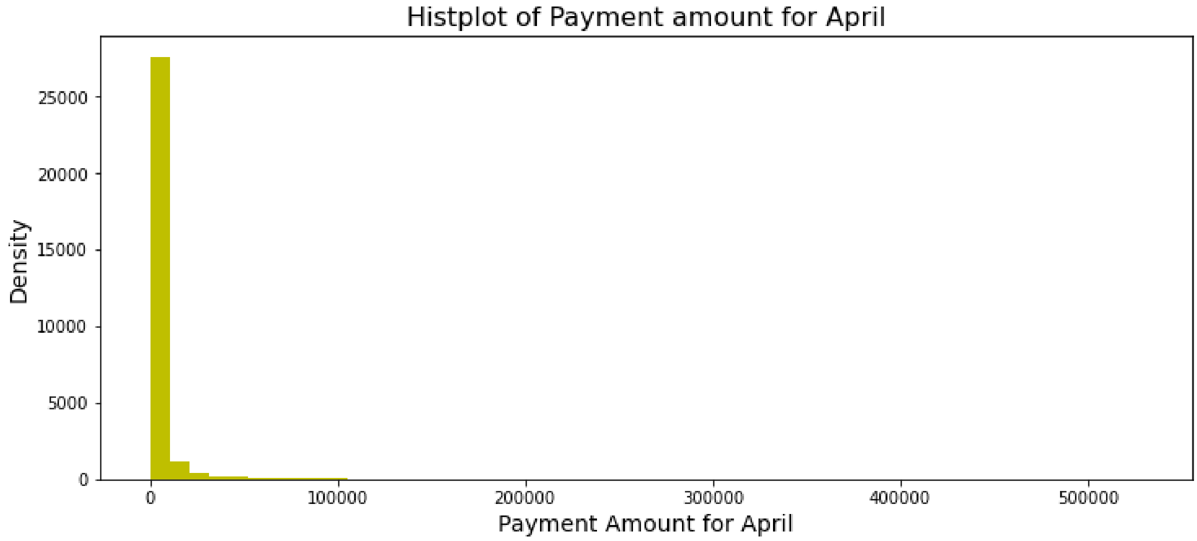












All the above figures show the density distribution of the number of previous payments for respective months.

Conversion of features

All the features in the dataset contain numeric data, but in the info of the dataset, it is showing the object as its data type. I have converted the features of the dataset into integers.

Variance Inflation Factor

The Variance Inflation Factor is used when we have the multi-collinearity between the features. The factor helps in reducing the inflation between the features by dropping some of the features which are having a high correlation among them.

In the given dataset the correlation between the features is very high and some of the features are dropped to reduce the correlation among them. Mainly bill amount features.

Data Modelling

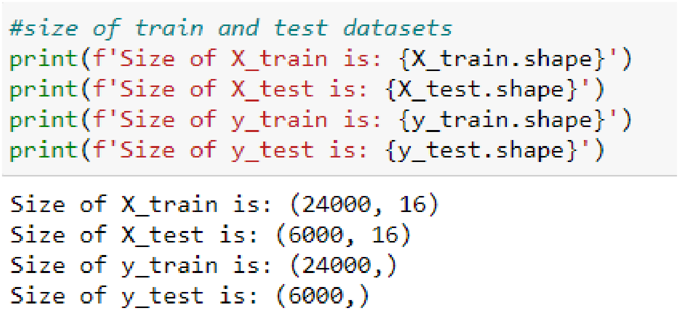
After the data preparation is completed it is ready for the purpose of analysis. Only numerical valued features are taken into consideration. The data were combined and labelled as X and y as independent and dependent variables respectively. The open, high and low columns are taken as independent variables (X) and the closing price is taken as dependent variable (y).

Splitting the data

The train_test_split was imported from the sklearn.model_selection. The data is now divided into 80% and 20% as train and test splits respectively. 80% of the data is taken for training the model and 20% is for a test and the random state was taken as 0.

Scaling the data

To normalise the data minmaxscaler was used from sklearn.preprocessing. It scales the data in the form of the standard deviation of the feature multiplied with the difference of maximum and minimum, again it was added to a minimum. At first, the training data was made fit into the scaling function and test data is transformed now. The output we get is X_train, X_test, y_train, y_test.



```
#size of train and test datasets
print(f'Size of X_train is: {X_train.shape}')
print(f'Size of X_test is: {X_test.shape}')
print(f'Size of y_train is: {y_train.shape}')
print(f'Size of y_test is: {y_test.shape}')
```

```
Size of X_train is: (24000, 16)
Size of X_test is: (6000, 16)
Size of y_train is: (24000,)
Size of y_test is: (6000,)
```

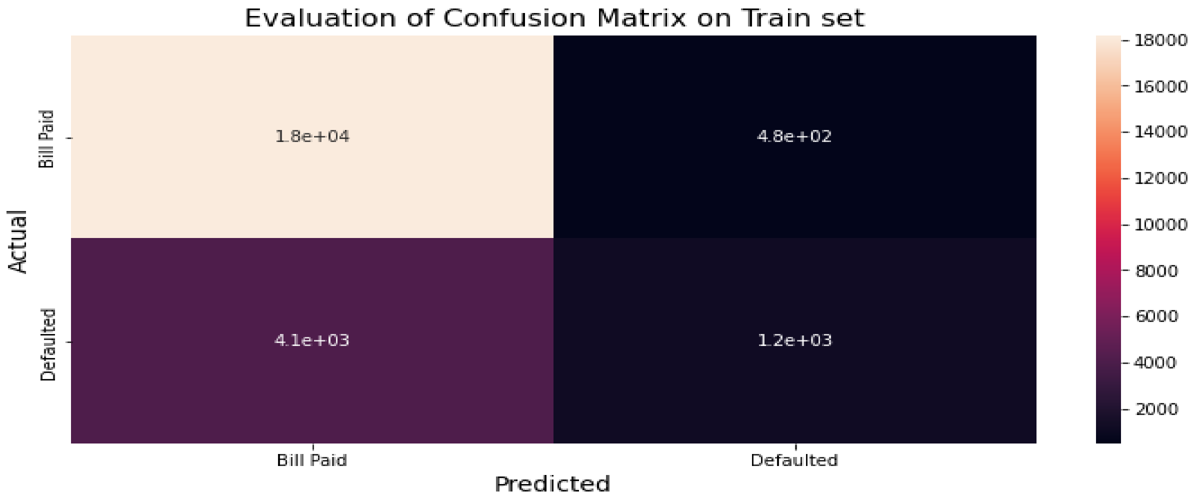
The next step is implementing the algorithm and training the model.

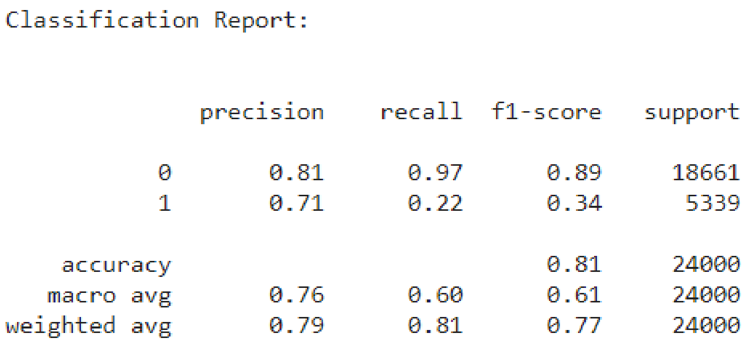
Logistic Regression

A logistic regression is a type of statistical procedure. It is used to refer specifically to the problem in which the dependent variable is binary, that is the number of available categories is two, while the problem with more than two categories is referred to as multi logistic regression.

Logistic Regression predicts the probability of the instance being positive.

Training data

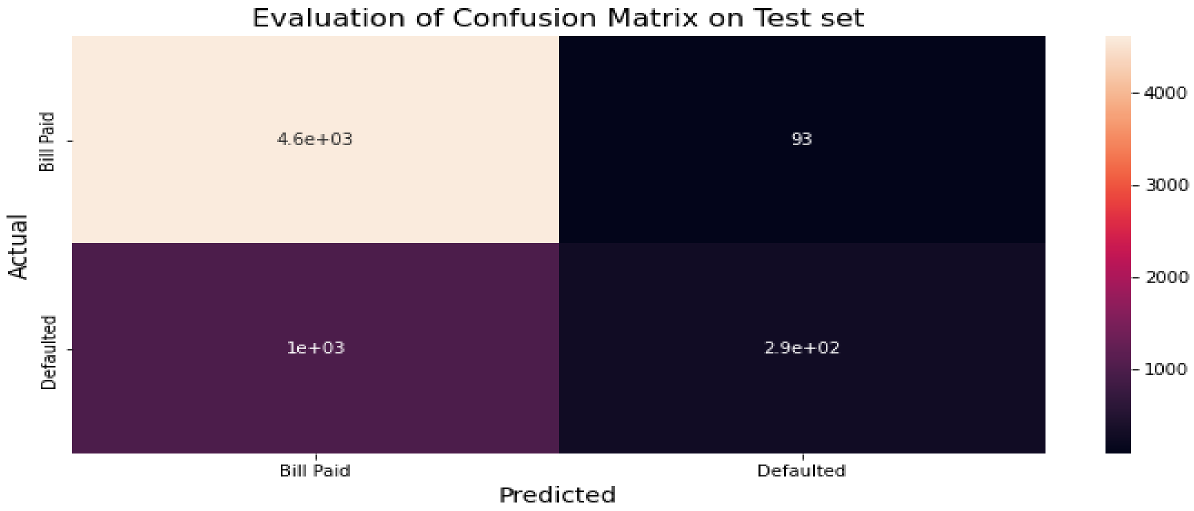


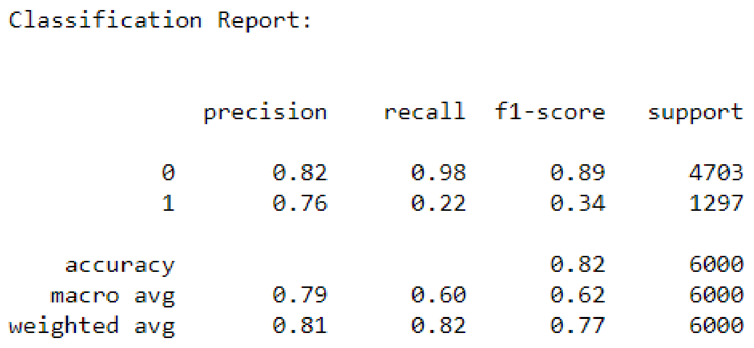


```
Classification Report:


              precision    recall  f1-score   support

           0       0.81      0.97      0.89     18661
           1       0.71      0.22      0.34      5339

    accuracy                           0.81     24000
   macro avg       0.76      0.60      0.61     24000
weighted avg       0.79      0.81      0.77     24000
```

The above figure shows how the model was trained and fit with the data. From the evaluation of the metrics of accuracy, we got a score of 81.65% on the training data. The confusion matrix shows the visualization of the classification of defaulters and non-defaulters. The classification report shows the accuracy level of the model is 81%.

Test dataset





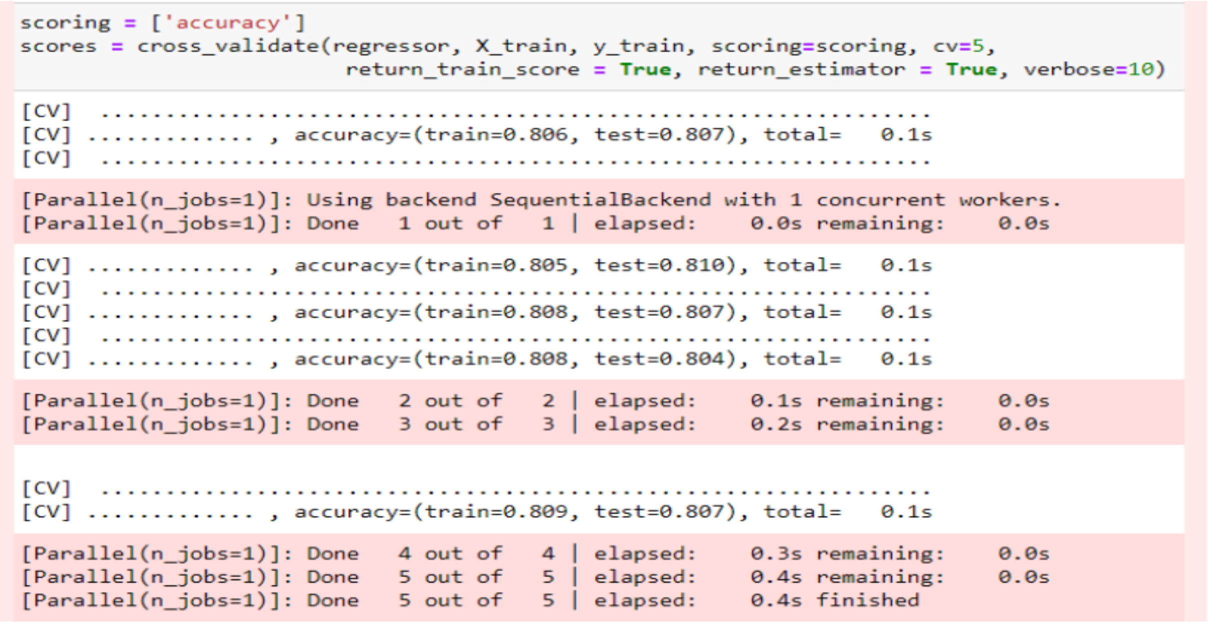
```
Classification Report:

              precision    recall  f1-score   support

           0       0.82      0.98      0.89      4703
           1       0.76      0.22      0.34      1297

    accuracy                           0.82      6000
   macro avg       0.79      0.60      0.62      6000
weighted avg       0.81      0.82      0.77      6000
```

The confusion matrix shows the visualization of the classification of defaulters and non-defaulters of the credit card bill which is due next month. The classification report shows the accuracy score on the test data of 82%.

Cross-validation on Logistic Regression

In cross-validation, we run our modelling process on different subsets of the data to get multiple measures of model quality. In this project, we could have 5 folds or experiments. We divide the data into 5 pieces, each being 20% of the full dataset.

Cross-validation gives a more accurate measure of model quality, which is especially important when we are making a lot of modelling decisions.



```
scoring = ['accuracy']
scores = cross_validate(regressor, X_train, y_train, scoring=scoring, cv=5,
                         return_train_score = True, return_estimator = True, verbose=10)
```

```
[CV]  ...............................................................
[CV] ............. , accuracy=(train=0.806, test=0.807), total=   0.1s
[CV]  ...............................................................
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done   1 out of   1 | elapsed:    0.0s remaining:    0.0s
[CV] ............. , accuracy=(train=0.805, test=0.810), total=   0.1s
[CV]  ...............................................................
[CV] ............. , accuracy=(train=0.808, test=0.807), total=   0.1s
[CV]  ...............................................................
[CV] ............. , accuracy=(train=0.808, test=0.804), total=   0.1s
[Parallel(n_jobs=1)]: Done   2 out of   2 | elapsed:    0.1s remaining:    0.0s
[Parallel(n_jobs=1)]: Done   3 out of   3 | elapsed:    0.2s remaining:    0.0s

[CV]  ...............................................................
[CV] ............. , accuracy=(train=0.809, test=0.807), total=   0.1s
[Parallel(n_jobs=1)]: Done   4 out of   4 | elapsed:    0.3s remaining:    0.0s
[Parallel(n_jobs=1)]: Done   5 out of   5 | elapsed:    0.4s remaining:    0.0s
[Parallel(n_jobs=1)]: Done   5 out of   5 | elapsed:    0.4s finished
```

From the above figure, we have the accuracy score on 5 cross-validation sets is 0.80708333, 0.81041667, 0.806875, 0.804375 and 0.80708333.

Model Evaluators(Metrics)

Classification Report

A Classification report is used to measure the quality of predictions from a classification algorithm. How many predictions are True and how many are False. More specifically, True Positives, False Positives, True negatives and False Negatives are used to predict the metrics of a classification report.

The report shows the main classification metrics precision, recall and f1-score on a per-class basis. The metrics are calculated by using true and false positives, true and false negatives.

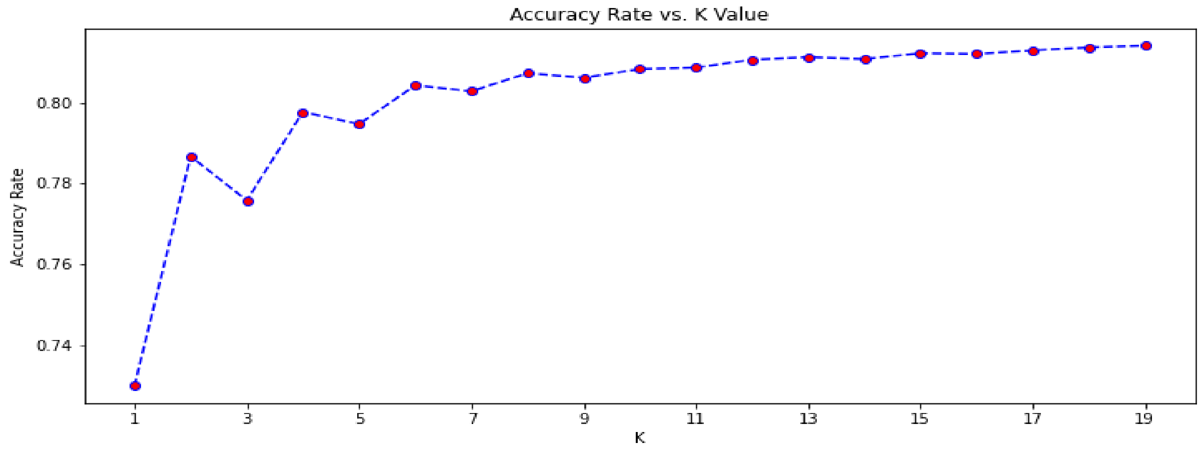
Confusion matrix

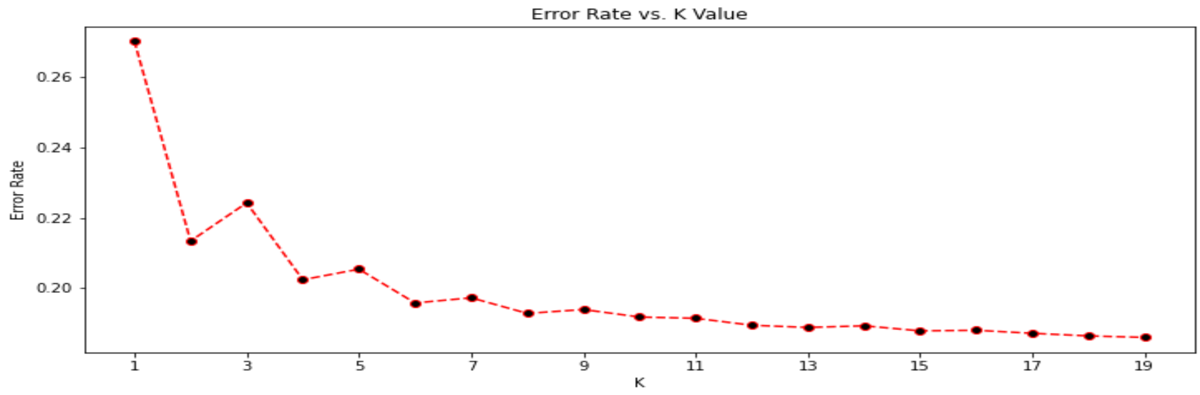
In machine learning and statistical classification, a confusion matrix is a table in which predictions are represented in columns and actual status is represented by rows. Sometimes this is reversed, with actual instances in rows and predictions in columns. The table is an extension of the confusion matrix in predictive analytics and makes it easy to see whether mislabeling has occurred and whether the predictions are more or less correct.

A confusion matrix is also known as an error matrix, and it is a type of contingency table.

K Near Neighbor

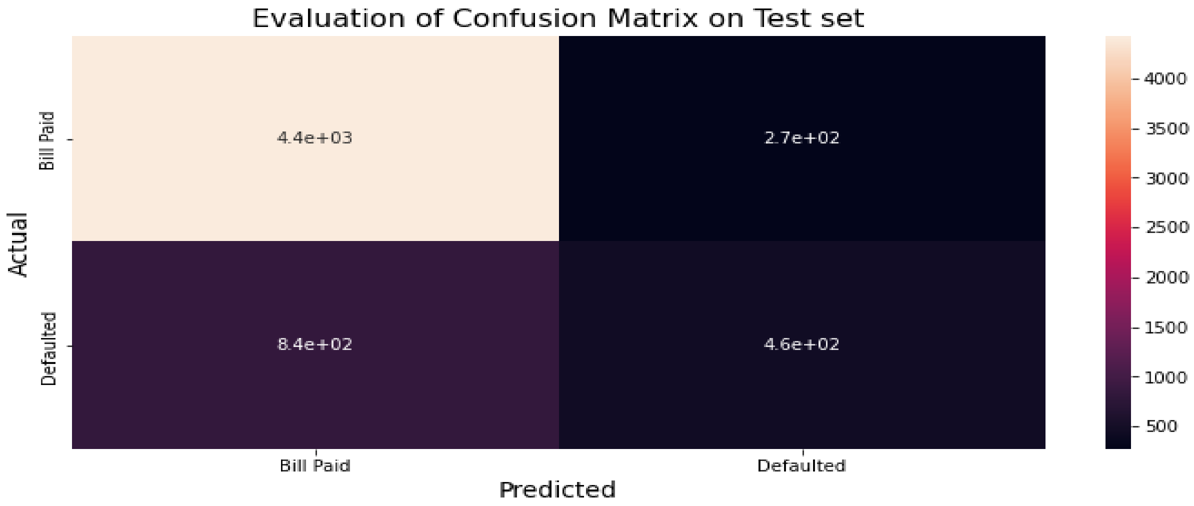
The k-Nearest Neighbors algorithm (or k-NN for short) is a non-parametric method used for classification and regression problems. In this, the output is a class membership. An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors (k is a positive integer, typically small). If k = 1, then the object is simply assigned to the class of that single nearest neighbor.

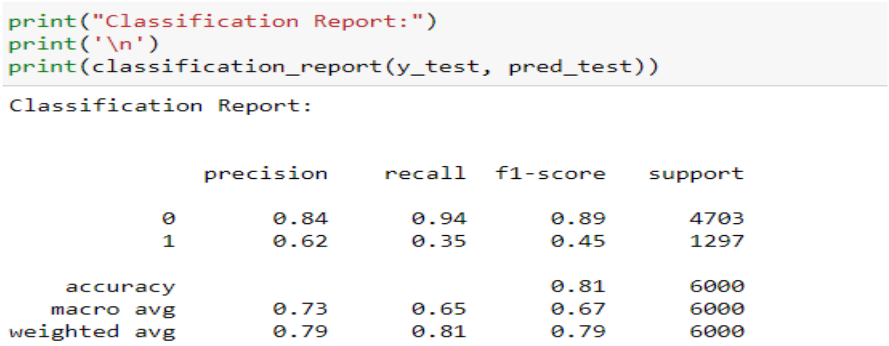




The above figure is used to find the value of k where we will have good accuracy with a minimum error rate. We can say that at 15 we have a good choice of k with an accuracy of 0.82 and an error rate of 0.18.

Test dataset





```
print("Classification Report:")
print('\n')
print(classification_report(y_test, pred_test))
```

```
Classification Report:


              precision    recall  f1-score   support

           0       0.84      0.94      0.89      4703
           1       0.62      0.35      0.45      1297

    accuracy                           0.81      6000
   macro avg       0.73      0.65      0.67      6000
weighted avg       0.79      0.81      0.79      6000
```

Trained the knn model at k as 15 to predict the defaulters of the credit card bill which is due next month.

The confusion matrix shows the visualization of the classification of defaulters and non-defaulters. The classification report shows the accuracy score on the test data of 81%.

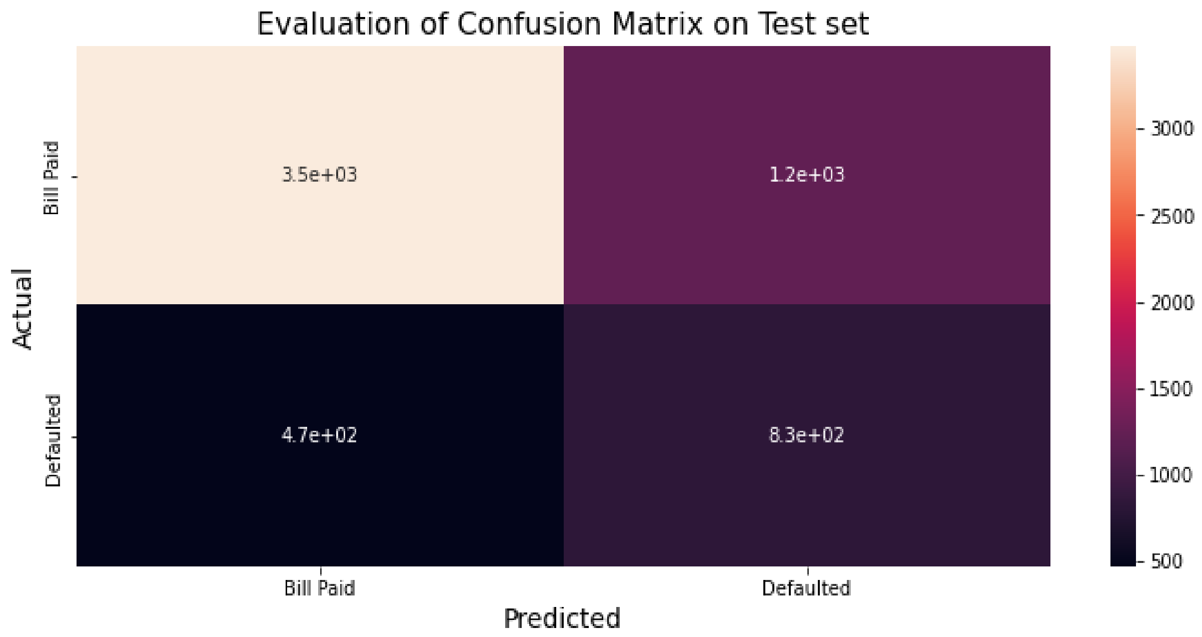
Naive Bayes

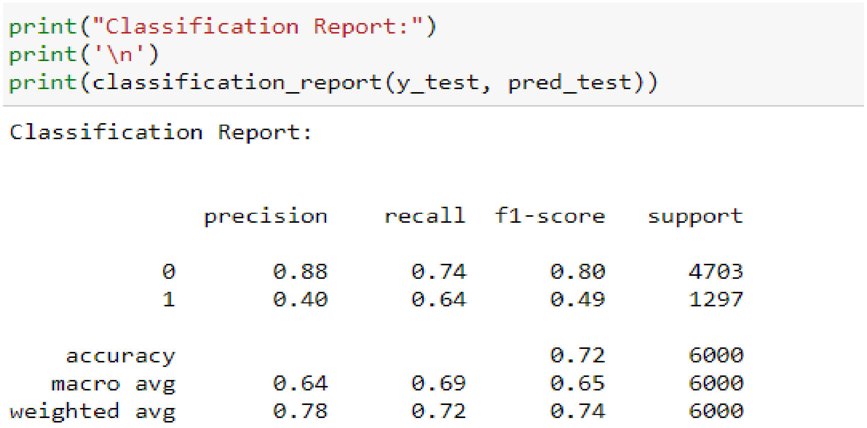
In machine learning, naive Bayes classifiers are a family of simple probabilistic classifiers based on applying Bayes' theorem with strong (naive) independence assumptions between the features.

In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.

$$P(L \mid \text{features}) = \frac{\text{P(features | L)P(L)}}{\text{P(features)}}$$

Test dataset





```
print("Classification Report:")
print('\n')
print(classification_report(y_test, pred_test))
```

```
Classification Report:


              precision    recall  f1-score   support

           0       0.88      0.74      0.80      4703
           1       0.40      0.64      0.49      1297

    accuracy                           0.72      6000
   macro avg       0.64      0.69      0.65      6000
weighted avg       0.78      0.72      0.74      6000
```

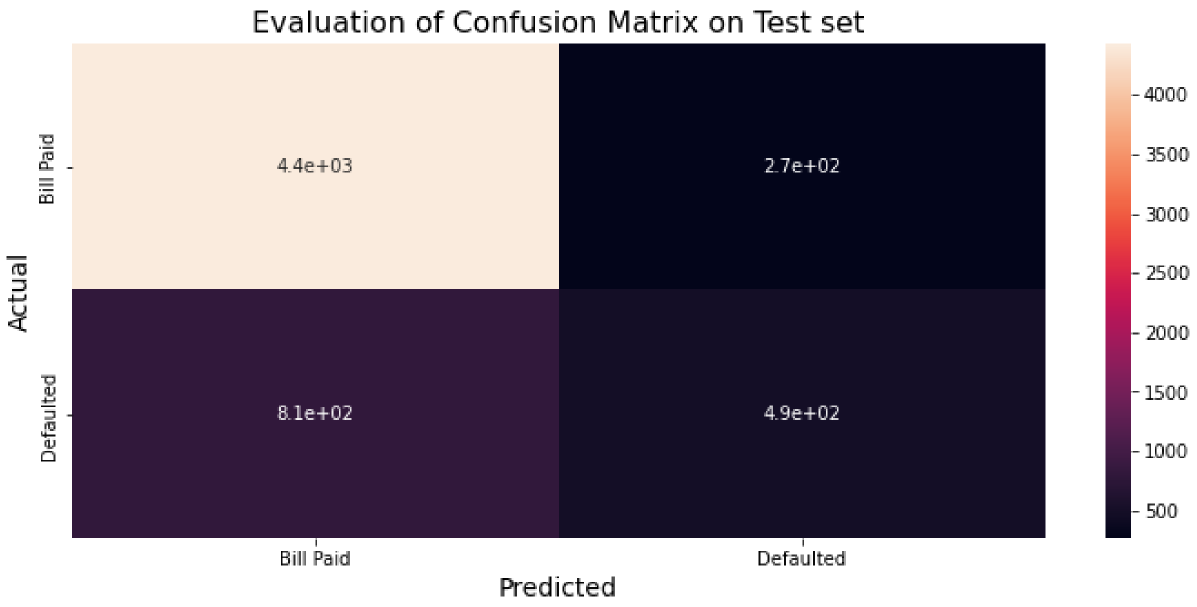
The confusion matrix shows the visualization of the classification of defaulters and non-defaulters of the credit card bill which is due next month. The classification report shows the accuracy score on the test data of 72%.

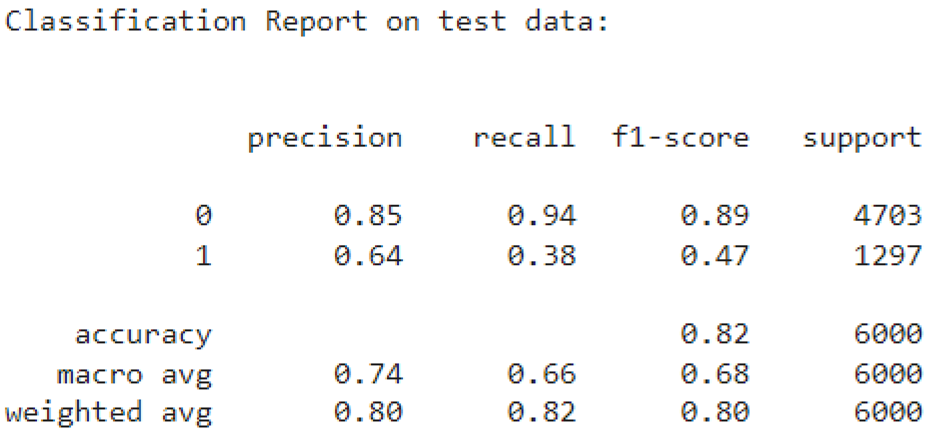
Random Forest Classifier

Random Forest is a supervised learning algorithm that uses ensemble learning methods for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes(classification).

The model was imported and trained with the data available in the training dataset. Defined the predicted variable and checked the score of the model.

Training Dataset





```
Classification Report on test data:

              precision    recall  f1-score   support

           0       0.85      0.94      0.89      4703
           1       0.64      0.38      0.47      1297

    accuracy                           0.82      6000
   macro avg       0.74      0.66      0.68      6000
weighted avg       0.80      0.82      0.80      6000
```

Conclusion

- 500000 is the highest limit balance and 327680 is the least one.
- As we have more customers from females, we have more defaulters from the same category.
- In both married and singles, the defaulters are equal in number.
- Coming to education, the university category customers are high in number as well as in default.

- The dataset contains customers from the 21 to 79 age group. Customers of the age group of 29 are more in number but there is not much difference in the age criterion in defaulting.
- Some of the bill amounts are extended towards the left i.e., it is showing the negative, it is assumed that there are some advance payments.

Logistic Regression, Random Forest, K near neighbor and Naive Bayes are the models used to train to predict the defaulters and non-defaulters of the credit card bill which is due for next month.

The model's accuracy of predictions is

Model	**Accuracy (%)**
Logistic Regression	81.65
Random Forest	82.00
K Near Neighbor	81.00
Naive Bayes	72.00

In the above table we can see that when the model is trained with Random Forest Classifier, the model is predicting the defaulters with the accuracy of 82% which is actually a good result of the prediction model.