Credit Card Default Prediction

Submitted By:
Lakshmi Narayana
Data Science Trainee
At Alma Better

Default Prediction

- Problem Statement
- EDA and Feature engineering
- Feature selection
- Data preparation
- Implementation of model
- Evaluation of Model

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.

Data Pipeline

- **Data processing-:** In this part I have changed the names of all columns, dropped the first row data and checked for null values.
- **Data processing-2:** In this part I have gone through each feature which are selected in the first phase and renamed some of the columns.
- **EDA:** In this part some exploratory data analysis was done on the features selected in part 1 and 2.
- **VIF:** Selected only the features which does not have multi correlation among them.
- **Create a model:** Finally in this part created models, trained and tested it on the available dataset.

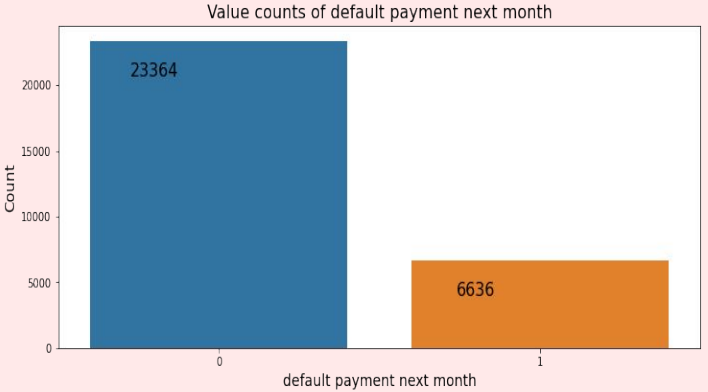
Data Description

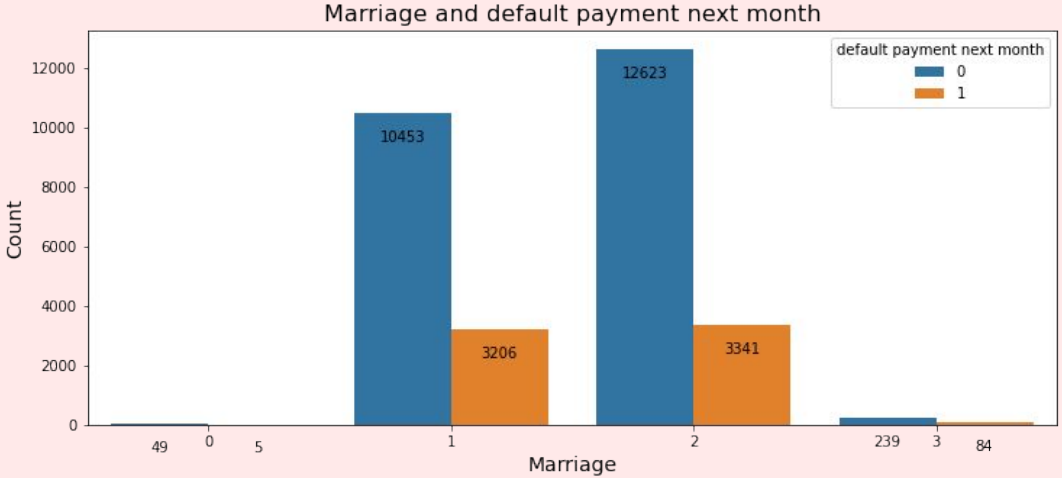
Attribute Information:

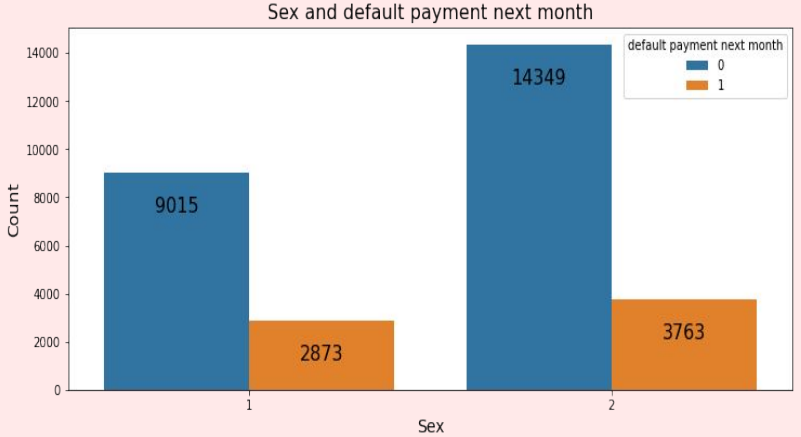
This research employed a binary variable, default payment (Yes = 1, No = 0), as the response variable. This study reviewed the literature and used the following 23 variables as explanatory variables:
X1: Amount of the given credit (NT dollar): it includes both the individual consumer credit and his/her 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.

Defining the Dependent Variable

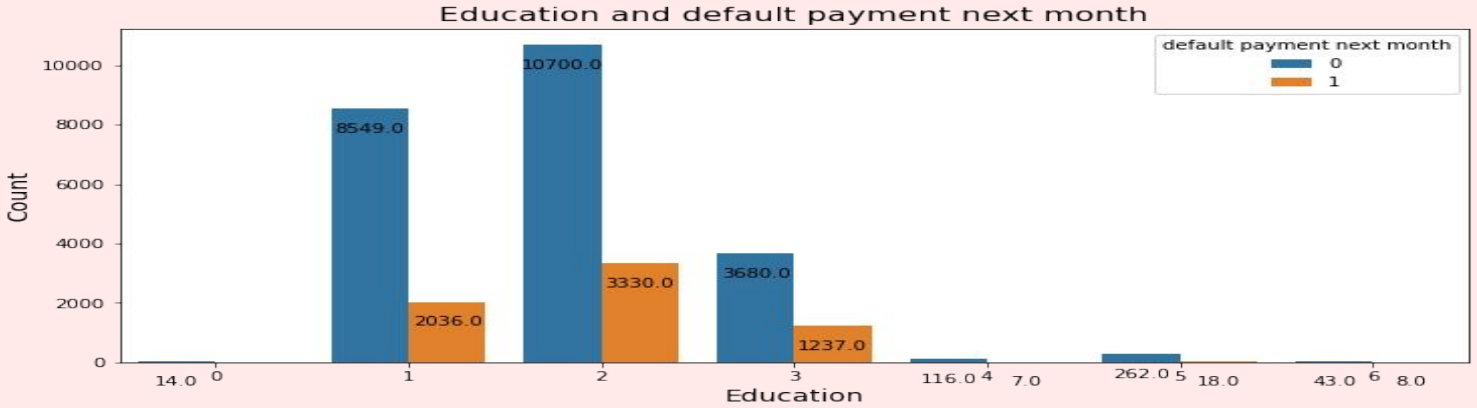
Default payment next month is taken as the dependent variable.
Default payment next month: Contains the binary values, 1 says that the customer did not pay the next month due and 0 says that the customer made payment of his due.

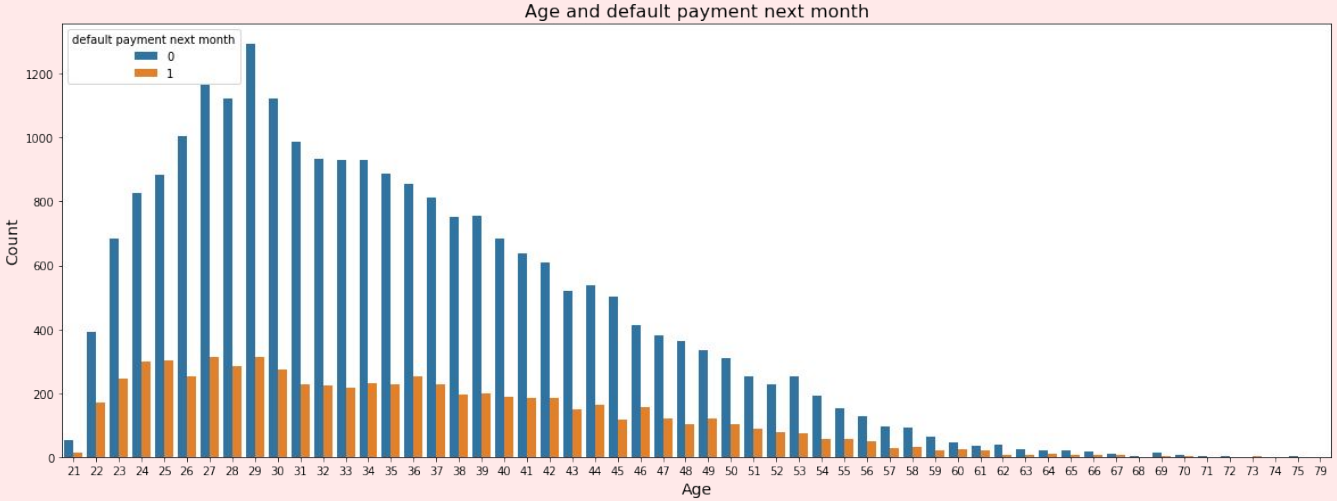






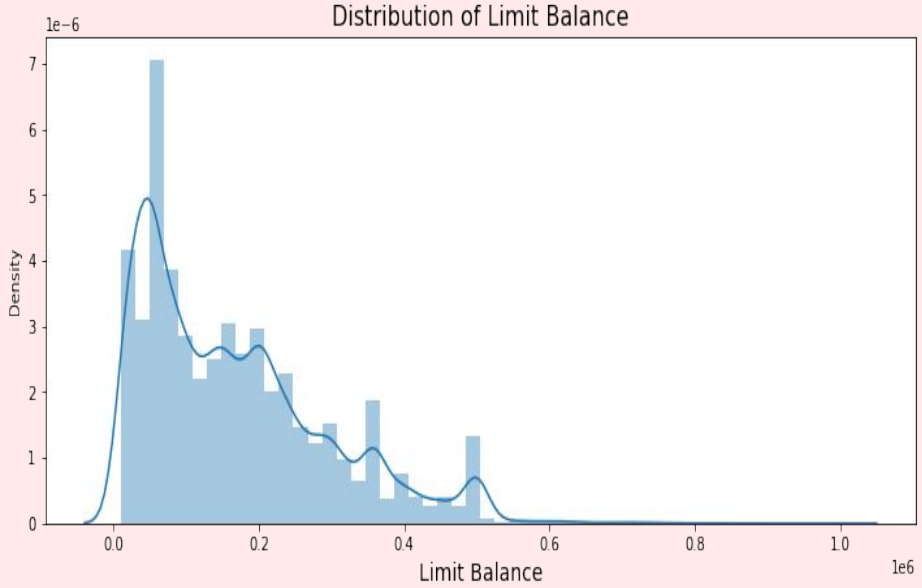
Defining the Dependent Variable

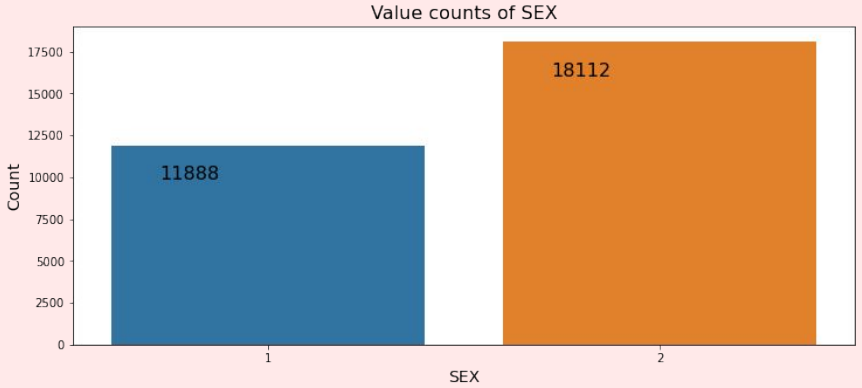


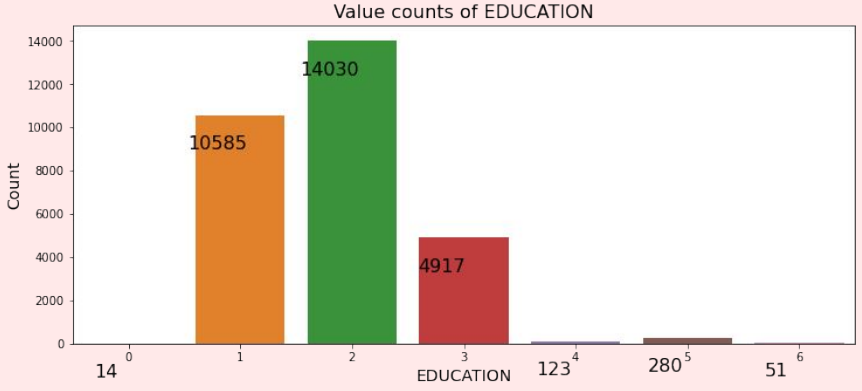


Exploratory Data Analysis

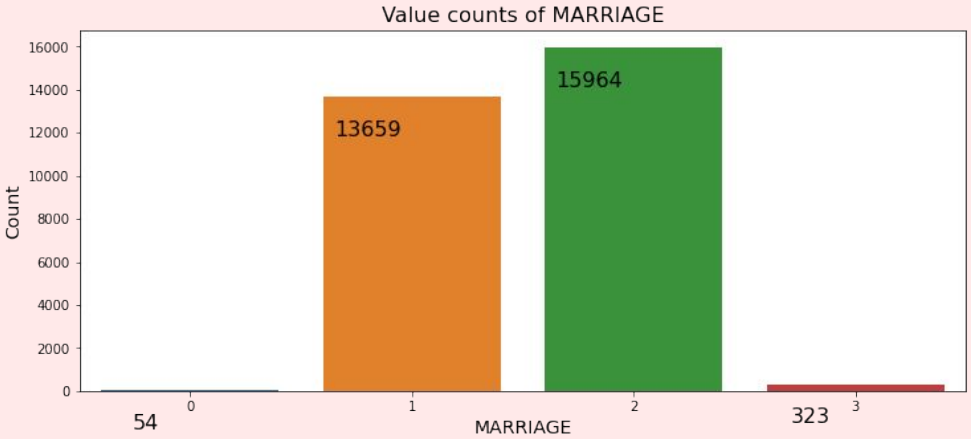
- Cleaning the null values.
- Exploratory Analysis.
- Conversion of data type of features.
- Checking for Multicollinearity.

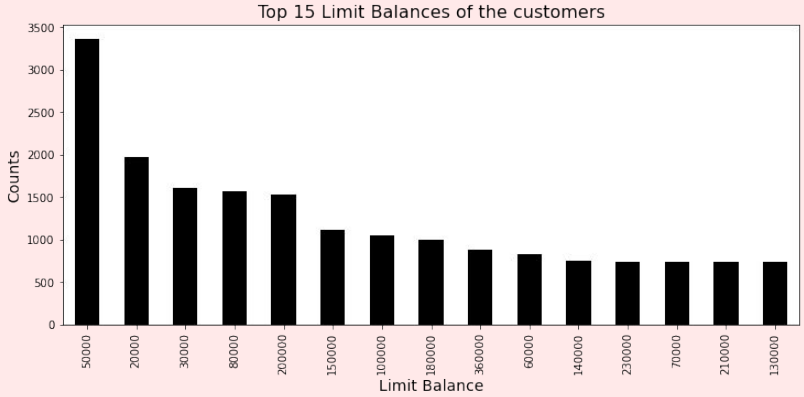


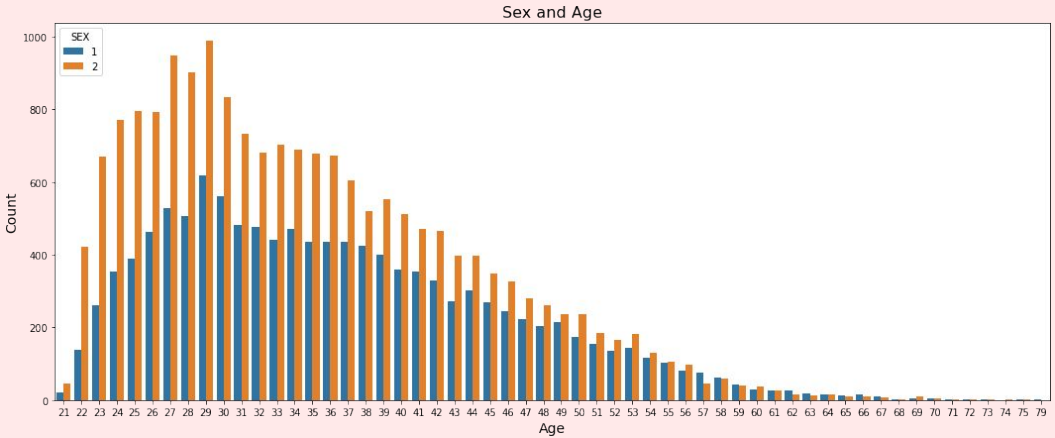


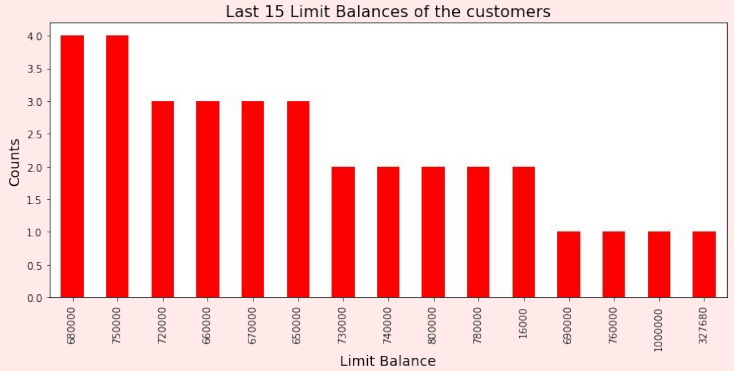


Exploratory Data Analysis

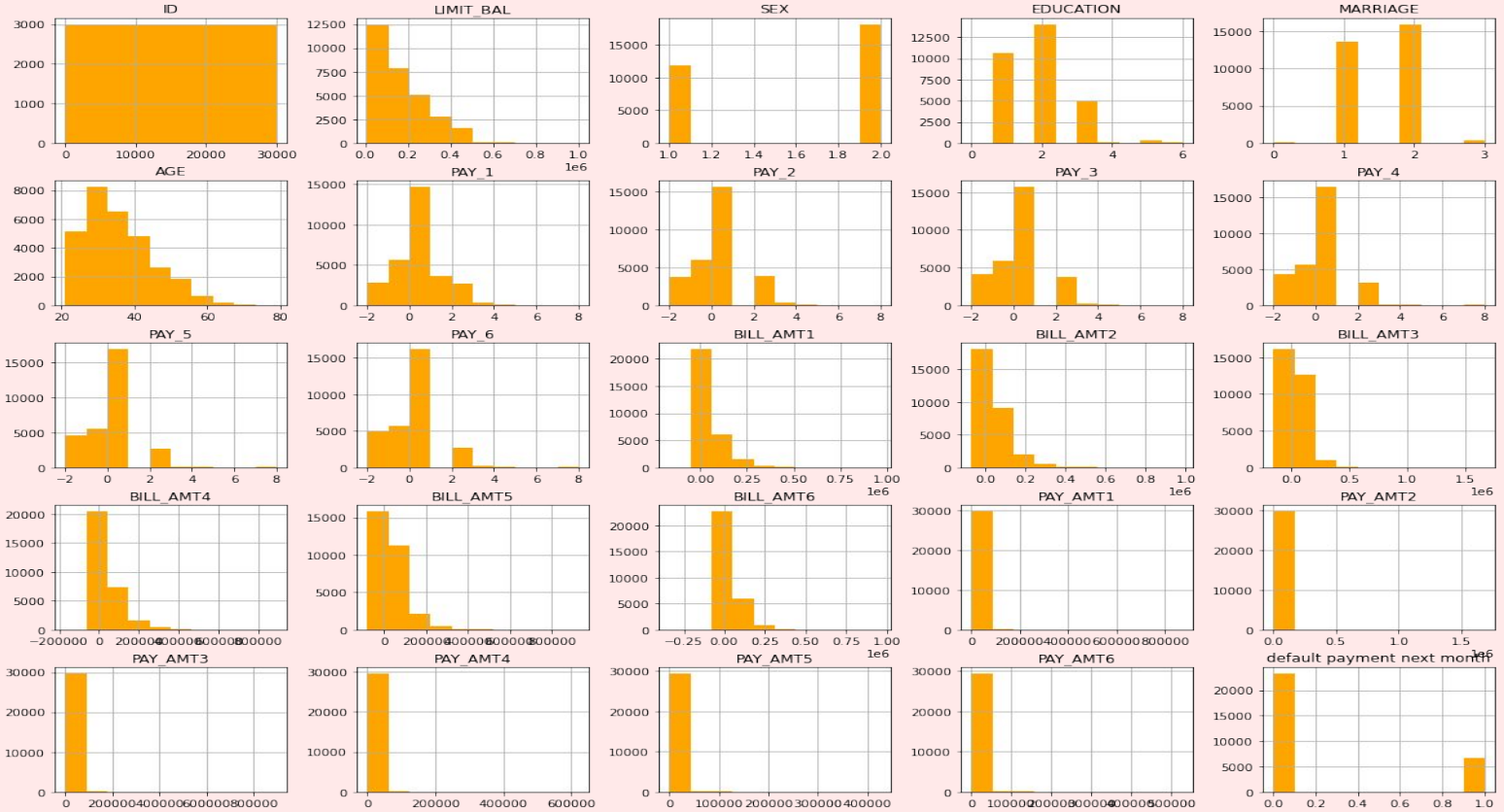




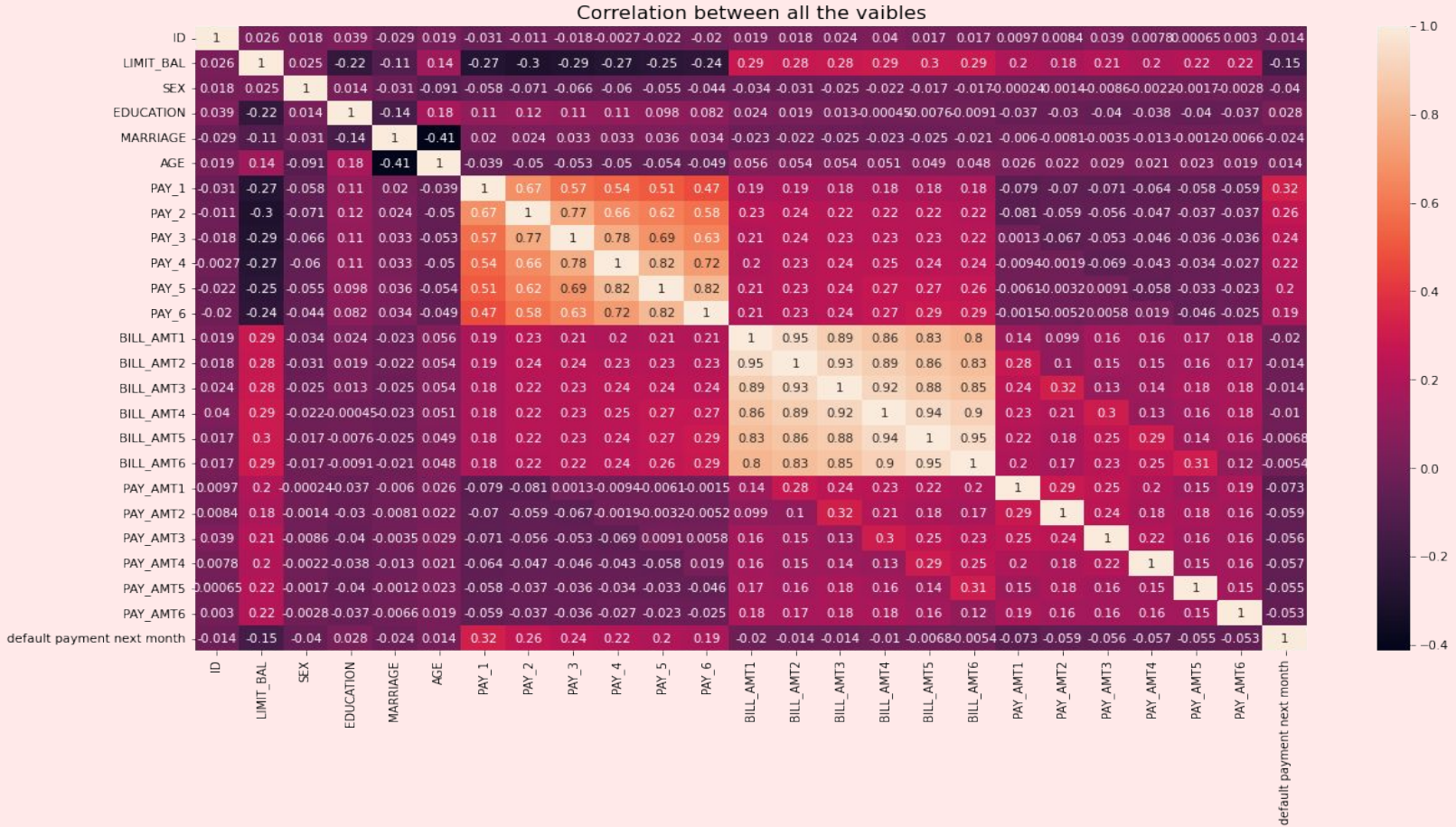




Exploratory Data Analysis



Exploratory Data Analysis



Data Preparation

- Before the model implementation the Variance Inflation Factor was used to do feature selection.
- Few features are dropped.
- Divided the dataset into train and test set in the ratio of 80:20 with random_state as 0.
- StandardScaler was used to scale the data.

```
#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,)
```

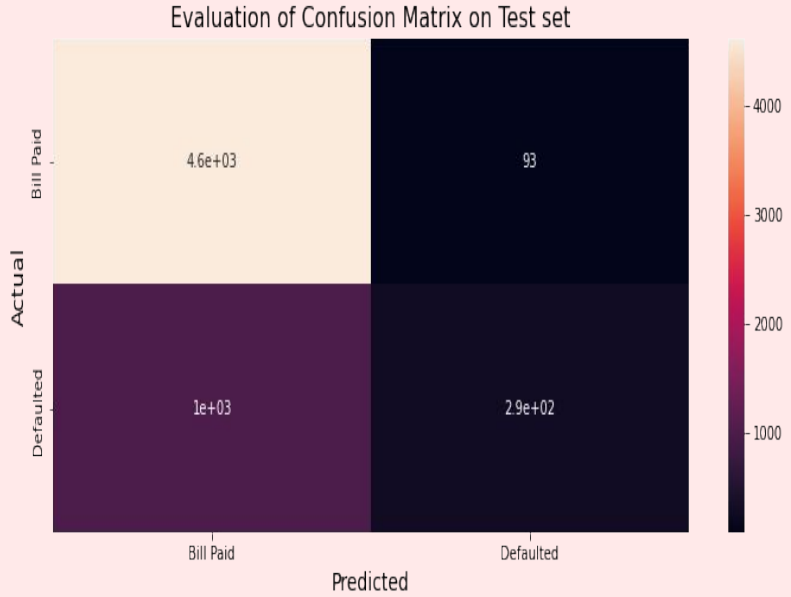
Model Implementation

Logistic Regression:

- The Scaled data was used for the model implementation.
- Fit the trained dataset to the model.
- The accuracy score on the test dataset is 81.65%
- Defining the predicted values form the model.

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



Model Implementation

Cross Validation on Logistic Regression:

```
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
```

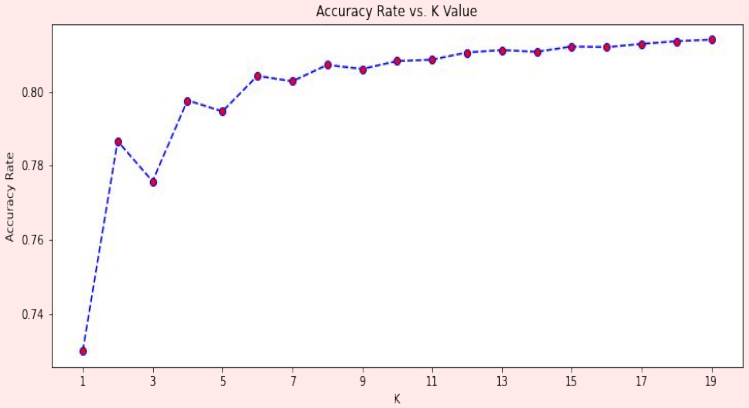
```
[CV]  ................................................................
[CV] ............. , accuracy=(train=0.809, test=0.807), total=   0.1s
```

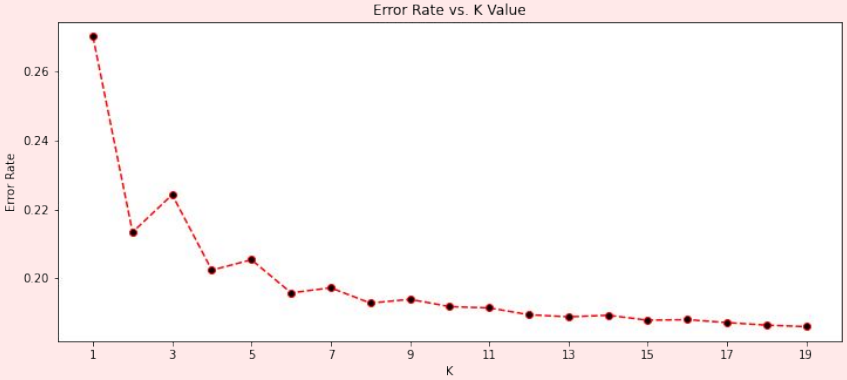
```
[Parallel(n_jobs=1)]: Done   4 out of   4 | elapsed:    0.3s remaining:    0.0s
[Parallel(n_jobs=1)]: Done   5 out of   5 | elapsed:    0.4s remaining:    0.0s
[Parallel(n_jobs=1)]: Done   5 out of   5 | elapsed:    0.4s finished
```

Model Implementation

K Near Neighbor Classifier:

- Few features are selected and made a separate dataframe.
- The same data is scaled and used for the model implementation.
- From evaluation 15 is as selected K value.
- The accuracy score on the test dataset is 81%
- Defining the predicted values form the model.






```
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
```

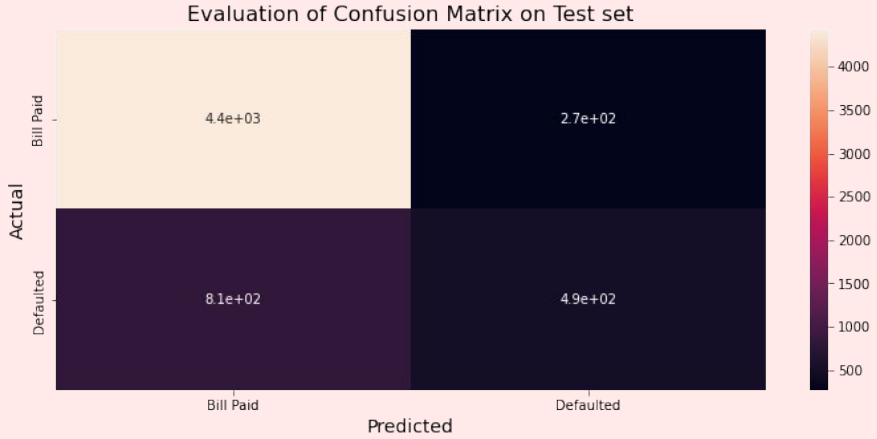
Model Implementation

Random Forest Classifier:

- The same scaled data was used for the model implementation.
- Fit the trained dataset to the model.
- The accuracy score on the test dataset is 82.00%
- Defining the predicted values form the model.

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.
- We have more number of defaulters from female customers.
- 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.
- We have customers from the 21 to 79 age group. Customers of 29 age are more in number but there is not much difference in the age criterion in defaulting.
- Some of the bill amount histplot are extended towards the left it is assumed that there are some advance payments.

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

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

From this, we can say that the model should be trained with the Random Forest Classifier, which can predict the defaulters with an accuracy of 82.00%

Thank You