“JAIHANUMANJI”

Statistical And Machine Learning

Individual Assignment

Credit Card Default

Submitted To:
Professor Minh Phan

Submitted By:
Aazad Ghoslya

Date: 31-March-2022

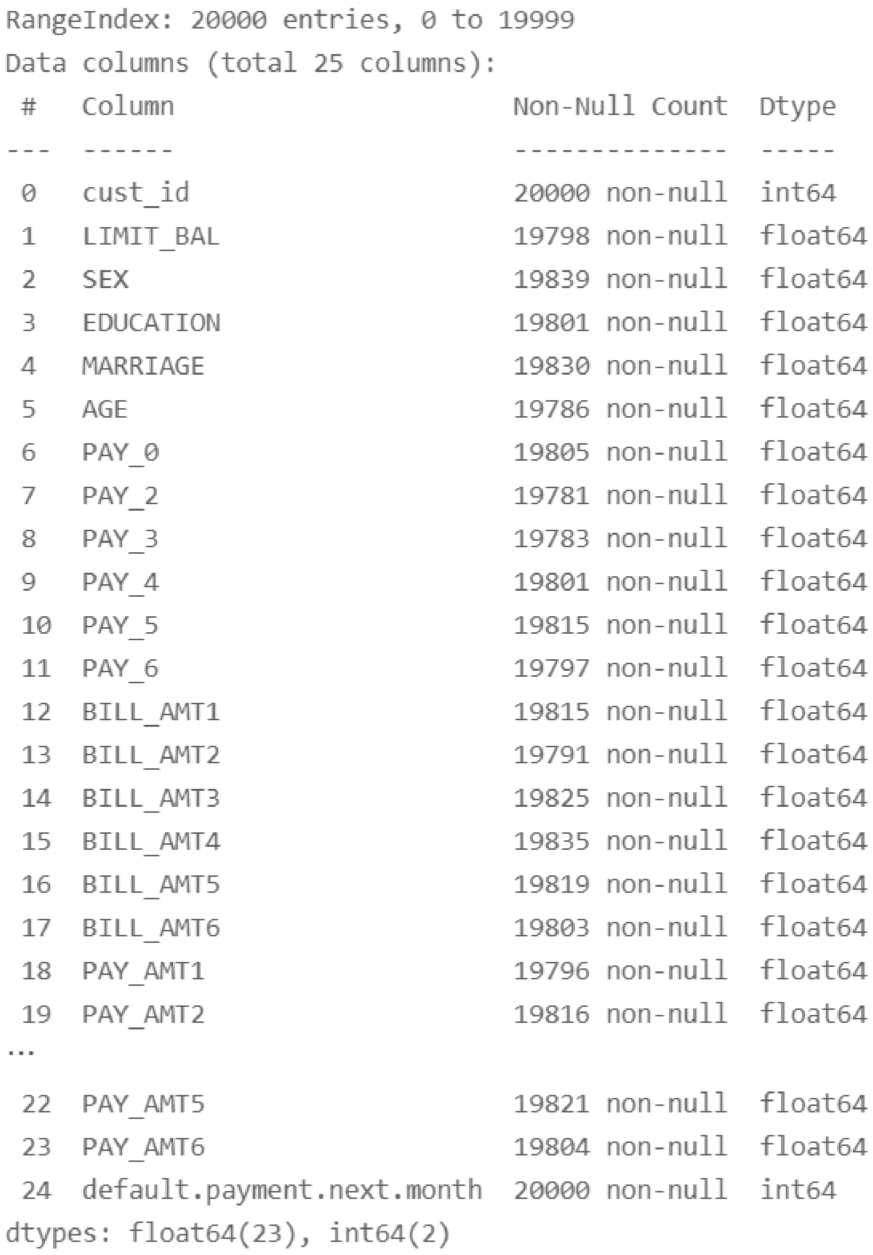
Objective: The main objective of this project is to setup a proper machine learning pipeline and implement the machine learning models on the given credit card dataset.

The machine learning algorithms used are:

- Logistic Regression
- KNN
- Decision Tree
- Random Forest
- Gradient Boosting

Credit Card Data Set:

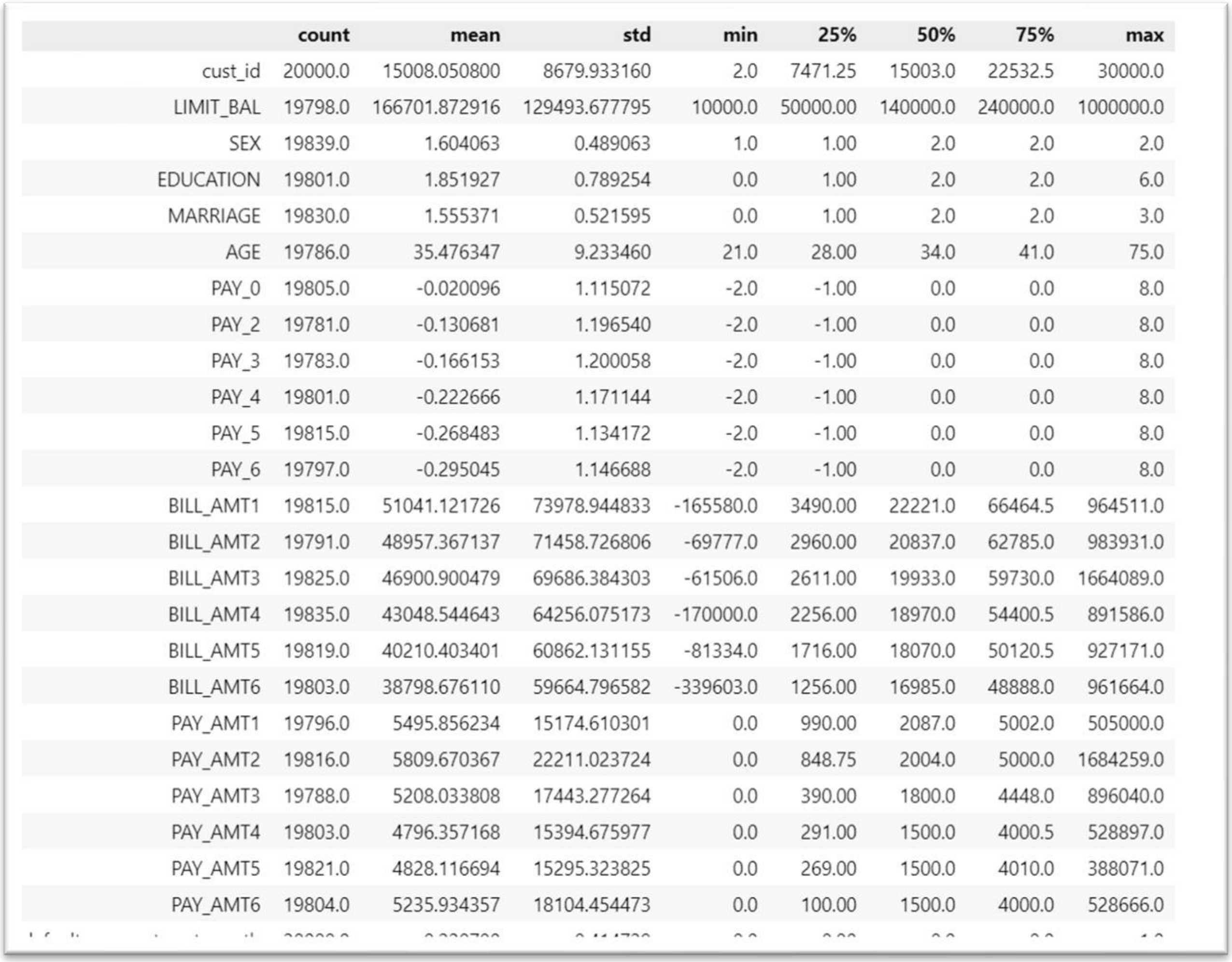
The data set consists different data of bank customers associated with credit card. The data set consist of 20,000 observations and 25 variables.



```
RangeIndex: 20000 entries, 0 to 19999
Data columns (total 25 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   cust_id                     20000 non-null  int64
 1   LIMIT_BAL                   19798 non-null  float64
 2   SEX                         19839 non-null  float64
 3   EDUCATION                   19801 non-null  float64
 4   MARRIAGE                    19830 non-null  float64
 5   AGE                         19786 non-null  float64
 6   PAY_0                       19805 non-null  float64
 7   PAY_2                       19781 non-null  float64
 8   PAY_3                       19783 non-null  float64
 9   PAY_4                       19801 non-null  float64
 10  PAY_5                       19815 non-null  float64
 11  PAY_6                       19797 non-null  float64
 12  BILL_AMT1                   19815 non-null  float64
 13  BILL_AMT2                   19791 non-null  float64
 14  BILL_AMT3                   19825 non-null  float64
 15  BILL_AMT4                   19835 non-null  float64
 16  BILL_AMT5                   19819 non-null  float64
 17  BILL_AMT6                   19803 non-null  float64
 18  PAY_AMT1                    19796 non-null  float64
 19  PAY_AMT2                    19816 non-null  float64
…
 22  PAY_AMT5                    19821 non-null  float64
 23  PAY_AMT6                    19804 non-null  float64
 24  default.payment.next.month  20000 non-null  int64
dtypes: float64(23), int64(2)
```

Data Summary:

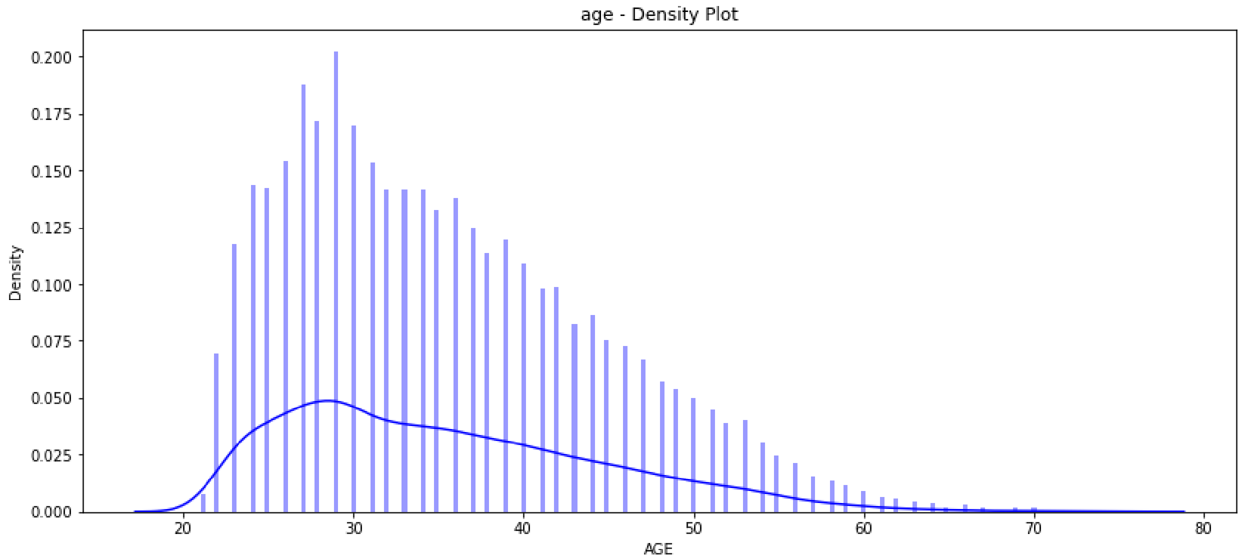
- The below table provides us the summary statistics of the entire dataset.
- The average value for the credit card is 166701. The standard deviation is unusually large as we can observe the maximum value as 1000000.
- The average age of the customers is 35.



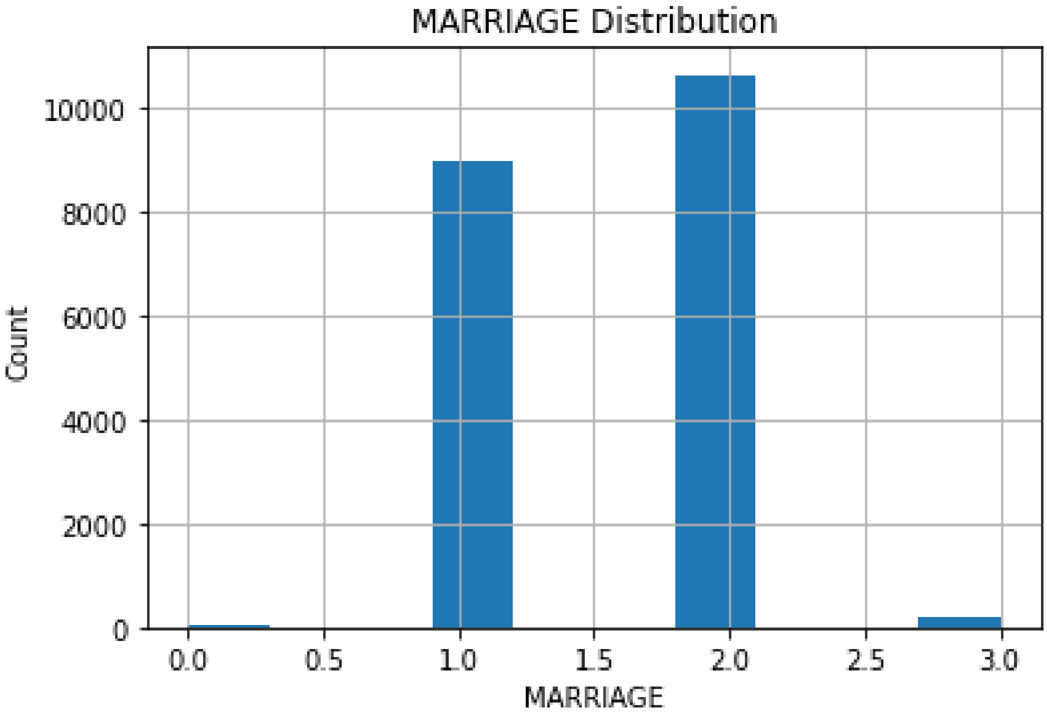
	count	mean	std	min	25%	50%	75%	max
cust_id	20000.0	15008.050800	8679.933160	2.0	7471.25	15003.0	22532.5	30000.0
LIMIT_BAL	19798.0	166701.872916	129493.677795	10000.0	50000.00	140000.0	240000.0	1000000.0
SEX	19839.0	1.604063	0.489063	1.0	1.00	2.0	2.0	2.0
EDUCATION	19801.0	1.851927	0.789254	0.0	1.00	2.0	2.0	6.0
MARRIAGE	19830.0	1.555371	0.521595	0.0	1.00	2.0	2.0	3.0
AGE	19786.0	35.476347	9.233460	21.0	28.00	34.0	41.0	75.0
PAY_0	19805.0	-0.020096	1.115072	-2.0	-1.00	0.0	0.0	8.0
PAY_2	19781.0	-0.130681	1.196540	-2.0	-1.00	0.0	0.0	8.0
PAY_3	19783.0	-0.166153	1.200058	-2.0	-1.00	0.0	0.0	8.0
PAY_4	19801.0	-0.222666	1.171144	-2.0	-1.00	0.0	0.0	8.0
PAY_5	19815.0	-0.268483	1.134172	-2.0	-1.00	0.0	0.0	8.0
PAY_6	19797.0	-0.295045	1.146688	-2.0	-1.00	0.0	0.0	8.0
BILL_AMT1	19815.0	51041.121726	73978.944833	-165580.0	3490.00	22221.0	66464.5	964511.0
BILL_AMT2	19791.0	48957.367137	71458.726806	-69777.0	2960.00	20837.0	62785.0	983931.0
BILL_AMT3	19825.0	46900.900479	69686.384303	-61506.0	2611.00	19933.0	59730.0	1664089.0
BILL_AMT4	19835.0	43048.544643	64256.075173	-170000.0	2256.00	18970.0	54400.5	891586.0
BILL_AMT5	19819.0	40210.403401	60862.131155	-81334.0	1716.00	18070.0	50120.5	927171.0
BILL_AMT6	19803.0	38798.676110	59664.796582	-339603.0	1256.00	16985.0	48888.0	961664.0
PAY_AMT1	19796.0	5495.856234	15174.610301	0.0	990.00	2087.0	5002.0	505000.0
PAY_AMT2	19816.0	5809.670367	22211.023724	0.0	848.75	2004.0	5000.0	1684259.0
PAY_AMT3	19788.0	5208.033808	17443.277264	0.0	390.00	1800.0	4448.0	896040.0
PAY_AMT4	19803.0	4796.357168	15394.675977	0.0	291.00	1500.0	4000.5	528897.0
PAY_AMT5	19821.0	4828.116694	15295.323825	0.0	269.00	1500.0	4010.0	388071.0
PAY_AMT6	19804.0	5235.934357	18104.454473	0.0	100.00	1500.0	4000.0	528666.0

Data Visualization and Analysis:

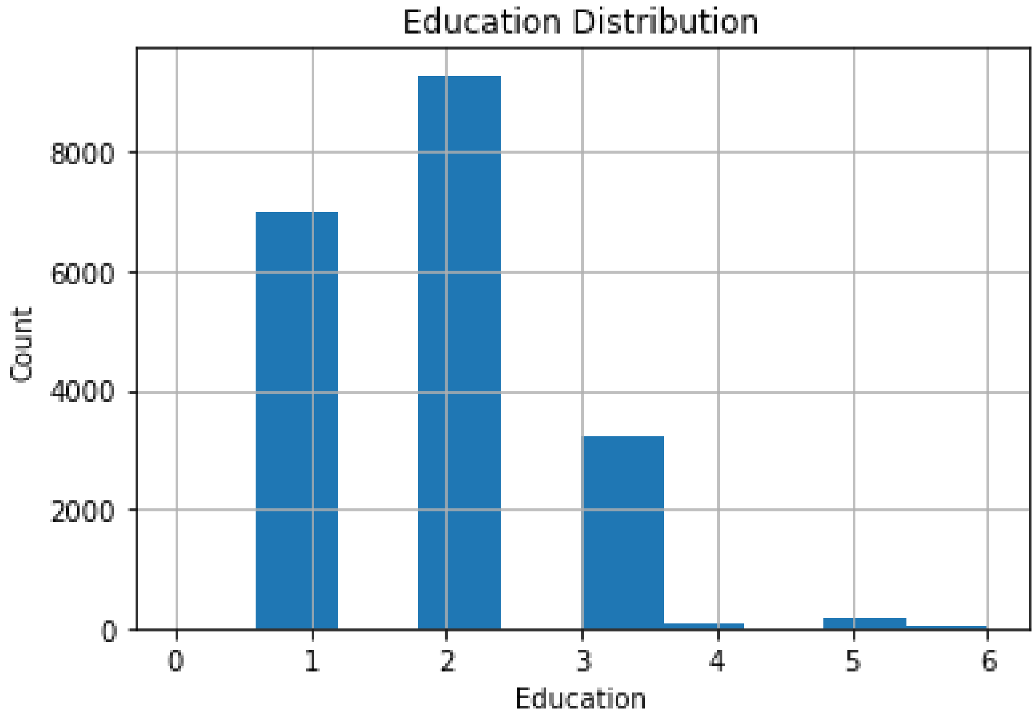
Age: This variable consists of the information regarding the age. This plot describes the distribution of the age of all customers. The maximum of the customers is with the age 39.



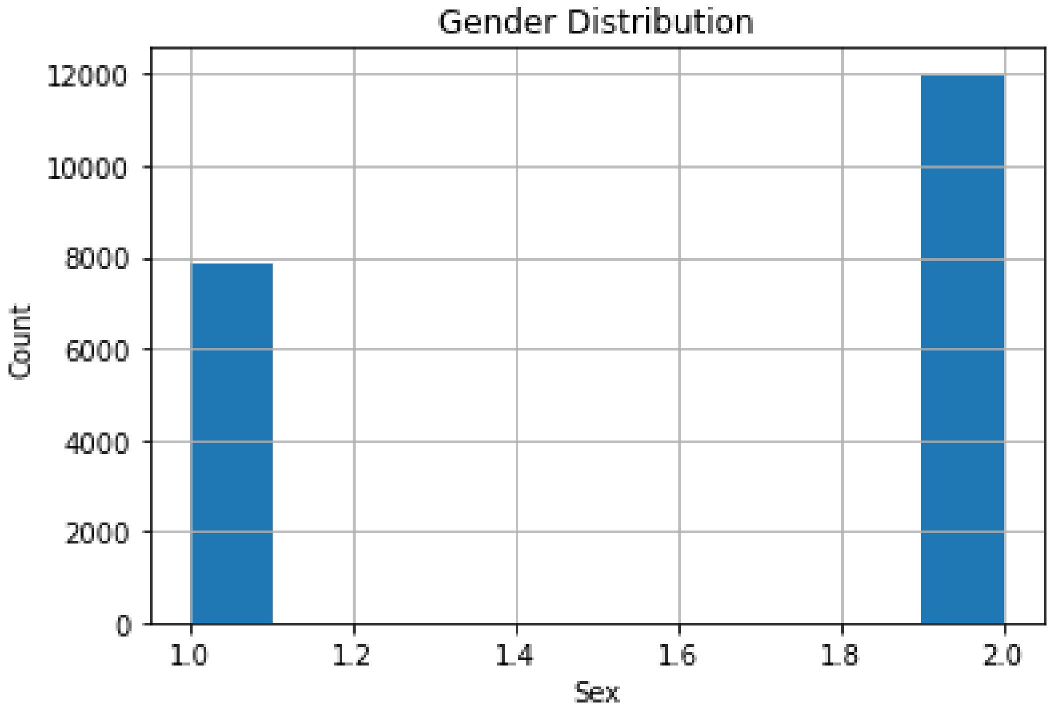
Marriage: This variable consists of the information regarding marriage status, consists of 5 unique values [0,1,2, 3, NA]. The maximum number of observations is with value 2.



Education: This variable consists of the information regarding education level, consists of 7 unique values [3., 2., 1., nan, 5., 4., 6., 0.]. The maximum number of observations is with value 2.

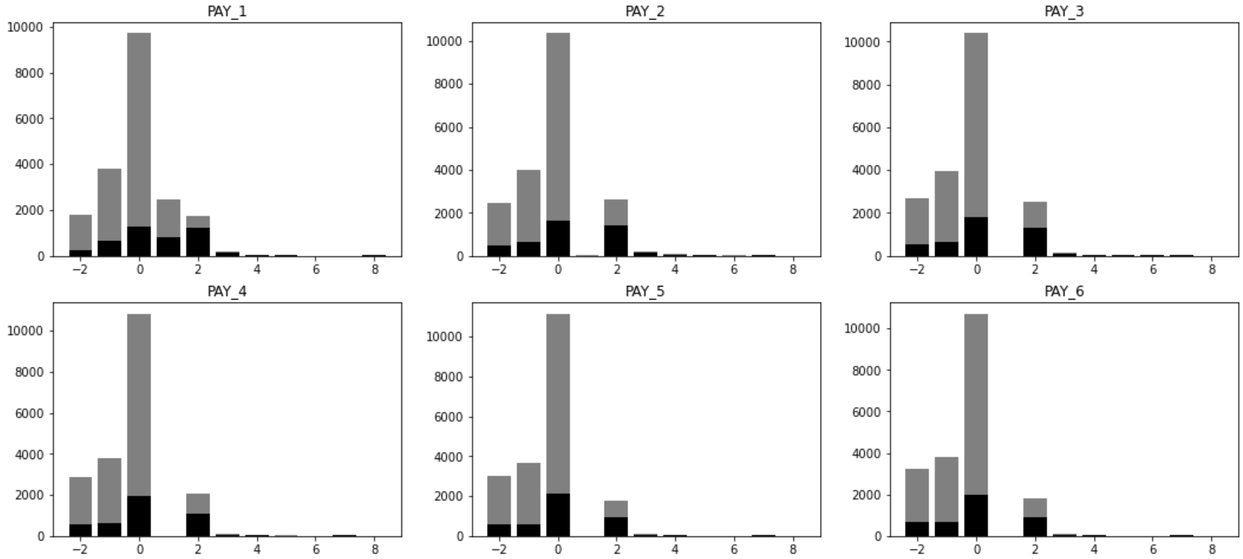


Gender: This variable consists of the gender information, 3 different values [1., 2., nan]. The maximum number of observations is with value 2.



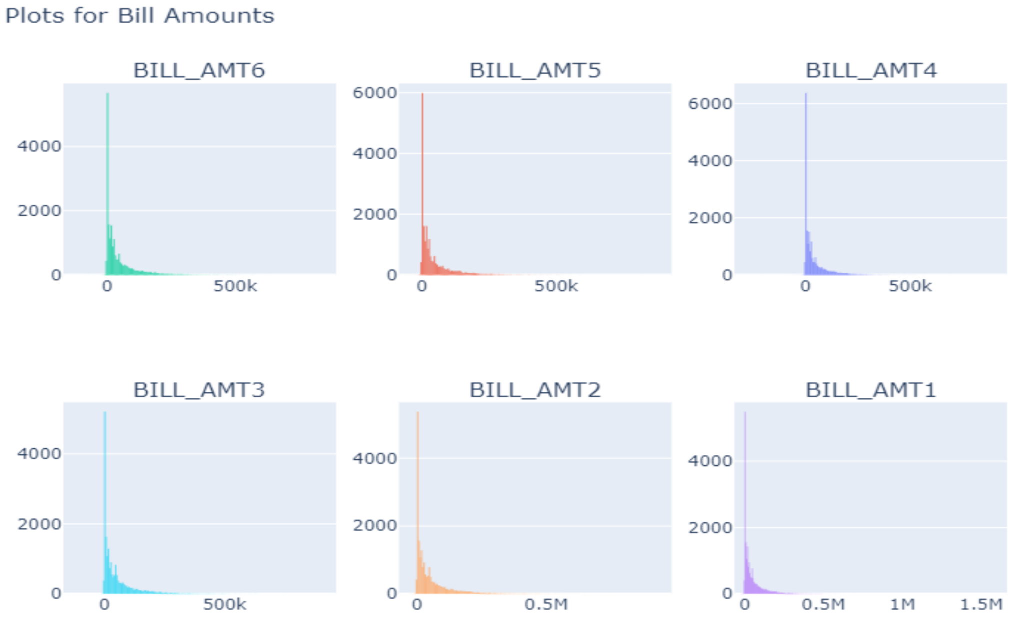
Pay:

This plot shows us the count of defaulters and non-defaulters for different months from PAY_1 to PAY_6.



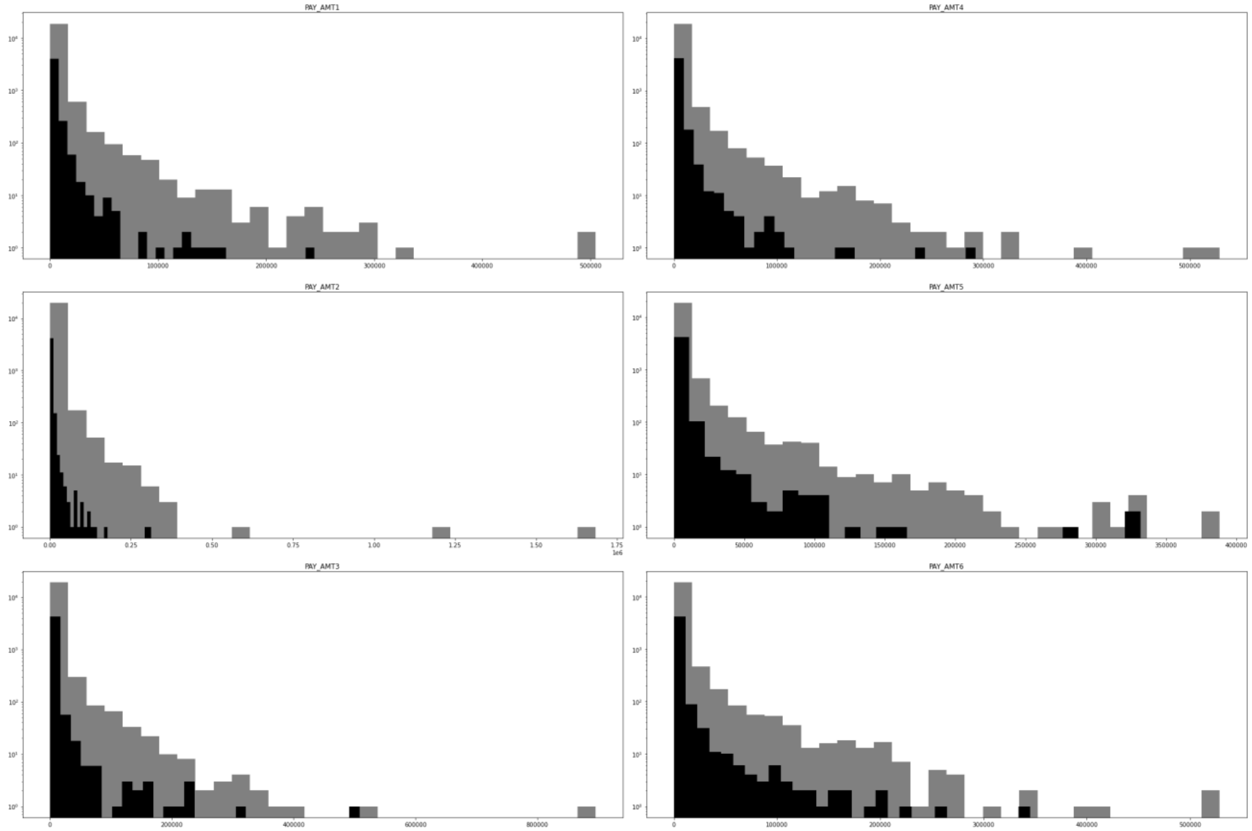
Bill Amount:

Distribution of bill amount for different month for both defaulters and non-defaulters from BILL_AMT1 to BILL_AMT6.

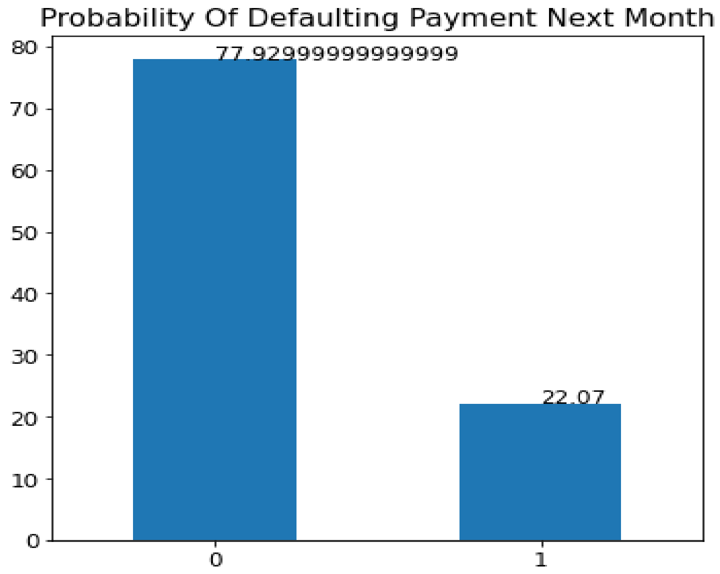


Payment Status:

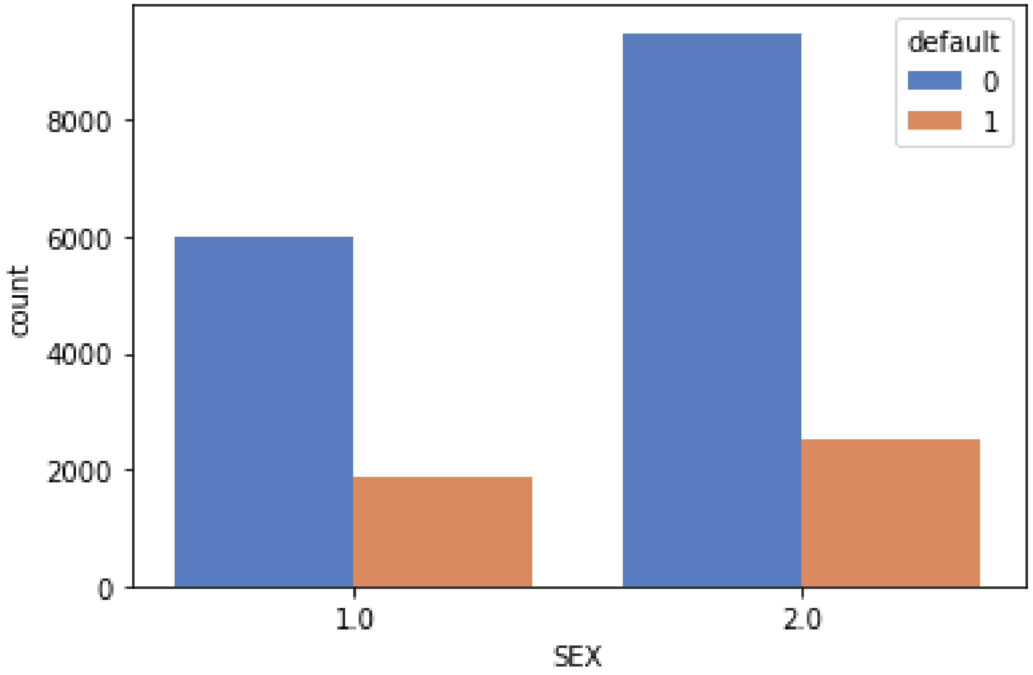
Distribution of payment amount for different months for defaulters and non-defaulters.



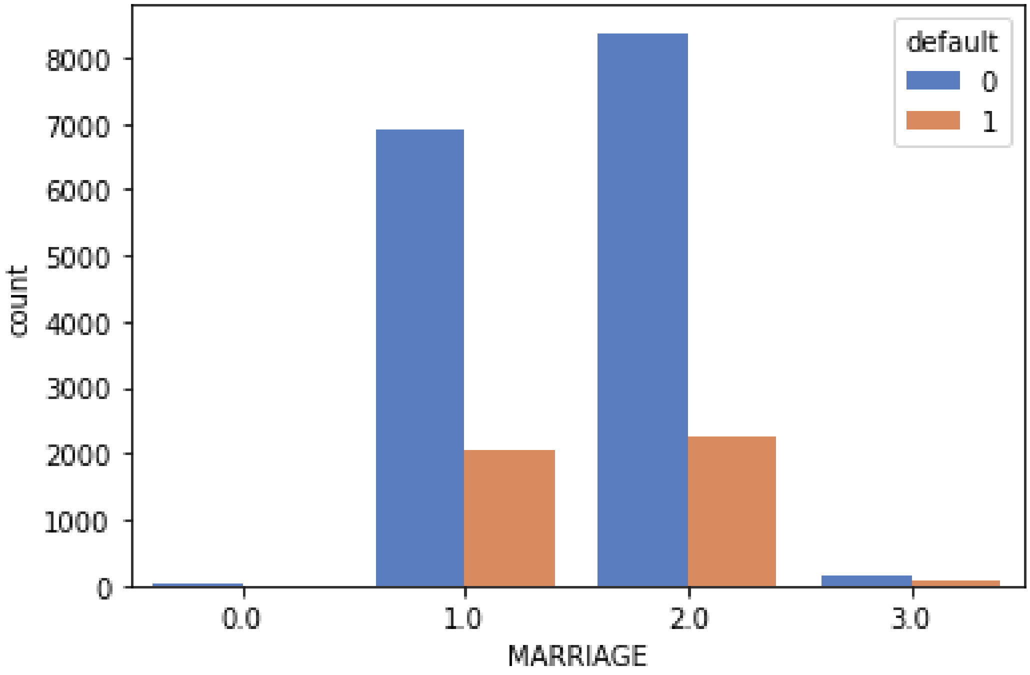
Default Pay (Target Variable): This is our target variable consist of information regarding default or not with the values of 1 and 0. The plot below shows the probability of being default and non-default.



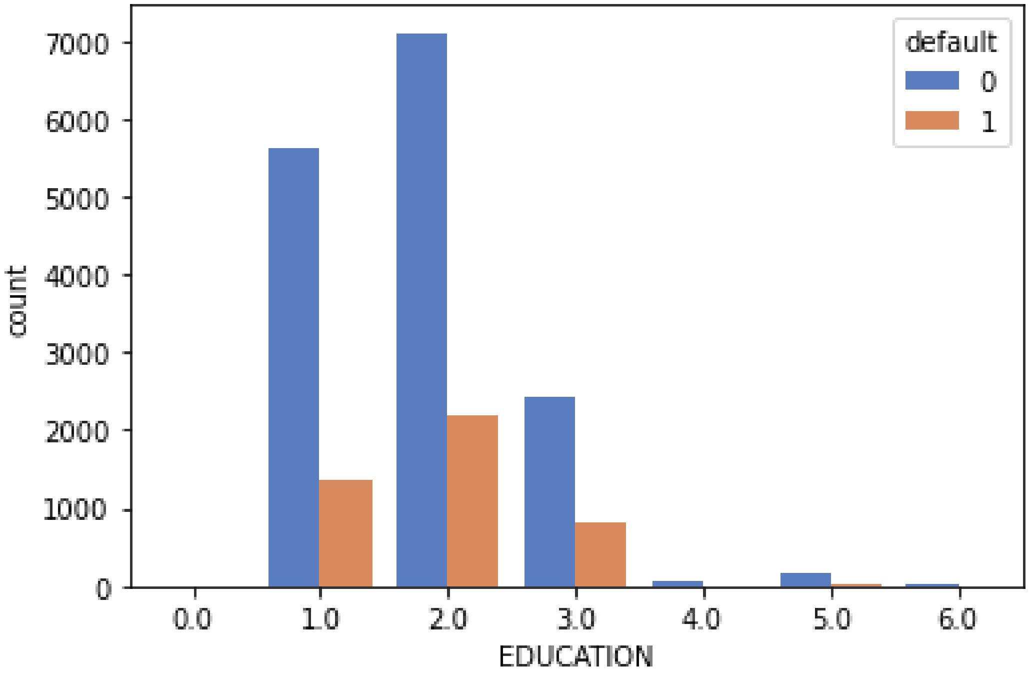
Defaulter v/s Gender: The ratio of default and non-default based on the gender.



Defaulter v/s Marriage: The ratio of default and non-default based on the Marriage status.



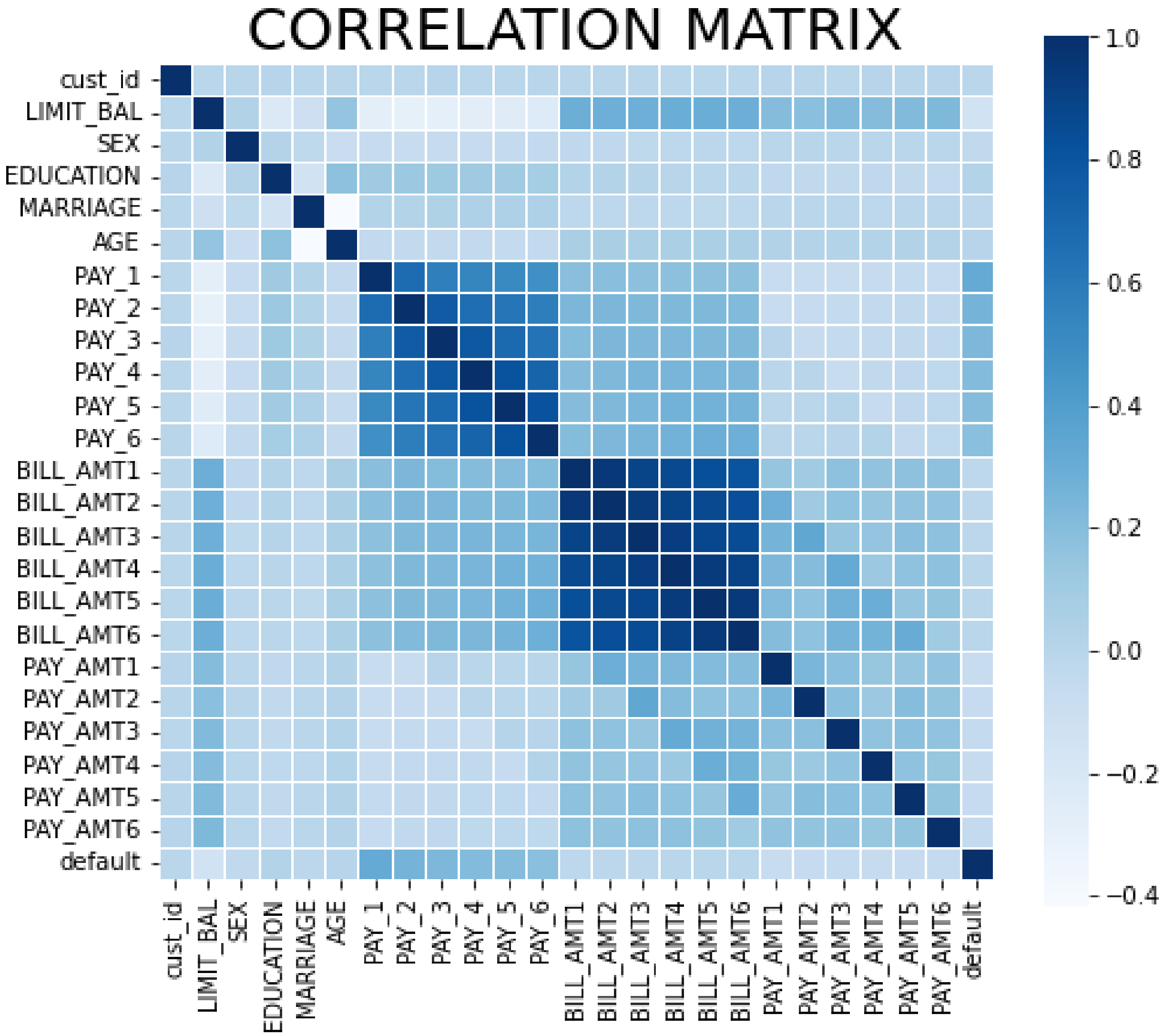
Defaulter v/s Education: The ratio of default and non-default based on the education status.



Correlation Between all the features:

Correlation:

- It is used to refer how close two variables are with each other to have a linear relationship with each other.
- Features with high correlation are more linearly dependent with each other and have same effect on the dependent variable so instead of using both we can drop either of them, helpful in feature selection.
- From this correlation matrix we can observe that BILL_AMT are highly positively correlated, payment amount is not correlated and multicollinearity exist between payment status.

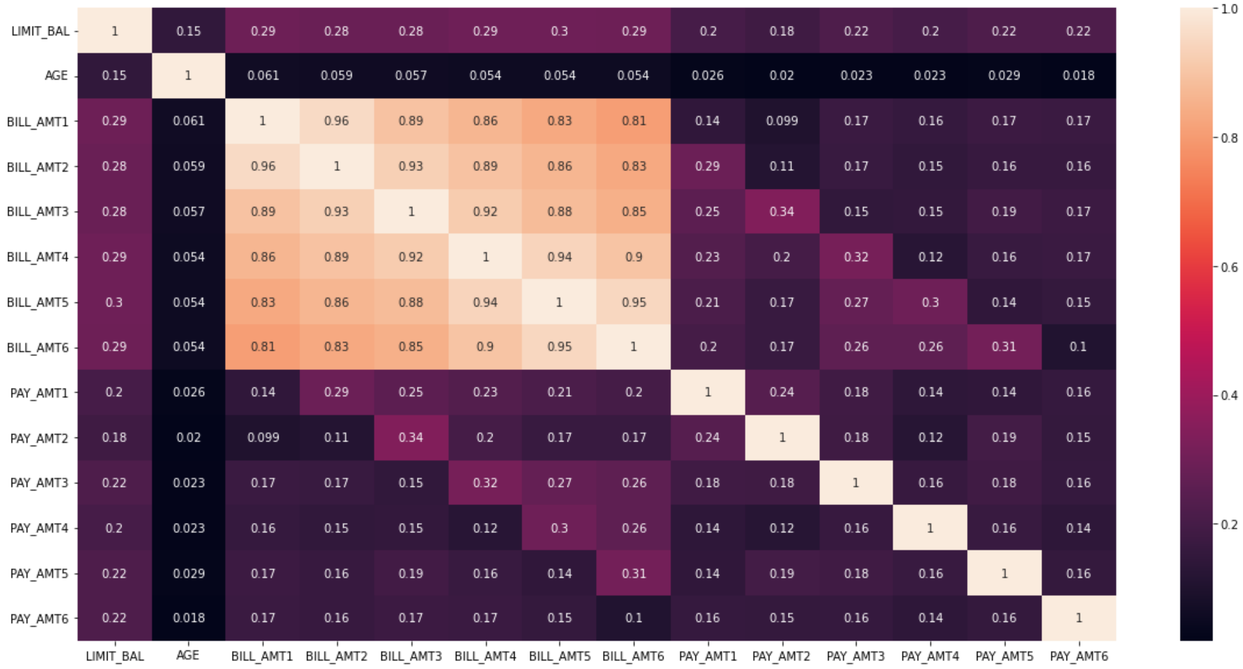


Data Preprocessing:

- Check for the NA values and there are lots of NA values in every variable.
- Separating the variables as numerical and categorical.
- Replacing the NA values separately.
    - Replacing NA values with mode for all categorical variables.
    - Replacing NA values with median for all numerical variables.
- One hot encoding for all the categorical variables.
- Scaling all the features of numerical variables.

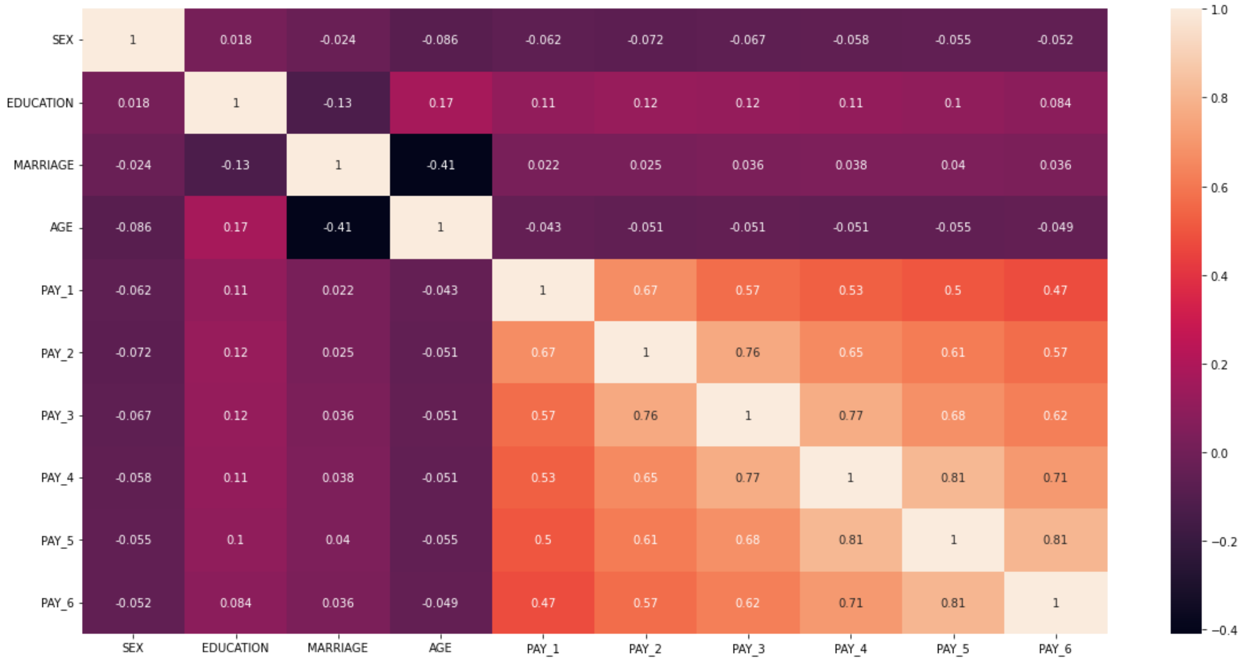
Correlation:

Numerical Variables: This is used to obtain the correlation among the numerical variables and choose the best from them to reduce the number of features.



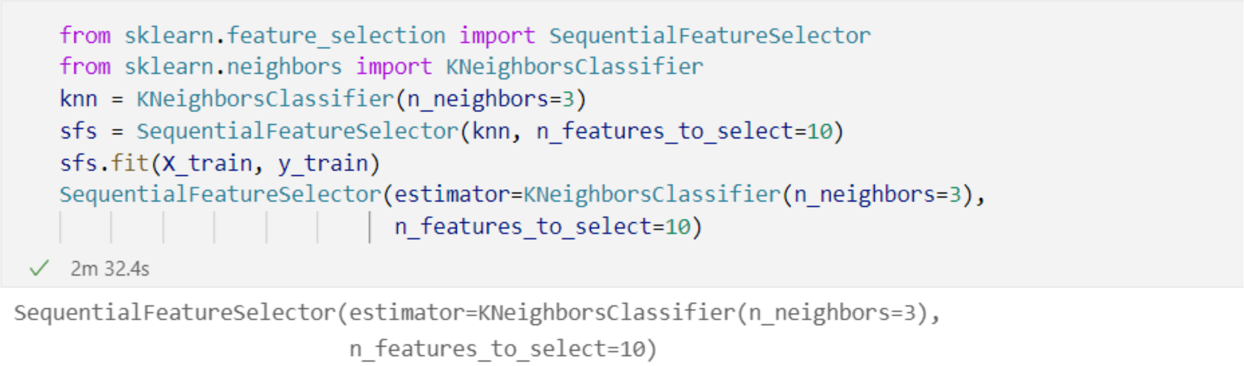
	LIMIT_BAL	AGE	BILL_AMT1	BILL_AMT2	BILL_AMT3	BILL_AMT4	BILL_AMT5	BILL_AMT6	PAY_AMT1	PAY_AMT2	PAY_AMT3	PAY_AMT4	PAY_AMT5	PAY_AMT6
LIMIT_BAL	1	0.15	0.29	0.28	0.28	0.29	0.3	0.29	0.2	0.18	0.22	0.2	0.22	0.22
AGE	0.15	1	0.061	0.059	0.057	0.054	0.054	0.054	0.026	0.02	0.023	0.023	0.029	0.018
BILL_AMT1	0.29	0.061	1	0.96	0.89	0.86	0.83	0.81	0.14	0.099	0.17	0.16	0.17	0.17
BILL_AMT2	0.28	0.059	0.96	1	0.93	0.89	0.86	0.83	0.29	0.11	0.17	0.15	0.16	0.16
BILL_AMT3	0.28	0.057	0.89	0.93	1	0.92	0.88	0.85	0.25	0.34	0.15	0.15	0.19	0.17
BILL_AMT4	0.29	0.054	0.86	0.89	0.92	1	0.94	0.9	0.23	0.2	0.32	0.12	0.16	0.17
BILL_AMT5	0.3	0.054	0.83	0.86	0.88	0.94	1	0.95	0.21	0.17	0.27	0.3	0.14	0.15
BILL_AMT6	0.29	0.054	0.81	0.83	0.85	0.9	0.95	1	0.2	0.17	0.26	0.26	0.31	0.1
PAY_AMT1	0.2	0.026	0.14	0.29	0.25	0.23	0.21	0.2	1	0.24	0.18	0.14	0.14	0.16
PAY_AMT2	0.18	0.02	0.099	0.11	0.34	0.2	0.17	0.17	0.24	1	0.18	0.12	0.19	0.15
PAY_AMT3	0.22	0.023	0.17	0.17	0.15	0.32	0.27	0.26	0.18	0.18	1	0.16	0.18	0.16
PAY_AMT4	0.2	0.023	0.16	0.15	0.15	0.12	0.3	0.26	0.14	0.12	0.16	1	0.16	0.14
PAY_AMT5	0.22	0.029	0.17	0.16	0.19	0.16	0.14	0.31	0.14	0.19	0.18	0.16	1	0.16
PAY_AMT6	0.22	0.018	0.17	0.16	0.17	0.17	0.15	0.1	0.16	0.15	0.16	0.14	0.16	1

Categorical Variables: This is used to obtain the correlation among the Categorical variables and choose the best from them to reduce the number of features.



Feature scaling: Scaling all the numerical variables to attain the same range for all the features and to handle highly varying magnitude and values.

Feature Selection: I have used sequential feature selection method using KNN with 3 neighbors to get the 10 best features.

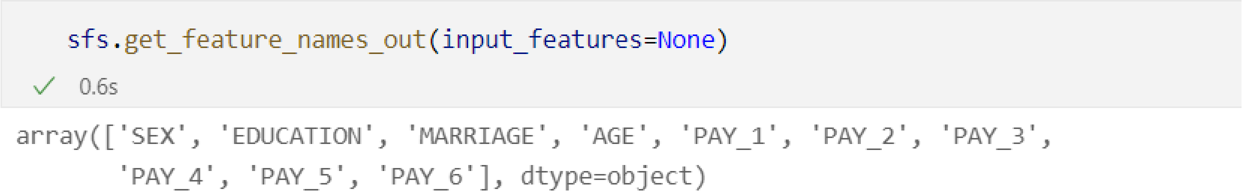


```
from sklearn.feature_selection import SequentialFeatureSelector
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors=3)
sfs = SequentialFeatureSelector(knn, n_features_to_select=10)
sfs.fit(X_train, y_train)
SequentialFeatureSelector(estimator=KNeighborsClassifier(n_neighbors=3),
                          n_features_to_select=10)
```

✓ 2m 32.4s

```
SequentialFeatureSelector(estimator=KNeighborsClassifier(n_neighbors=3),
                          n_features_to_select=10)
```

The top 10 features are:



```
sfs.get_feature_names_out(input_features=None)
```

✓ 0.6s

```
array(['SEX', 'EDUCATION', 'MARRIAGE', 'AGE', 'PAY_1', 'PAY_2', 'PAY_3',
       'PAY_4', 'PAY_5', 'PAY_6'], dtype=object)
```

I am using these 10 features for both X_train and X_test and then fitting the models.

Machine Learning:

Machine learning is the field of study that gives computers the ability to learn without being explicitly learned. There are mainly 3 types exists.

1. Supervised Learning:
   - In this we already have a pre-defined data on which we train our model.
   - Based on the train data the algorithm tries to find the similar pattern on model test set.
2. Unsupervised Learning:
   - In this we don’t have any pre-defined data, we are only having simple data and we need to find some structure in the data.
   - The algorithm divides the data into different clusters of same type or pattern and that’s why it is called clustering problem.
3. Reinforcement Learning:
   - In this model tries to learn by hit and trial.
   - It performs similarly like we use to play a chess game: based on predefined steps model will give the output.

I have used 5 different machine learning algorithms.

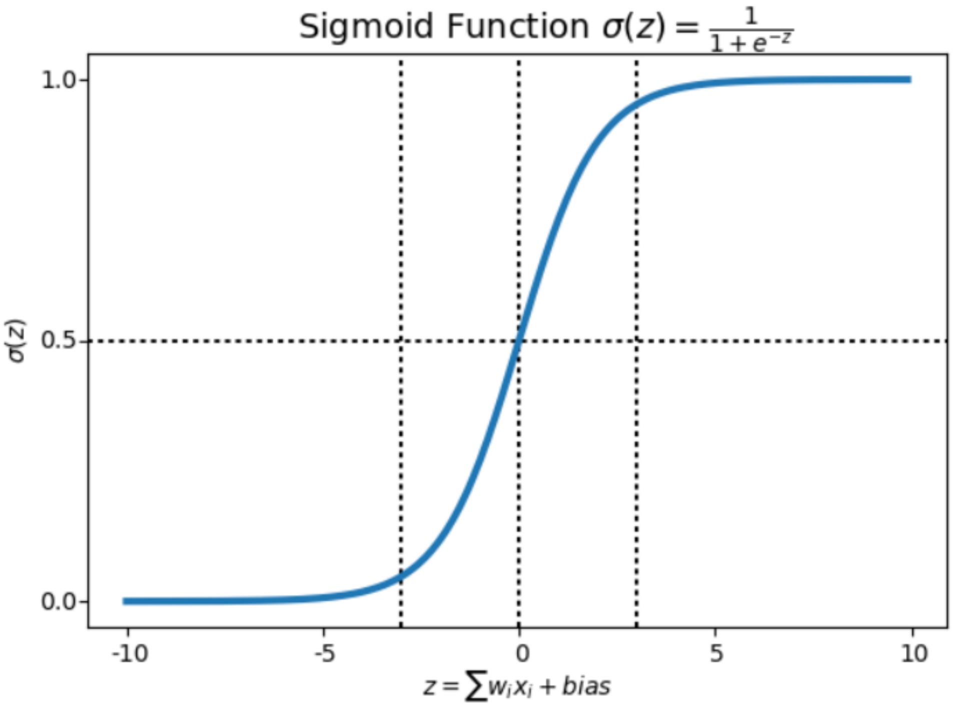
- Logistic Regression
- KNN
- Decision Tree
- Random Forest
- Gradient Boosting

1. Logistic Regression:

- It is an example of supervised learning.
- It is mainly used to solve the classification problems, when the output or dependent variable is categorical and most used is binary classification problem.
- It is used to predict the probability of binary event occurring saying that when an event is having binary outcome (yes/no).
- In case of logistic regression, the predicted outcome is discrete and restricted to a fixed number of values.
- The logistic regression main hypothesis is to limit the logistic function which is sigmoid function between 0 and 1.

$$0 \leq h_{\emptyset}(x) \leq 1$$

Hypothesis Expectation



Sigmoid Function Graph

- Logistic regression is represented by the following equation:

$$p(x) = \frac{e^{\beta_0+\beta_1 x_1+\beta_2 x_2+\cdots+\beta_n x_n}}{1+e^{\beta_0+\beta_1 x_1+\beta_2 x_2+\cdots+\beta_n x_n}}$$

Where: x_1 to x_n represents the predictors of the model.
P(x) the probability of response variable we are predicting.

Maximum Likelihood Function

$$\ell(\beta_0, \beta) = \prod_{i:y_i=1} p(x_i) \prod_{i:y_i=0} (1 - p(x_i))$$

- In logistic regression our main goal is to maximize the likelihood function by choosing the appropriate value of β_0 and β_1.
- The coefficients in maximum likelihood function can also be used to get the relationship between response variables and their corresponding predictors.

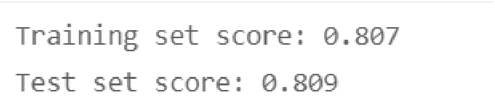
Types:

1. **Binary Logistic Regression**: When the number of possible outcomes is two like either A or B.
2. **Multinomial Logistic Regression**: When the number of possible outcomes is more than two say many outcomes like [A, B, C, D].
3. **Ordinal Logistic Regression**: When the number of possible outcomes are in an ordered way.

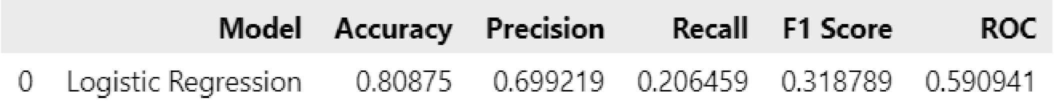
Implementation:

Model Fitting and Results:

I have fitted the model and have obtained following accuracy for test and train :

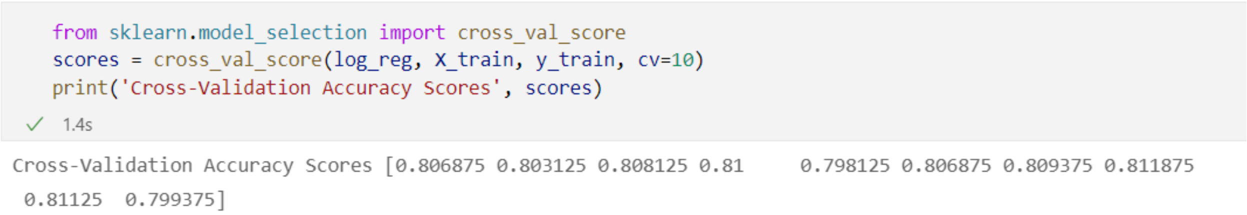


```
Training set score: 0.807
Test set score: 0.809
```



	Model	Accuracy	Precision	Recall	F1 Score	ROC
0	Logistic Regression	0.80875	0.699219	0.206459	0.318789	0.590941

Cross Validation: This is mainly used to cross check the accuracy obtained. It is repeated model fitting. Each fit is done on the major portion of data and then tested on the left-out portion. This is repeated until every observation is used for testing. I have used cross validation with 10 folds to find the best coefficients.



```
from sklearn.model_selection import cross_val_score
scores = cross_val_score(log_reg, X_train, y_train, cv=10)
print('Cross-Validation Accuracy Scores', scores)
```

✓ 1.4s

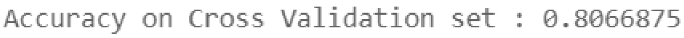
```
Cross-Validation Accuracy Scores [0.806875 0.803125 0.808125 0.81     0.798125 0.806875 0.809375 0.811875
 0.81125  0.799375]
```

We are getting almost same accuracy as we are getting before indicating that our model is neither under fit nor over fit.

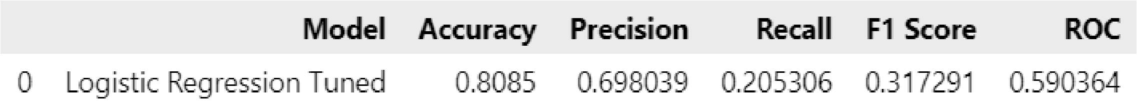
Grid Search:

In order to get the parameters right I used GridSerachCV. I passed the parameter space and test my model on this getting the ideal combination of parameters.For grid search I am using parameters {'C': [0.001, 0.01, 0.1, 1, 10], 'class_weight': [None, 'balanced'], 'penalty': ['l1', 'l2']}.

The results are:



```
Accuracy on Cross Validation set : 0.8066875
```

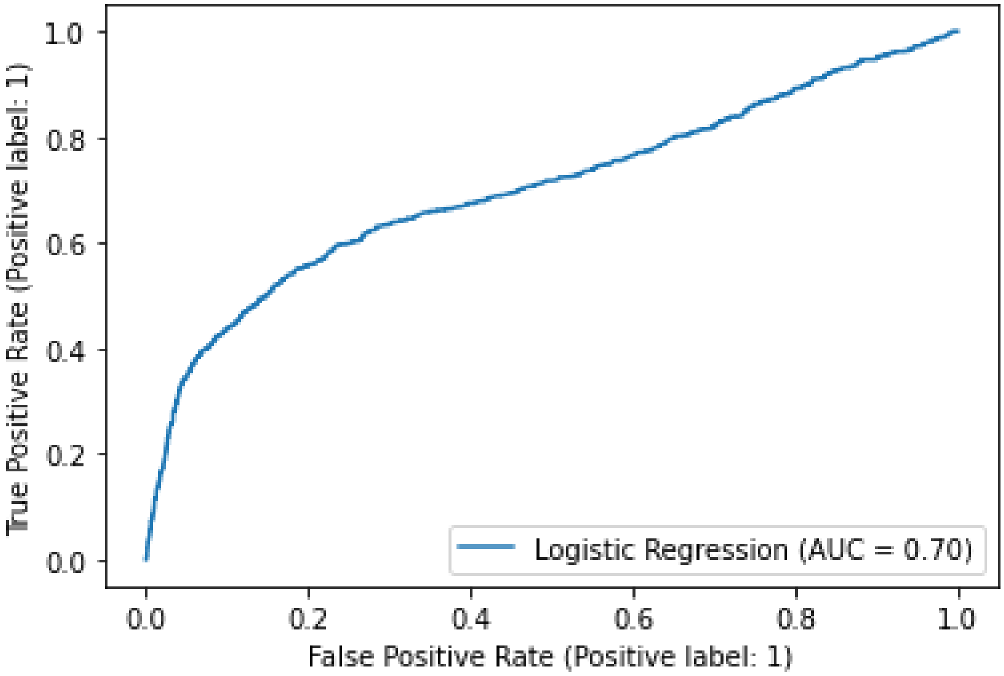


	Model	Accuracy	Precision	Recall	F1 Score	ROC
0	Logistic Regression Tuned	0.8085	0.698039	0.205306	0.317291	0.590364

ROC Curve:

The **receiver operating characteristic (ROC) curve** is frequently used for evaluating the performance of binary classification algorithms.

The ROC curve is produced by calculating and plotting the **true positive rate** against the **false positive rate.**



AUC: From the above plot we can observe that we get an AUC of 70% whereas the AUC before is 0.59 which is 59%.

Advantages:

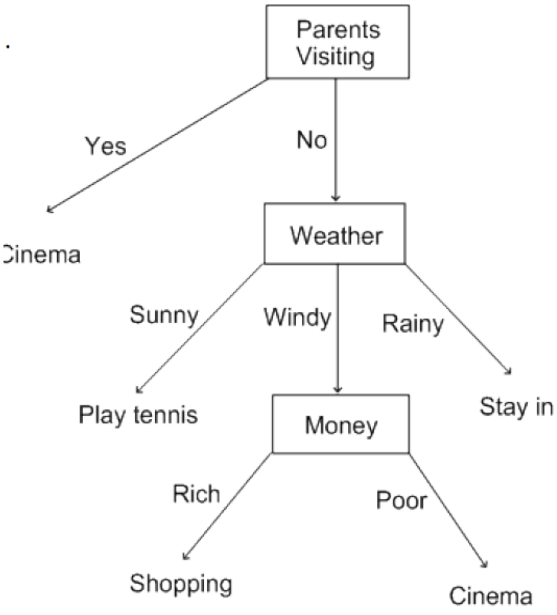
- Logistic Regression is easy to implement, Interpret and very efficient to train.
- It makes no assumption about distribution of classes in feature space.
- It can easily extend to multiple classes (multinomial logistic regression).
- It is very fast in classification of unknown records.
- Good accuracy and performs well with linearly separable dataset.

Disadvantages:

- When the number of features is more than the number of observations it may lead to overfitting.
- It constructs linear boundaries.
- The major limitation is the assumption of linearity between dependent and independent variables.
- We can not solve non-linear problems with logistic regression as it is having linear decision surface.

2. Decision Tree:

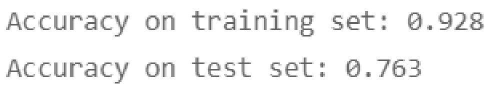
- It is a type of supervised machine learning where the split of data takes place based on certain parameter.
- The main goal in Decision Tree is to create a training model that we can use to predict the value of the target variable by learning simple decision rules inferred from train set.
- In Decision Tree, for predicting the target variable we start from the root of the tree and then compare the values of the root attribute with the trains attribute. Based on the comparison, we follow the branch corresponding to that value and move to the next node.
- Decision trees classify the examples by sorting them down the tree from the root to some leaf/terminal node. The leaves are the decisions or final outcomes, and the decision nodes are where the split of the data takes place.



Implementation:

Model Fitting and Results:

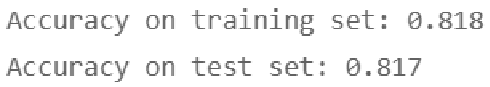
I have fitted the model using selected features and without any parameter tuning have obtained following results:



```
Accuracy on training set: 0.928
Accuracy on test set: 0.763
```

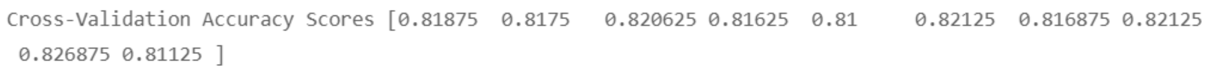
This shows that model is somewhere overfitting and the importance of parameter tuning in models.

On setting the value of max_depth as 3 the accuracy has been dropped as model seems to be perfectly fitted.



```
Accuracy on training set: 0.818
Accuracy on test set: 0.817
```

Cross Validation: It is repeated model fitting. Each fit is done on the major portion of data and then tested on the left-out portion. This is repeated until every observation is used for testing. I have used cross validation with 10 folds to find the best coefficients.

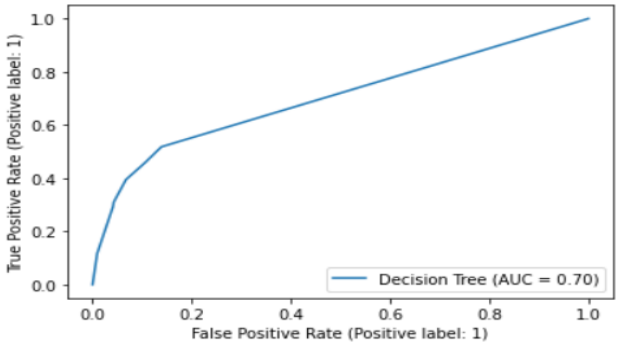


```
Cross-Validation Accuracy Scores [0.81875  0.8175   0.820625 0.81625  0.81     0.82125  0.816875 0.82125
 0.826875 0.81125 ]
```

After cross validation, I am getting almost same accuracy as we are getting before indicating that our model is neither under fit nor over fit.

ROC Curve:

The **receiver operating characteristic (ROC) curve** is frequently used for evaluating the performance of binary classification algorithms.
The ROC curve is produced by calculating and plotting the **true positive rate** against the **false positive rate.**



The normal AUC is 0.63 and the plot AUC value is around 0.70.

Types:

1. **Categorical Variable Decision Tree:** Decision tree with a categorical target variable.
2. **Continuous Variable Decision Tree:** Decision tree with a continuous target variable.

Advantages:

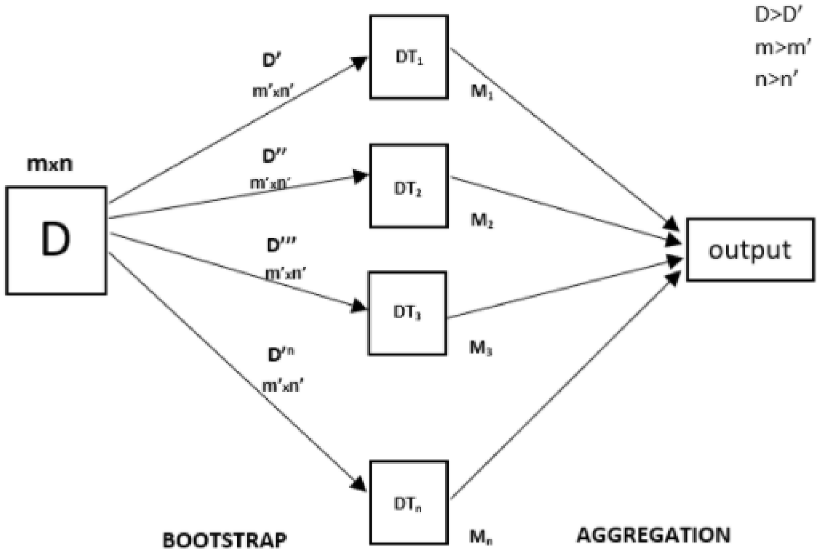
- Decision trees can generate understandable rules.
- In order perform classification less computation is needed.
- Can easily handle both continuous and categorical variables.
- Provides a clear view regarding the importance of field associated with classification and prediction.

Disadvantages:

- Decision trees are less appropriate for estimation tasks involving predicting the value of a continuous attribute.
- Decision trees are more likely to get error with classification problems involving small number of training set and many classes.
- Decision tree can be computationally expensive.

3. Random Forest:

- Random forest is an ensemble technique which can perform both regression and classification tasks with the use of multiple decision tree and a technique called Bootstrap and aggregation, commonly known as bagging.
- The main idea to come with an output using multiple decision trees instead of using a single decision tree.
- We use multiple decision trees as our base model and then we randomly perform sampling and feature sampling from the dataset, known as Bootstrap.

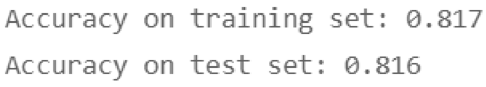


- The total number of predictive variables is specified such that at each node, the number of predictive variables are selected at random out of the total number of variables.
- The best split of these predictive variables is used to split the node. This value remains constant while we grow all the decision trees for the model.
- We then specify the number of trees we want in our random forest model, and each of these trees is grown as much as possible.
- After that, we fit the model and we can predict new data by giving the prediction as the class with most votes, after evaluating each new observation in each tree of the model.

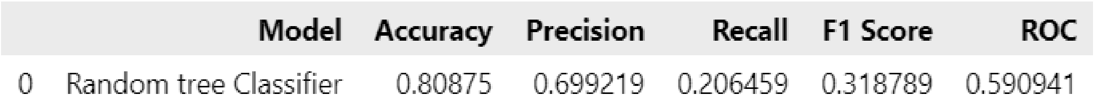
Implementation:

Model Fitting and Results:

I have fitted the model using selected features and have obtained following accuracy for test and train results:

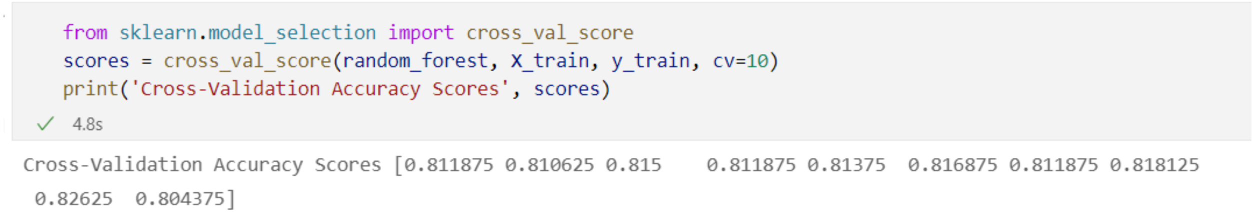


```
Accuracy on training set: 0.817
Accuracy on test set: 0.816
```



	Model	Accuracy	Precision	Recall	F1 Score	ROC
0	Random tree Classifier	0.80875	0.699219	0.206459	0.318789	0.590941

Cross Validation: It is repeated model fitting. Each fit is done on the major portion of data and then tested on the left-out portion. This is repeated until every observation is used for testing. I have used cross validation with 10 folds to find the best coefficients.



```
from sklearn.model_selection import cross_val_score
scores = cross_val_score(random_forest, X_train, y_train, cv=10)
print('Cross-Validation Accuracy Scores', scores)
```

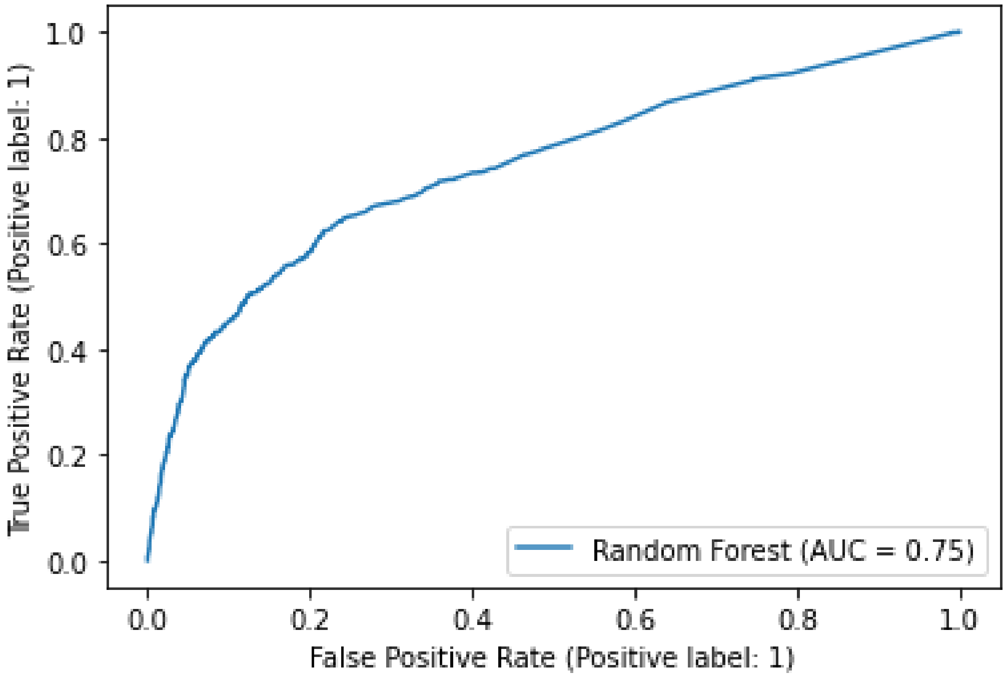
```
Cross-Validation Accuracy Scores [0.811875 0.810625 0.815    0.811875 0.81375  0.816875 0.811875 0.818125
 0.82625  0.804375]
```

After cross validation, I am getting almost same accuracy as we are getting before indicating that our model is neither under fit nor over fit.

ROC Curve:

The **receiver operating characteristic (ROC) curve** is frequently used for evaluating the performance of binary classification algorithms.

The ROC curve is produced by calculating and plotting the **true positive rate** against the **false positive rate.**



The normal AUC is 0.59 and the plot AUC value is around 0.75.

Advantages:

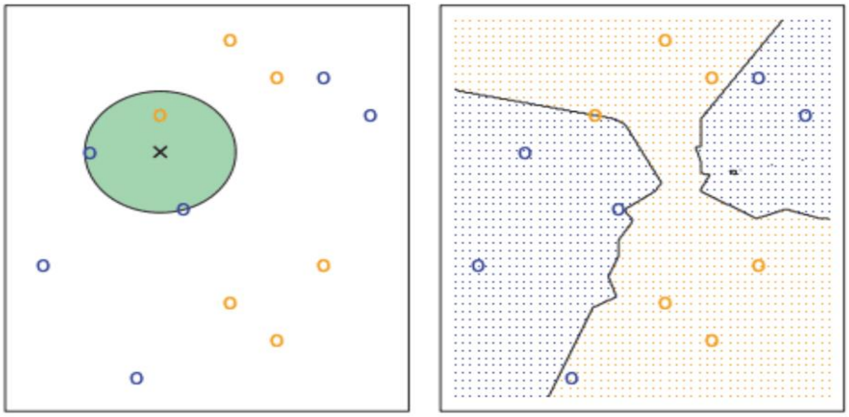
- It is based on bagging algorithm and uses ensemble technique, and it involves creation of many trees so overall reduces the overfitting and variance and gives high accuracy.
- Random forest works well with both categorical and continuous variables.
- Random forest can automatically handle missing values, robust to outliers and can handle them easily.
- It is very stable and less impacted by noise.

Disadvantages:

Random forest creates lots of trees and then combine their outputs which in turn makes it more complex than normal decision tree.
It requires more time to train in comparison to decision tree as the number of trees are more in case of random forest.

4. KNN:

- KNN also known as K nearest neighbor is a supervised and pattern classification algorithm which helps us to find the class to which our target variable belongs to when k nearest neighbors are chosen and distance between them is calculated.
- KNNs attempts to estimate the conditional distribution of Y given X, and classify a given observation (test value) to the class with highest estimated probability.
- It first identifies the k points in the training data that are closest to the target value and calculates the distance between all those categories. The target value will belong to the category with the least distance.



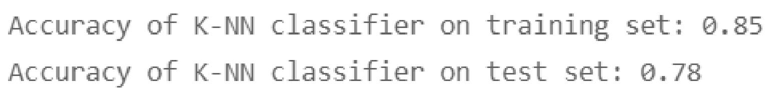
- Probability of classification of target variable is calculated using.

$$P_r(Y = j|X = x_0) = \frac{1}{K}\sum\nolimits_{i\,\in\,N_0} I(y_i = j)$$

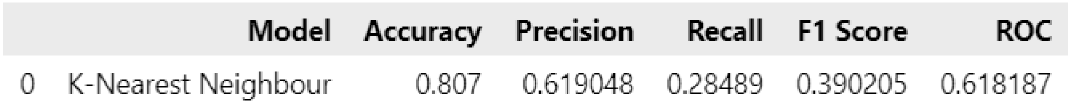
- The distance in KNN can be calculated using different ways:
  - Euclidean Method
  - Manhattan Method
  - Minkowski Method
- In KNN we firstly use to plot all the entries and after this when a new entry comes, we will set a value of k and based on this value of k the new entry will look for distance of nearest k values and then based on the distance the one with the smallest distance will g=have more probability and the new entry will be assigned to that.

Implementation:

Model Fitting and Results:
I have fitted the model using selected features and setting number of neighbors as 3, have obtained following results:

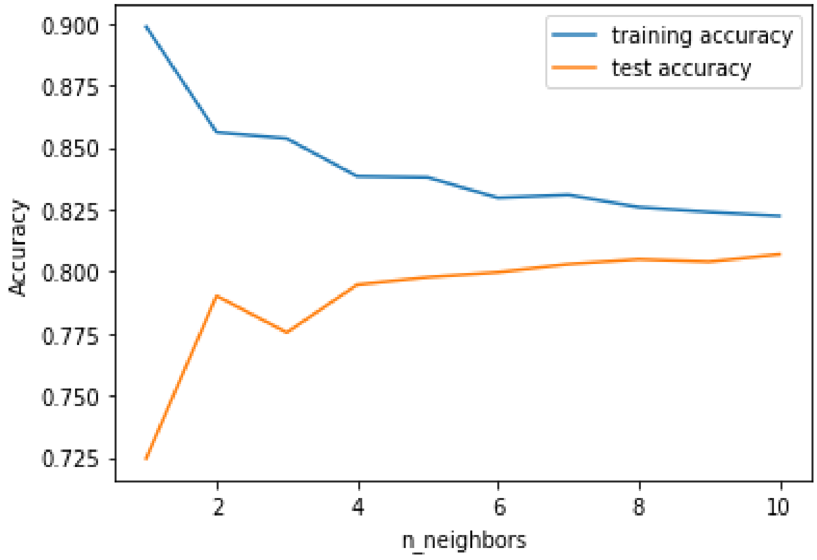


```
Accuracy of K-NN classifier on training set: 0.85
Accuracy of K-NN classifier on test set: 0.78
```

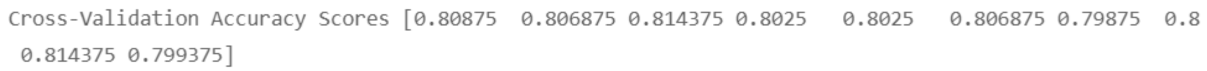


	Model	Accuracy	Precision	Recall	F1 Score	ROC
0	K-Nearest Neighbour	0.807	0.619048	0.28489	0.390205	0.618187

The training and test accuracy for KNN based on different value of neighbors are:



Cross Validation: It is repeated model fitting. Each fit is done on the major portion of data and then tested on the left-out portion. This is repeated until every observation is used for testing. I have used cross validation with 10 folds to find the best coefficients.

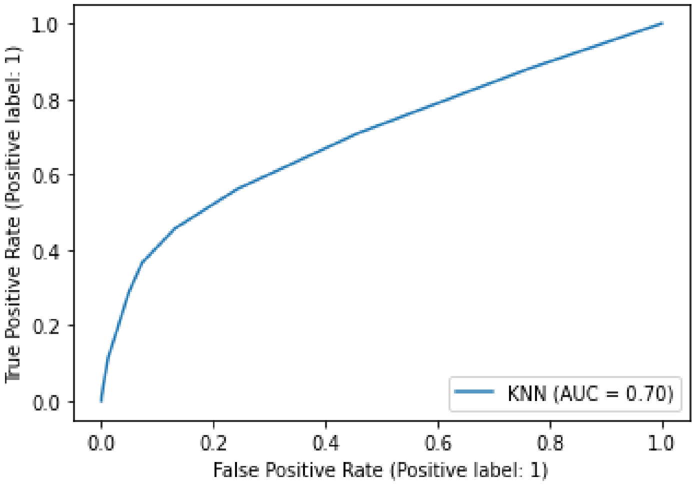


```
Cross-Validation Accuracy Scores [0.80875  0.806875 0.814375 0.8025   0.8025   0.806875 0.79875  0.8
 0.814375 0.799375]
```

After cross validation, I am getting almost same accuracy as we are getting before indicating that our model is neither under fit nor over fit.

ROC Curve:

The **receiver operating characteristic (ROC) curve** is frequently used for evaluating the performance of binary classification algorithms.
The ROC curve is produced by calculating and plotting the **true positive rate** against the **false positive rate.**



The normal AUC is 0.61 and the plot AUC value is around 0.70.

Advantages:

- It is uncomplicated and easier to apply.
- It is useful for no-linear data because it does make assumptions about the underlying data distribution.
- There are a smaller number of metrics to provide to algorithm value of k and distance metric.

Disadvantages:

- It is computationally expensive because the algorithm stores all the training data, and each new observation is classified by computing the distances between the nearest points which leads to increase in the cost of k nearest neighbors.
- It is difficult to work with the categorical variables.
- There is no method to get the best value of k. That may lead to overfitting.

5. Gradient Boosting:

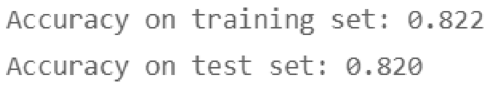
- Gradient Boosting is an extension of the boosting method. It uses gradient descent algorithm.
- An ensemble of trees is built one by one and then individual trees are summed sequentially. The next tree tries to minimize the difference between actual and predicted values of the previously built tree.
- **Gradient Descent** is an optimization algorithm which is used to get the minimum value of a differentiable function.
- It is used to minimize the cost function also known as Mean square error.

$$J(\theta_0, \theta_1) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^i - y^i))^2$$

- The main goal is getting the minimum value of θ_0, θ_1 to minimize $J(\theta_0, \theta_1)$.
- We start with some random θ_0, θ_1 and keep changing the value of θ_0, θ_1 until we obtain the minimum value of $J(\theta_0, \theta_1)$.
- **Learning Rate (α):** It is used to decide the length of a step, to achieve the local value of function.
  - If $\boldsymbol{\alpha}$ is too small, it makes the gradient descent slow.
  - If $\boldsymbol{\alpha}$ is too large, gradient descent can overshoot the minimum value also it may even fail to converge or diverge.
- **Derivative term ($\frac{\partial}{\partial \theta_j} j(\theta_0, \theta_1)$):** This helps us to decide that whether we are taking the steps in the right direction or not in order to achieve the min. value of the function.
- In Gradient descent we always follow simultaneous update for θ_0, θ_1.
- Algorithm begins by training a decision tree in which we assign equal weight to every observation.
- After evaluating the first tree, we increase the weights of those observations that are difficult to classify and lower the weights for those that are easy to classify.
- The second tree is therefore grown on this weighted data. Here, the idea is to improve upon the predictions of the first tree.
- Our new model is therefore *Tree 1 + Tree 2*. We then compute the classification error from this new 2-tree ensemble model and grow a third tree to predict the revised residuals.
- We repeat this process for a specified number of iterations. Subsequent trees help us to classify observations that are not well classified by the previous trees.

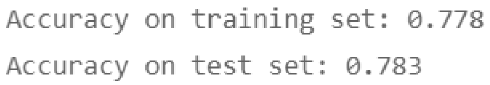
- Predictions of the final ensemble model is therefore the weighted sum of the predictions made by the previous tree models.

Implementation:
Model Fitting and Results:
I have fitted the model using selected features, and without setting any value for learning rate, have obtained following results:



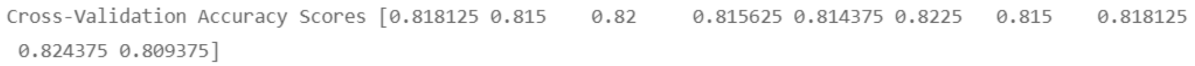
```
Accuracy on training set: 0.822
Accuracy on test set: 0.820
```

On setting the value of learning rate as 0.001 the accuracy has been changed. Showing the impact of α.



```
Accuracy on training set: 0.778
Accuracy on test set: 0.783
```

Cross Validation: It is repeated model fitting. Each fit is done on the major portion of data and then tested on the left-out portion. This is repeated until every observation is used for testing. I have used cross validation with 10 folds to find the best coefficients.

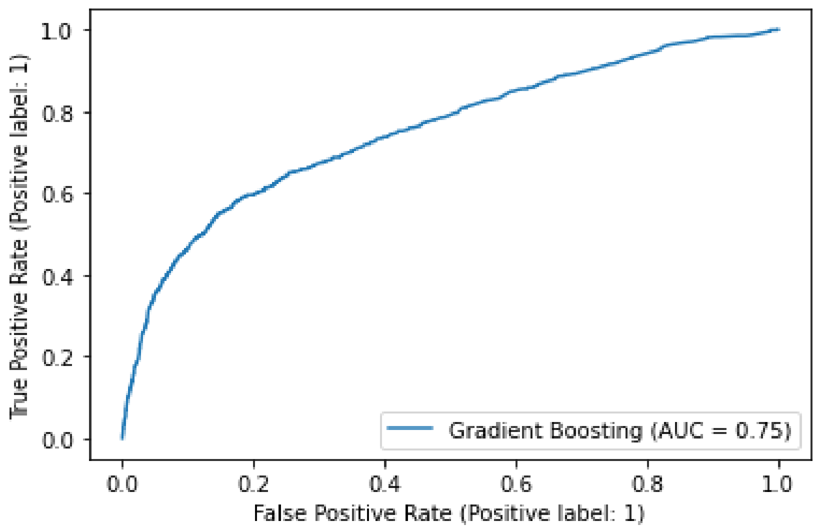


```
Cross-Validation Accuracy Scores [0.818125 0.815    0.82     0.815625 0.814375 0.8225   0.815    0.818125
 0.824375 0.809375]
```

After cross validation, I am getting almost same accuracy as we are getting before indicating that our model is neither under fit nor over fit.

ROC Curve:

The **receiver operating characteristic (ROC) curve** is frequently used for evaluating the performance of binary classification algorithms. The ROC curve is produced by calculating and plotting the **true positive rate** against the **false positive rate.**



The normal AUC is 0.64 and the plot AUC value is around 0.75.

Advantages:

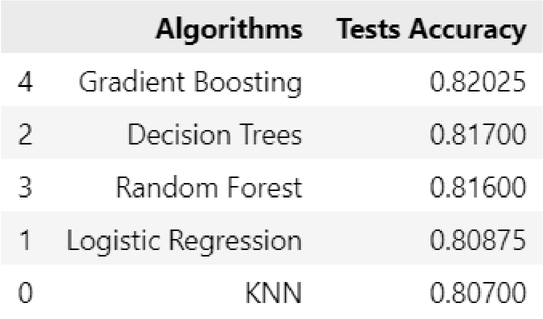
- Provides the accuracy that cannot be get easily.
- Provides us with several hyperparameter tuning options and we can also optimize different loss functions making it more flexible.
- It often works great with categorical and numerical values without any pre-processing.
- Does not require any imputation for missing data.

Disadvantages:

The models continue to improve to minimize the errors, may lead to overfitting.

It requires lots of trees which can be time and memory exhaustive.

Conclusion: The final test accuracy of all the models are as follows:



	Algorithms	Tests Accuracy
4	Gradient Boosting	0.82025
2	Decision Trees	0.81700
3	Random Forest	0.81600
1	Logistic Regression	0.80875
0	KNN	0.80700

All the models have similar accuracy, Gradient Boosting have the highest accuracy.

References:

- An Introduction to Statistical Learning with Applications in R
- Machine Learning, Tom Mitchell, McGraw Hill, 1997
- Lecture Notes and Slides
- Scikit Learn
- KD Nuggets
- Towards Data Science

THANK YOU