# BANK LOAN DEFAULT CASE PREDICTION

***Prepared by- Shrishti Sharma (14-07-2020)***

# Content

1. **Chapter 1 (Introduction)**
   1. Problem Statement.................................................................................. 3
   2. Data......................................................................................................... 3
2. **Chapter 2 (Methodology)** 
   1. Pre Processing............................................................................................4
   2. Missing Values...........................................................................................5
   3. Plots..........................................................................................................6
   4. Outlier Analysis.........................................................................................8
   5. Heatmap....................................................................................................9
   6. Scaling.........................................................................................................9
3. **Chapter 3 (Modeling)**
   1. Logistic Regression..................................................................................10
   2. Naive Bayes.............................................................................................11
   3. Decision Tree...............................................................................................12
   4. Random Forest...........................................................................................13
   5. KNN............................................................................................................14
   6. Evaluation metric.......................................................................................15
4. **Chapter 4 (Final Model ) .........................................................................16**

**CHAPTER-1**

# INTRODUCTION

* 1. **Problem Statement**

The loan default dataset has 8 variables and 850 records, each record being loan default status for each customer. Each Applicant was rated as “Defaulted” or “Not-Defaulted”. New applicants for loan application can also be evaluated on these 8 predictor variables and classified as a default or non-default based on predictor variables.

* 1. **Data**

Given below is the sample of dataset we would be using to predict the loan defaulter.

Table 1.1

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| age | ed | employ | address | income | debtinc | creddebt | othdebt | default |
| 41 | 3 | 17 | 12 | 176 | 9.3 | 11.35939 | 5.008608 | 1 |
| 27 | 1 | 10 | 6 | 31 | 17.3 | 1.362202 | 4.000798 | 0 |
| 40 | 1 | 15 | 14 | 55 | 5.5 | 0.856075 | 2.168925 | 0 |
| 41 | 1 | 15 | 14 | 120 | 2.9 | 2.65872 | 0.82128 | 0 |
| 24 | 2 | 2 | 0 | 28 | 17.3 | 1.787436 | 3.056564 | 1 |
| 41 | 2 | 5 | 5 | 25 | 10.2 | 0.3927 | 2.1573 | 0 |
| 39 | 1 | 20 | 9 | 67 | 30.6 | 3.833874 | 16.66813 | 0 |

Given below is the description of variables using which we have to classify the loan defaulters and non-defaulters.

Table 1.2

Variable Description

age Age of each customer

education Education Categories

employment Employment status- Corresponds to job status and being converted

to numeric format.

Address Geographic area-converted to numeric values

Income Gross income of each customer

debtinc Individuals debt payment to his or her gross income

creddebt debt-to-credit ratio

othdebt any other debts

default 1- defaulter 0 – non-defaulter

Hence, “default” is our target variable which we need to predict using rest of the variables.

# CHAPTER-2

# METHODOLOGY

**2.1 Pre-Processing**

Data pre-processing in Machine Learning is a crucial step that helps enhance the quality of data to promote the extraction of meaningful insights from the data. Data pre-processing in Machine Learning refers to the technique of preparing (cleaning and organizing) the raw data to make it suitable for a building and training Machine Learning models. In simple words, data pre-processing in Machine Learning is a data mining technique that transforms raw data into an understandable and readable format.

To start with the pre processing of data, first separate out the numerical and categorical data to further analyse each of them separately.

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 850 entries, 0 to 849

Data columns (total 9 columns):

age 850 non-null int64

ed 850 non-null category

employ 850 non-null int64

address 850 non-null int64

income 850 non-null int64

debtinc 850 non-null float64

creddebt 850 non-null float64

othdebt 850 non-null float64

default 700 non-null float64

dtypes: category(1), float64(4), int64(4)

memory usage: 54.3 KB

Hence there is only one categorical variable “ed” i.e. education categories and rest are numerical variables.

Summary of data

age ed employ address income

Min.:20.00 Min.:1.000 Min.: 0.000 Min.: 0.000 Min: 13.00 1st Qu.:29.00 1st Qu.:1.000 1st Qu.: 3.000 1st Qu.: 3.000 1st Qu.: 24.00

Median :34.00 Median :1.000 Median : 7.000 Median : 7.000 Median : 35.00

Mean :35.03 Mean :1.711 Mean : 8.566 Mean : 8.372 Mean : 46.68

3rd Qu.:41.00 3rd Qu.:2.000 3rd Qu.:13.000 3rd Qu.:12.000 3rd Qu.: 55.75

Max. :56.00 Max. :5.000 Max. :33.000 Max. :34.000 Max. :446.00

debtinc creddebt othdebt default

Min. : 0.10 Min. : 0.0117 Min. : 0.04558 Min. :0.0000

1st Qu.: 5.10 1st Qu.: 0.3822 1st Qu.: 1.04594 1st Qu.:0.0000

Median : 8.70 Median : 0.8851 Median : 2.00324 Median :0.0000

Mean :10.17 Mean : 1.5768 Mean : 3.07879 Mean :0.2614

3rd Qu.:13.80 3rd Qu.: 1.8984 3rd Qu.: 3.90300 3rd Qu.:1.0000

Max. :41.30 Max. :20.5613 Max. :35.19750 Max. :1.0000

NA's :150

**2.1.1 Handling Missing Values**

In data pre-processing, it is pivotal to identify and correctly handle the missing values, failing to do this, we might draw inaccurate and faulty conclusions and inferences from the data. And , this will hamper our project. In this bank loan default project there are missing values for target variable. Hence, we will predict those missing values by our model formation.

age 0

ed 0

employ 0

address 0

income 0

debtinc 0

creddebt 0

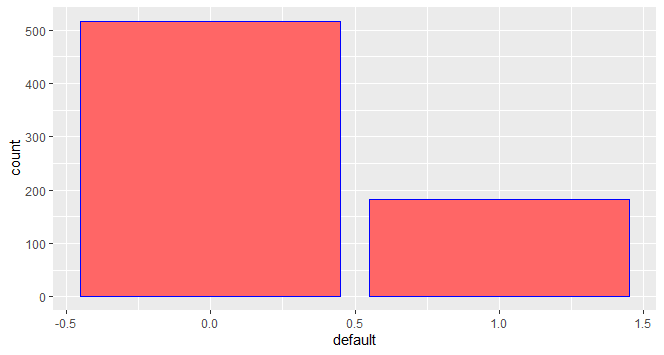
othdebt 0

default 150

dtype: int64

**2.1.2 Class Distribution**

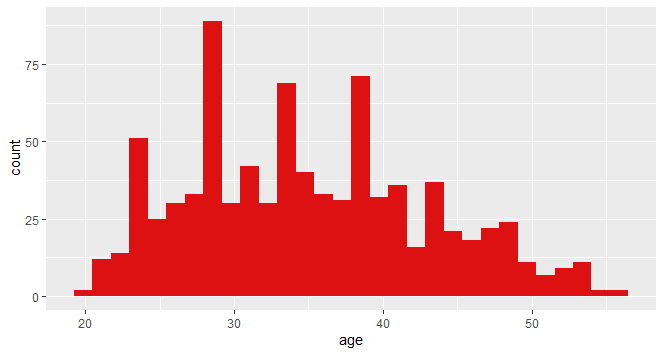
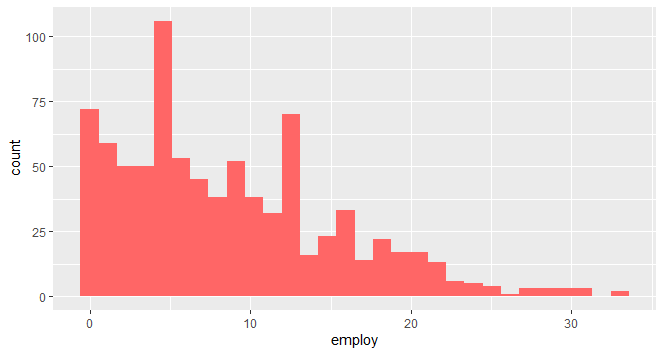
We check target class distribution in order to check the imbalance in the dataset. Imbalanced classifications pose a challenge for predictive modelling as most of the machine learning algorithms used for classification were designed around the assumption of an equal number of examples for each class. This results in models that have poor predictive performance, specifically for the minority class. This is a problem because typically, the minority class is more important and therefore the problem is more sensitive to classification errors for the minority class than the majority class.

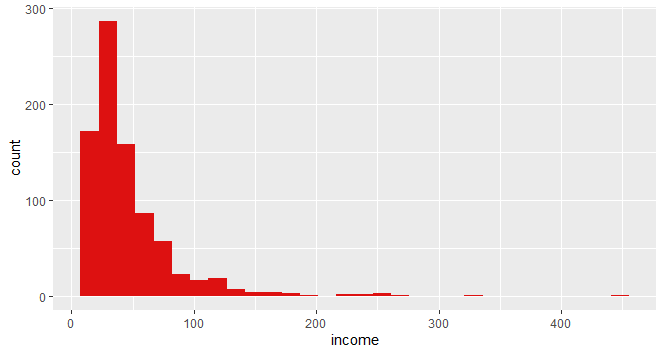
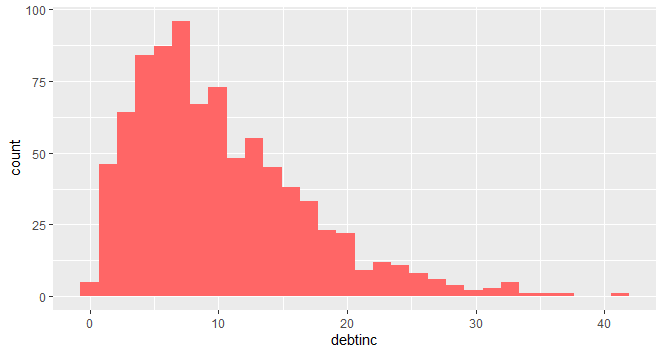


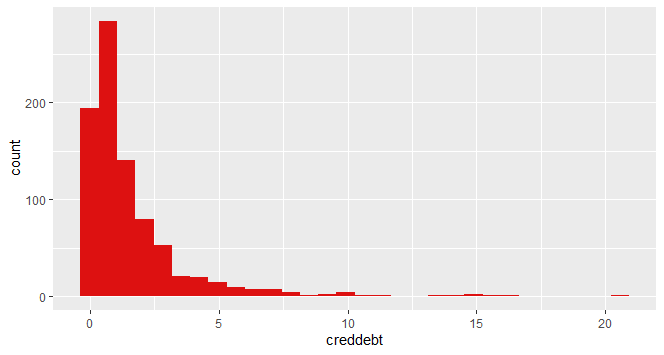
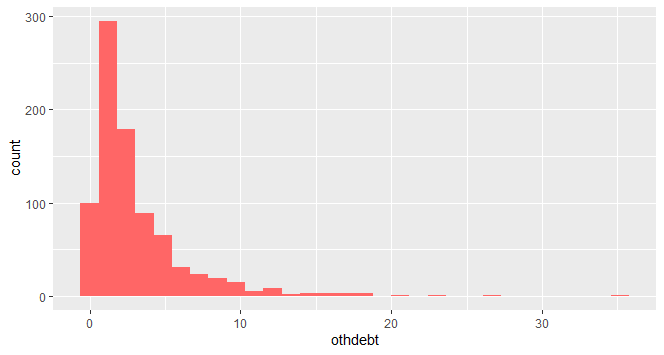
Prob of 0 = 0.7385714285714285

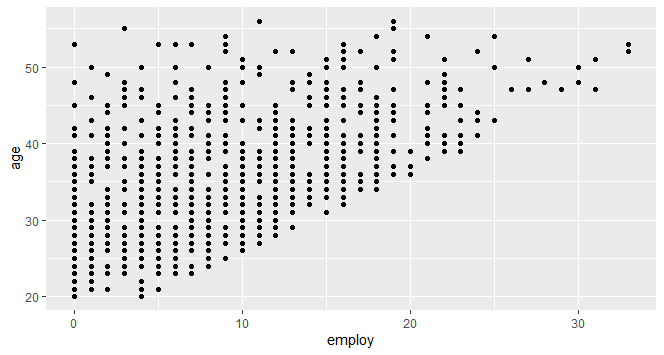
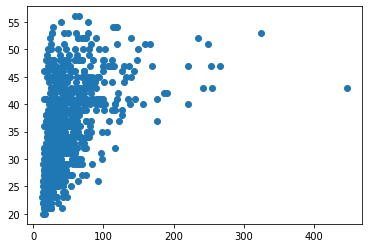
Prob of 1 = 0.26142857142857145

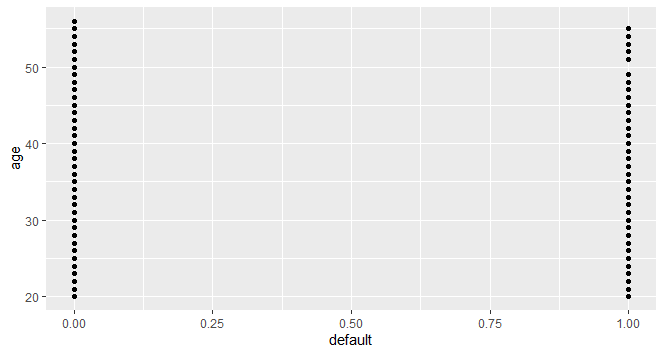
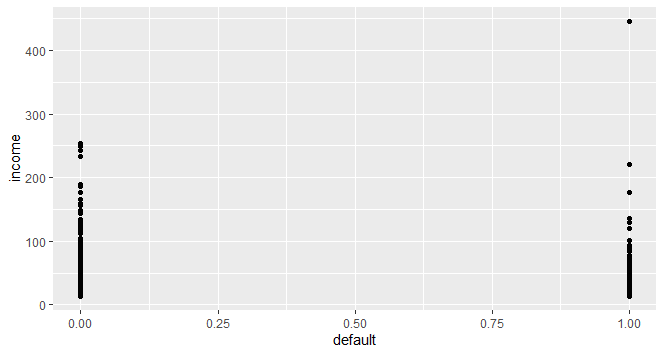
**2.1.3 Visualization using plots**

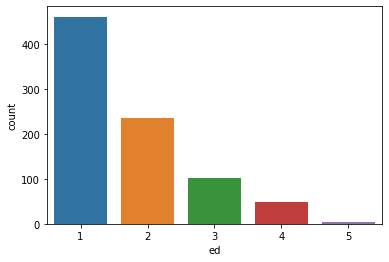
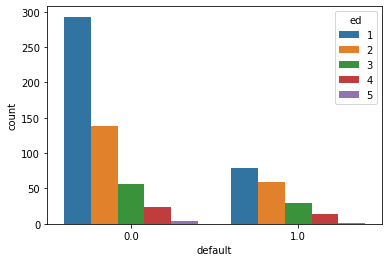
 

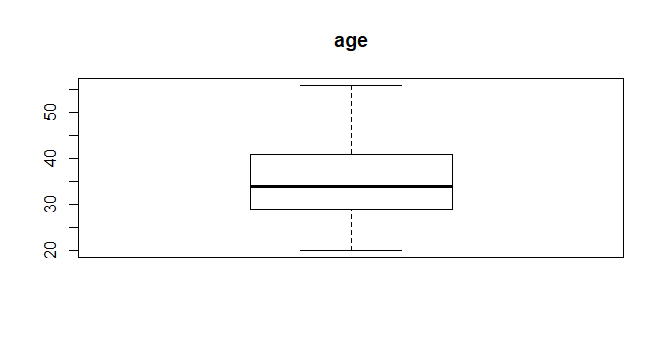
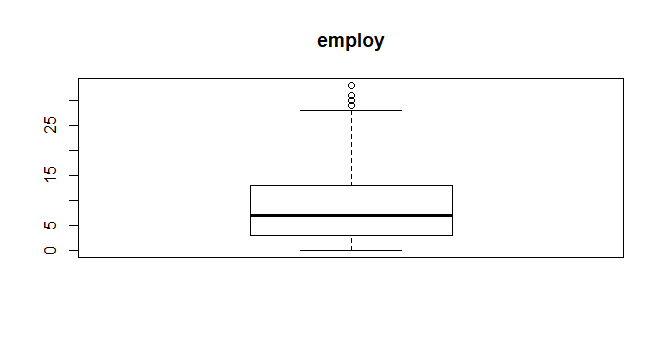
 

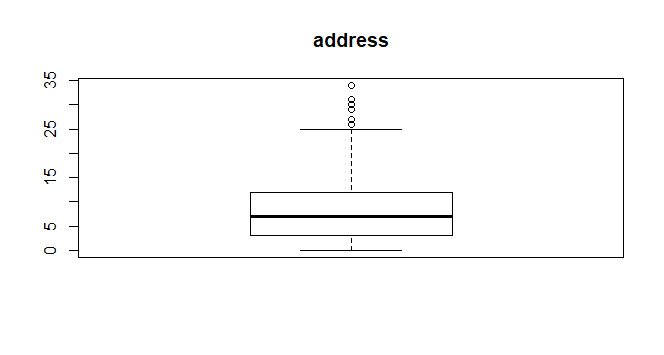
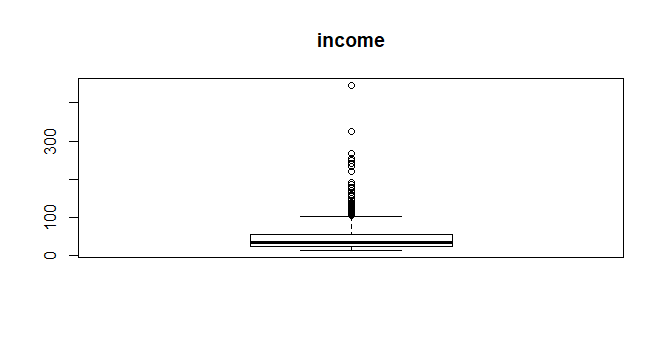
 

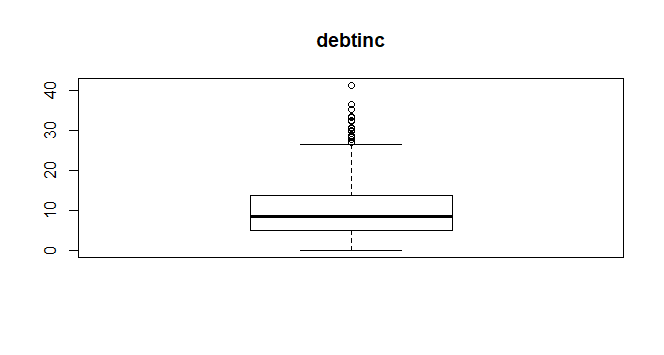
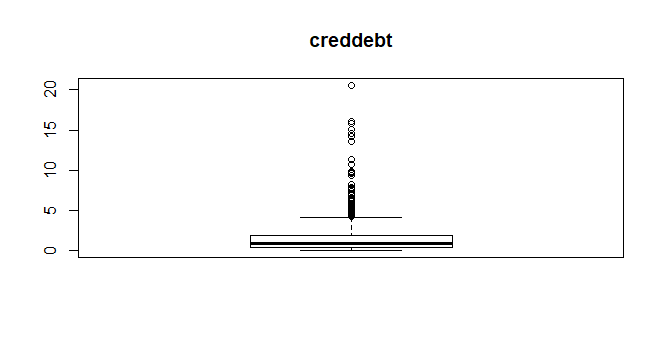
 

**2.1.4 Box plot for Outlier Analysis**

** **

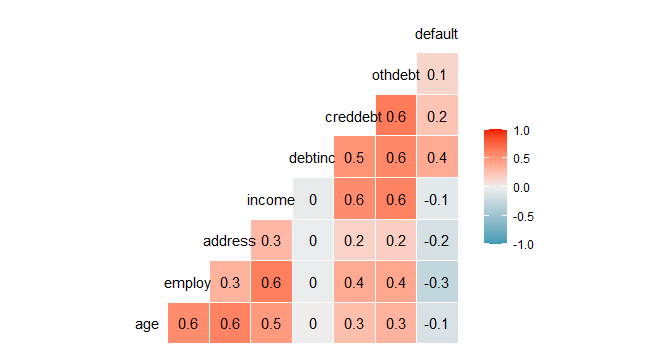
** **

**` **

**2.2 Feature Selection**

**2.2.1 Correlation Matrix with heat map**

The first step we took for the feature Selection was to examine a heatmap in order to understand the correlation among the variables. Correlation states how the features are related to each other or the target variable.Correlation can be positive (increase in one value of feature increases the value of the target variable) or negative (increase in one value of feature decreases the value of the target variable)Heatmap makes it easy to identify which features are most related to the target variable.The heat map indicating the correlation value is given below.

****

From the above heatmap we see that there no such highly co related variables. Hence we do not drop any variables and require every variable for our model formation.

**2.3 Scaling**

Feature Scaling is a technique to standardize the independent features present in the data in a fixed range. It is performed during the data pre-processing to handle highly varying magnitudes or values or units. If feature scaling is not done, then a machine learning algorithm tends to weigh greater values, higher and consider smaller values as the lower values, regardless of the unit of the values.

**Chapter -3**

**MODEL FORMATION**

**3.1.1 Logistic Regression**

Logistic regression is a supervised learning classification algorithm used to predict the probability of a target variable. The nature of target or dependent variable is dichotomous, which means there would be only two possible classes.

In simple words, the dependent variable is binary in nature having data coded as either 1 (stands for success/yes) or 0 (stands for failure/no).

Mathematically, a logistic regression model predicts P(Y=1) as a function of X. It is one of the simplest ML algorithms that can be used for various classification problems.

***Model***

LogisticRegression(C=1.0, class\_weight=None, dual=False, fit\_intercept=True,

intercept\_scaling=1, l1\_ratio=None, max\_iter=100,

multi\_class='warn', n\_jobs=None, penalty='l2',

random\_state=0, solver='warn', tol=0.0001, verbose=0,

warm\_start=False)

***Confusion Matrix***

array([[90, 10],

[20, 20]], dtype=int64)

***Classification Report***

precision recall f1-score support

0.0 0.82 0.90 0.86 100

1.0 0.67 0.50 0.57 40

accuracy 0.79 140

macro avg 0.74 0.70 0.71 140

weighted avg 0.77 0.79 0.78 140

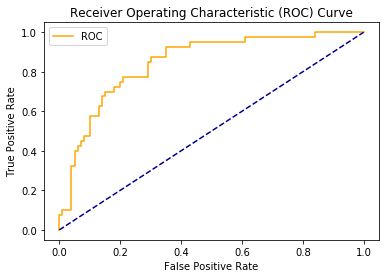
***Accuracy***

0.7857142857142857

***Area Under Curve (AUC)***

AUC= 0.846

***ROC Curve***



**3.1.2 Naive Bayes**

Naïve Bayes algorithms is a classification technique based on applying Bayes’ theorem with a strong assumption that all the predictors are independent to each other. In simple words, the assumption is that the presence of a feature in a class is independent to the presence of any other feature in the same class.

***Model***

GaussianNB(priors=None, var\_smoothing=1e-09)

***Confusion Matrix***

[[88 12]

[23 17]]

***Classification Report***

precision recall f1-score support

0.0 0.79 0.88 0.83 100

1.0 0.59 0.42 0.49 40

accuracy 0.75 140

macro avg 0.69 0.65 0.66 140

weighted avg 0.73 0.75 0.74 140

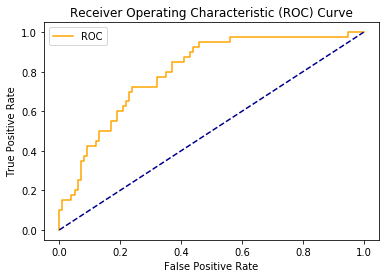
***Accuracy***

0.75

***Area Under Curve (AUC)***

AUC= 0.8005

***ROC Curve***



**3.1.3 Decision Tree**

 Decision tree analysis is a predictive modelling tool that can be applied across many areas. Decision trees can be constructed by an algorithmic approach that can split the dataset in different ways based on different conditions. Decisions trees are the most powerful algorithms that falls under the category of supervised algorithms.

They can be used for both classification and regression tasks. The two main entities of a tree are decision nodes, where the data is split and leaves, where we got outcome.

***Model***

DecisionTreeClassifier(class\_weight=None, criterion='entropy', max\_depth=None,

max\_features=None, max\_leaf\_nodes=None,

min\_impurity\_decrease=0.0, min\_impurity\_split=None,

min\_samples\_leaf=1, min\_samples\_split=2,

min\_weight\_fraction\_leaf=0.0, presort=False,

random\_state=0, splitter='best')

***Confusion Matrix***

[[85 15]

[22 18]]

***Classification Report***

precision recall f1-score support

0.0 0.79 0.85 0.82 100

1.0 0.55 0.45 0.49 40

accuracy 0.74 140

macro avg 0.67 0.65 0.66 140

weighted avg 0.72 0.74 0.73 140

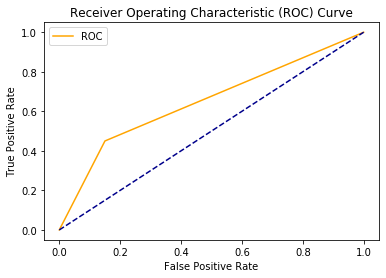
***Accuracy***

0.7357142857142858

***Area Under Curve (AUC)***

AUC= 0.649999999999999

***ROC Curve***

******

**3.1.4 Random Forest**

Random forest is a supervised learning algorithm which is used for both classification as well as regression. But however, it is mainly used for classification problems. As we know that a forest is made up of trees and more trees means more robust forest. Similarly, random forest algorithm creates decision trees on data samples and then gets the prediction from each of them and finally selects the best solution by means of voting. It is an ensemble method which is better than a single decision tree because it reduces the over-fitting by averaging the result.

***Model***

RandomForestClassifier(bootstrap=True, class\_weight=None, criterion='entropy',

max\_depth=None, max\_features='auto', max\_leaf\_nodes=None,

min\_impurity\_decrease=0.0, min\_impurity\_split=None,

min\_samples\_leaf=1, min\_samples\_split=2,

min\_weight\_fraction\_leaf=0.0, n\_estimators=20,

n\_jobs=None, oob\_score=False, random\_state=0, verbose=0,

warm\_start=False)

***Confusion Matrix***

[[92 8]

[25 15]]

***Classification Report***

precision recall f1-score support

0.0 0.79 0.92 0.85 100

1.0 0.65 0.38 0.48 40

accuracy 0.76 140

macro avg 0.72 0.65 0.66 140

weighted avg 0.75 0.76 0.74 140

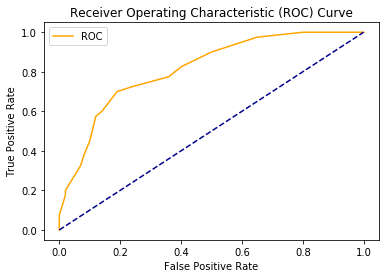
***Accuracy***

0.7642857142857142

***Area Under Curve (AUC)***

AUC= 0.8145000000000001

***ROC Curve***

****

**3.1.5 KNN**

K-nearest neighbors (KNN) algorithm is a type of supervised ML algorithm which can be used for both classification as well as regression predictive problems. However, it is mainly used for classification predictive problems in industry. KNN is a lazy learning algorithm because it does not have a specialized training phase and uses all the data for training while classification. KNN is also a non-parametric learning algorithm because it doesn’t assume anything about the underlying data.

***Model***

KNeighborsClassifier(algorithm='auto', leaf\_size=30, metric='minkowski',

metric\_params=None, n\_jobs=None, n\_neighbors=5, p=2,

weights='uniform')

***Confusion Matrix***

[[92 4]

[27 17]]

***Accuracy***

0.7785714285714286

***Classification Report***

precision recall f1-score support

0.0 0.77 0.96 0.86 96

1.0 0.81 0.39 0.52 44

accuracy 0.78 140

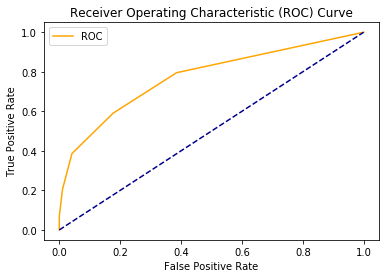
macro avg 0.79 0.67 0.69 140

weighted avg 0.78 0.78 0.75 140

***AUC***

AUC= 0.7729640151515151

***ROC Curve***

******

**3.2 Evaluation**

**3.2.1 Confusion Matrix**

It is the easiest way to measure the performance of a classification problem where the output can be of two or more type of classes. A confusion matrix is nothing but a table with two dimensions viz. “Actual” and “Predicted” and furthermore, both the dimensions have “True Positives (TP)”, “True Negatives (TN)”, “False Positives (FP)”, “False Negatives (FN)”.

* True Positives (TP) − It is the case when both actual class & predicted class of data point is 1.
* True Negatives (TN) − It is the case when both actual class & predicted class of data point is 0.
* False Positives (FP) − It is the case when actual class of data point is 0 & predicted class of data point is 1.
* False Negatives (FN) − It is the case when actual class of data point is 1 & predicted class of data point is 0.

**3.2.2 Classification Accuracy**

It is most common performance metric for classification algorithms. It may be defined as the number of correct predictions made as a ratio of all predictions made. We can easily calculate it by confusion matrix with the help of following formula −

Accuracy=TP+TN/(TP+FP+FN+TN)

**3.2.3 Classification Report**

This report consists of the scores of Precisions, Recall, F1 and Support.

Precision=TP/(TP+FN)

Recall=TP/(TP+FN)

Specificity=TN/(TN+FP)

F1 Score-This score will give us the harmonic mean of precision and recall. Mathematically, F1 score is the weighted average of the precision and recall. The best value of F1 would be 1 and worst would be 0. F1 score is having equal relative contribution of precision and recall.

F1=2∗(precision∗recall)/(precision+recall) recall.

**3.2.4 AUC (Area Under ROC curve)**

AUC (Area Under Curve)-ROC (Receiver Operating Characteristic) is a performance metric, based on varying threshold values, for classification problems. As name suggests, ROC is a probability curve and AUC measure the separability. In simple words, AUC-ROC metric will tell us about the capability of model in distinguishing the classes. Higher the AUC, better the model.Mathematically, it can be created by plotting TPR (True Positive Rate) i.e. Sensitivity or recall vs FPR (False Positive Rate) i.e. 1-Specificity, at various threshold values.

**Chapter-4**

**Final Model**

**Decision Tree**

The best performing model among all the model chosen is Decision Tree.It has greater accuracy with a large value of area under curve.It also has the high F-1 score.

Predicted the missing value using Decision tree model.Then forming a decision tree model on the imputed missing values for the further prediction of the data.

***Model***

RandomForestClassifier(bootstrap=True, class\_weight=None, criterion='entropy',

max\_depth=None, max\_features='auto', max\_leaf\_nodes=None,

min\_impurity\_decrease=0.0, min\_impurity\_split=None,

min\_samples\_leaf=1, min\_samples\_split=2,

min\_weight\_fraction\_leaf=0.0, n\_estimators=200,

n\_jobs=None, oob\_score=False, random\_state=0, verbose=0,

warm\_start=False)

***Confusion Matrix***

[[124 5]

[ 21 20]]

Accuray = 0.8470588235294118

***Area under Curve***

AUC= 0.8393836263944036

Thank you