PROJECT REPORT

ON

BANK LOAN DEFAULTER PREDICTION

SUBMITTED BY:

ARPIT DUBEY

[04/07/2020]

1.1 Problem Statement

Predictive analytics is the stream of advanced analytics which utilizes diverse techniques like mining, predictive modelling, statistics, machine learning and artificial intelligence to analyze current data and predict future. Loans default will cause huge loss for the bank so they pay much attention on this issue to apply various method to detect and predict default behaviors of their customers.

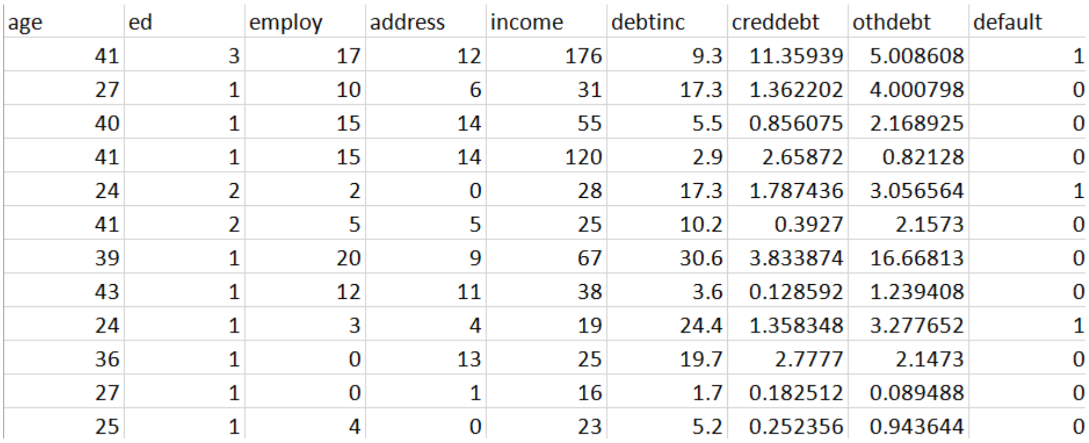
The loan default dataset has 8 variables and 850 records, each record being loan default status for each customer. Each Applicant was rated as “Defaulted” or “Not-Defaulted”. New applicants for loan application can also be evaluated on these 8 predictor variables and classified as a default or non-default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

What is classification?

In machine learning and statistics, classification is a supervised learning approach in which the computer program learns from the data input given to it and then uses this learning to classify new observation. This data set may simply be bi-class (like identifying whether the person is male or female or that the mail is spam or non-spam) or it may be multi-class too. Some examples of classification problems are: speech recognition, handwriting recognition, bio metric identification, document classification etc.

1.2 Data

Our task is to build the classification model to predict the whether the customer is “Defaulted” or “Not-Defaulted” based on predictor variables. Given below is a sample of the data set that we are using to predict the count:



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
43	1	12	11	38	3.6	0.128592	1.239408	0
24	1	3	4	19	24.4	1.358348	3.277652	1
36	1	0	13	25	19.7	2.7777	2.1473	0
27	1	0	1	16	1.7	0.182512	0.089488	0
25	1	4	0	23	5.2	0.252356	0.943644	0

Table 1.1: Bank Loan Default Case. Sample data (Columns:1- 8)

Below are the variables we used to predict the default status here 8 variables are independent variables and predictors and one variable ‘default’ is target variable.

Sr.no	Column Name
1	Age
2	Education
3	Employment
4	Address
5	Income
6	Debitinc
7	creddebt
8	othdebt

Table 1.2: Customer default status Prediction variables

1.3 Software and Hardware Requirement:

A) R 3.6.1 for 64 bit
B) Anaconda 3 for 64 bit
C) R studio
D) 64 bit OS
E) Python 3
F) Jupyter Notebook
G) 4GB of RAM

Chapter 2

Methodology

2.1 Pre-Processing:

Any predictive modeling requires that we look at the data before we start modeling. However, in data mining terms looking at data refers to so much more than just looking. Looking at data refers to exploring the data, cleaning the data as well as visualizing the data through graphs and plots. This is often called as Exploratory Data Analysis. To start this process we will first try and look at class imbalance of Target variable in most of the classification class imbalance will create severe problems during the modelling.

2.1.1 Univariate Analysis

Target Value 'defaulted' contains 85% of data contains 'not-defaulted' Customers and 14.5 % of data contains defaulted' , it may be chance that class imbalance problem may occurs because of less proportion of data contains defaulted customers , we should be very careful on during evaluation of Model instead of concentration on only Accuracy we should also concentrate on Precision and Recall also and we should make sure that Precision and Recall should also be high.

2.1.2 Distribution of Dependent Numeric Variables:

Except few like "Income", there is chance of outlier presents in this variables, other variables are looking almost Normally Distributed.

2.2 Exploratory Data Analysis:

Exploratory Data Analysis refers to the critical process of performing initial investigations on data so as to discover patterns, to spot anomalies, to test hypothesis and to check assumptions with the help of summary statistics and graphical representations. It is a good practice to understand the data first and try to gather as many insights from it. EDA is all about making sense of data in hand, before getting them dirty with it.

Chapter 3

Missing Value Analysis:

Missing values in data is a common phenomenon in real world problems. Knowing how to handle missing values effectively is a required step to reduce bias and to produce powerful models. Missing value analysis is done to check is there any missing value present in given dataset. Missing values can be easily treated using various methods like mean, median method, KNN method to impute missing value.

Below table illustrate the missing value present in the data.

Column Name	Missing Values
Age	0
Education	0
Employment	0
Address	0
Income	0
Debitinc	0
creddebt	0
othdebt	0
default	150

Table 2.1 Missing Values in bank loan data

In R function(x){sum(is.na(x))} is the function used to check the sum of missing values. In python df.isnull().sum() is used to detect any missing value.

There are 150 missing values here and the percentage of missing values is 17.64%.

So, we remove missing values using df=df.dropna() in python.

Only 700 observations were left.

Chapter 4

Outlier Analysis:

Outlier analysis is done to handle all inconsistent observations present in given dataset. As outlier analysis can only be done on continuous variable.

Figure 4.1 and 4.2 are visualization of numeric variable present in our dataset to detect outliers using boxplot and distribution plots. Outliers will be detected with black color. According to above visualizations there all features columns shows outliers. All independent variables are right skewed/positively skewed.

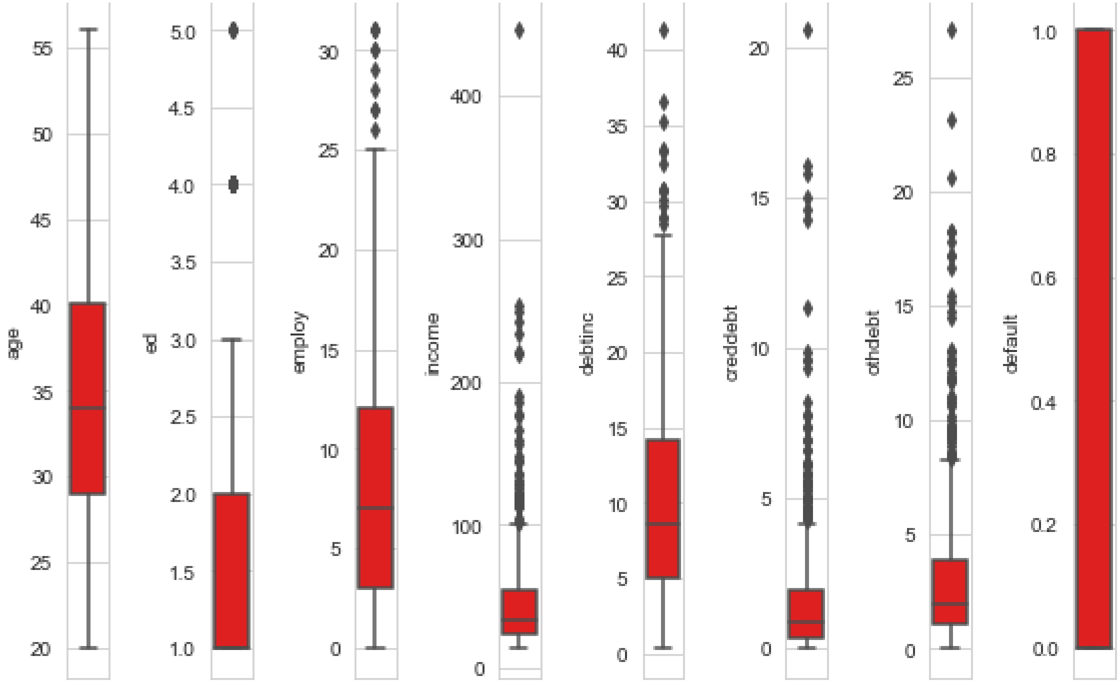


Figure 4.1 Boxplot graph of variables

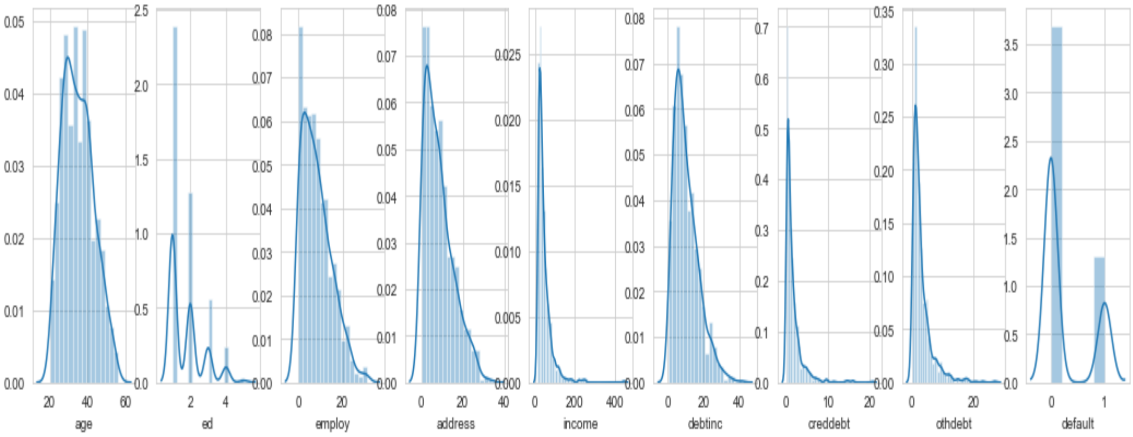


Figure 2.2 Distribution plots -To check distribution-Skewness

Chapter 5

5.1 Features Selections

Machine learning works on a simple rule – if you put garbage in, you will only get garbage to come out. By garbage here, I mean noise in data.

This becomes even more important when the number of features are very large. You need not use every feature at your disposal for creating an algorithm. You can assist your algorithm by feeding in only those features that are really important. I have myself witnessed feature subsets giving better results than complete set of features for the same algorithm or – "Sometimes, less is better!".

We should consider the selection of feature for model based on below criteria

1) The relationship between two independent variables should be less.
2) The relationship between Independent and Target variables should be high.

Feature Selection is the process of selecting the attributes that can make the predicted variable more accurate or eliminating those attributes that are irrelevant and can decrease the model accuracy and quality.

Data and feature correlation is considered one important step in the feature selection phase of the data pre-processing especially if the data type for the features is continuous.

Positive Correlation: Means that if feature A increases then feature B also increases or if feature A decreases then feature B also decreases. Both features move in tandem and they have a linear relationship.

Negative Correlation: Means that if feature A increases then feature B decreases and vice versa.

No Correlation: No relationship between those two attributes.

Each of those correlation types can exist in a spectrum represented by values from 0 to 1 where slightly or highly positive correlation features can be something like 0.5 or 0.7. If there is a strong and perfect positive correlation, then the result is represented by a correlation score value of 0.9 or 1. If there is a strong negative correlation, it will be represented by a value of -1.

Below fig 5.1 illustrates that relationship between all numeric variables using heattmap plot.

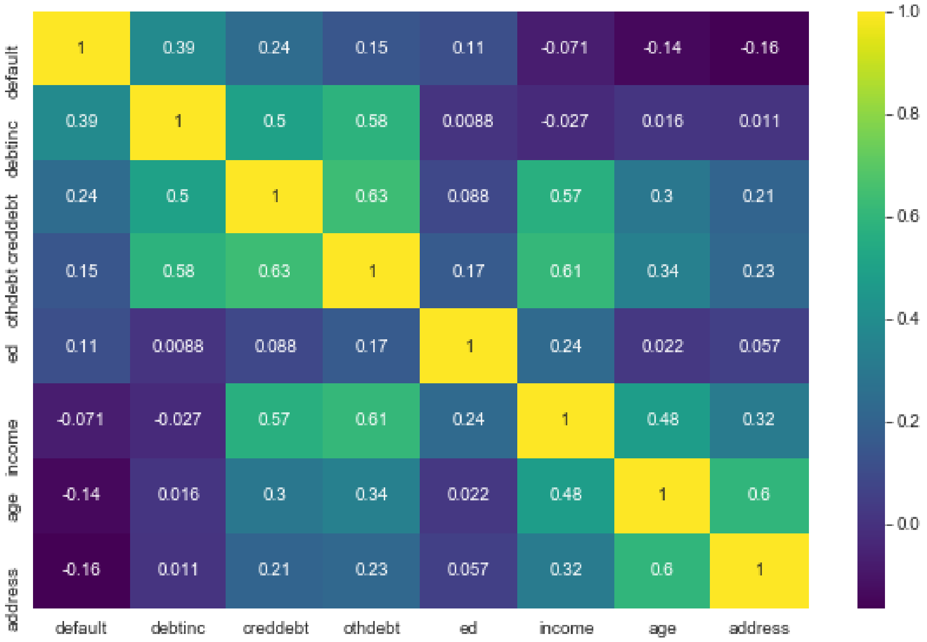


Figure 2.5 correlation plot of numeric variables

5.2 Dimensional Reduction using Chi –Square Test for Categorical Variable:

In this we have only one categorical variable so we are not going to use any test to detect correlation of categorical variable.

Chapter 6

Feature Scaling

Feature scaling includes two functions normalization and standardization. It is done reduce unwanted variation either within or between variables and to bring all of the variables into proportion with one another.

In given dataset all numeric values are already present in normalized form.

Features Scaling Using Standardization: Most of the Machine Learning algorithms performance depends on data we are passing through it.

If two variables are in different ranges than there is chance that Model will bias towards that higher range variable so it is important to Scale Numeric variables in same range.

As we observed in histogram that there is almost all the variable are varying from different range so first, we apply the normalization, then we are using Standardization (Z - Score) technique to scale the Numeric Variable.

$$Value_{new} = \frac{Value - minValue}{maxValue - minValue}$$

The above one is the formula to normalize all the data.

$$z = \frac{x - \mu}{\sigma}$$

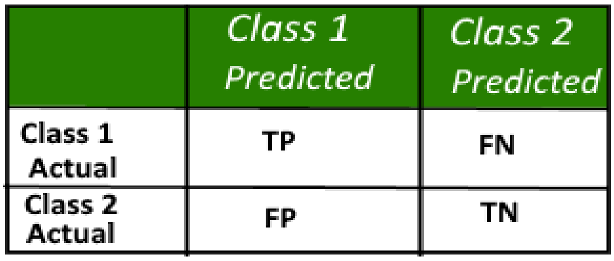
The above one is the formula to Standardization all the data.

Chapter 7

Modelling

7.1 Confusion Matrix

A confusion matrix is a table that is often used to **describe the performance of a classification model** (or "classifier") on a set of test data for which the true values are



	Class 1 Predicted	Class 2 Predicted
Class 1 Actual	TP	FN
Class 2 Actual	FP	TN

Figure 7.1 Confusion Matrix

Known here,

Class 1: Positive

Class 2: Negative

Definition of the Terms:

- Positive (P): Observation is positive (for example: is an apple).
- Negative (N): Observation is not positive (for example: is not an apple).
- True Positive (TP): Observation is positive, and is predicted to be positive.
- False Negative (FN): Observation is positive, but is predicted negative.
- True Negative (TN): Observation is negative, and is predicted to be negative.
- False Positive (FP): Observation is negative, but is predicted positive.

Classification Rate/Accuracy:

Precision: Precision is fraction of items the classifier flags as being in the class actually are in the class.

Precision = TP/TP+FP

Recall: What fraction of things that are in the class are detected by the classifier.

Recall = TP/TP + FN

Accuracy: Below is the actual over all Accuracy of the Model

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

F1 Score: It is the combination of the Precision and recall

F1 Score = 2*(Precision*Recall)/(Precision+Recall)

However, there are problems with accuracy. It assumes equal costs for both kinds of errors. A 99% accuracy can be excellent, good, mediocre, poor or terrible depending upon the problem.

ROC CURVES and AUC

Receiver operating characteristic (ROC) curve: plots the true positive rate (TPR) versus the false positive rate (FPR) as a function of the model's threshold for classifying a positive.

Area under the curve (AUC): metric to calculate the overall performance of a classification model based on area under the ROC curve.

7.2 Model Selection:

In this case we have to Each Applicant was rated as "Defaulted" or "Not-Defaulted". New applicants for loan application can also be evaluated on the 8 predictor variables and classified as a default or non-default based on predictor variables. Model having less error rate and more accuracy will be our final model. In these we have divided the dataset into train and test part using random sampling. For this model we have divided the dataset into train and test part using random sampling Where train contains 80% data of data set and test contains 20% data and contains 8 variables where 8th variable is the target variable.

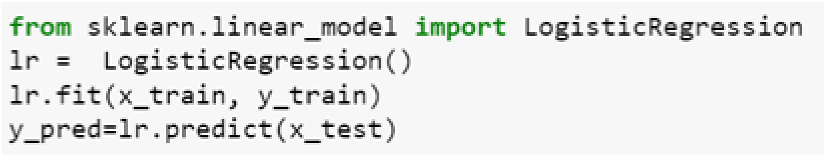
Classification Models used are:

7.2.1 Logistic Regression:

Logistic regression is a machine learning algorithm for classification. In this algorithm, the probabilities describing the possible outcomes of a single trial are modelled using a logistic function.

Advantages: Logistic regression is designed for this purpose (classification), and is most useful for understanding the influence of several independent variables on a single outcome variable.

Disadvantages: Works only when the predicted variable is binary, assumes all predictors are independent of each other, and assumes data is free of missing values.



```
from sklearn.linear_model import LogisticRegression
lr =  LogisticRegression()
lr.fit(x_train, y_train)
y_pred=lr.predict(x_test)
```

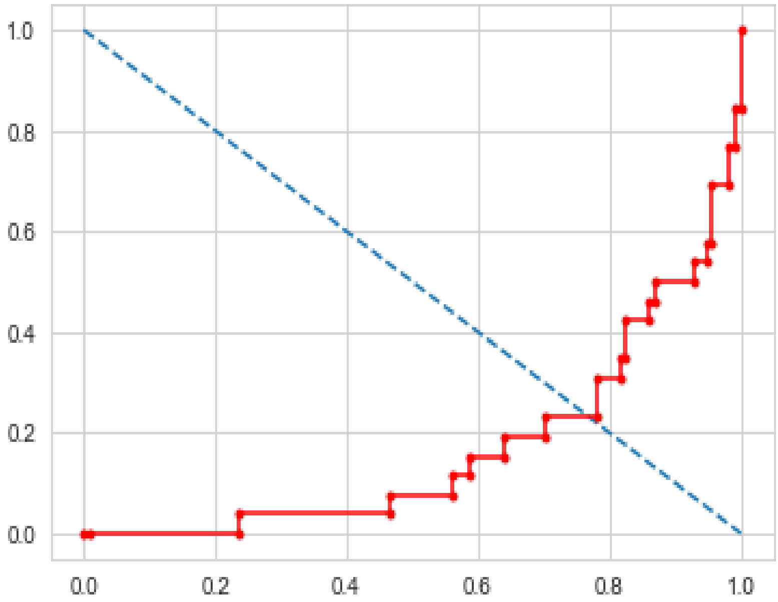
Accuracy: 84.28571428571429

FNR: 50.0

Defaulted: 22

Non-defaulted: 118

ROC -Curve:



7.2.2 Decision Trees:

Given a data of attributes together with its classes, a decision tree produces a sequence of rules that can be used to classify the data.

Advantages: Decision Tree is simple to understand and visualize, requires little data preparation, and can handle both numerical and categorical data.

Disadvantages: Decision tree can create complex trees that do not generalize well, and decision trees can be unstable because small variations in the data might result in a completely different tree being generated.

Accuracy: 74.28571428571429

FNR: 65.38461538461539

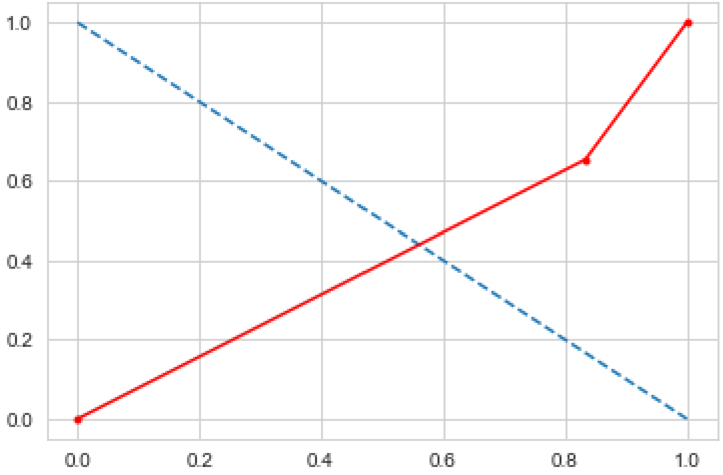
Defaulted: 28

Non-defaulted: 112

Confusion Matrix:

95 19
17 9

ROC Curve:



7.2.3 Random Forest Classifier:

Random forest classifier is a meta-estimator that fits a number of decision trees on various sub-samples of datasets and uses average to improve the predictive accuracy of the model and controls over-fitting. The sub-sample size is always the same as the original input sample size but the samples are drawn with replacement.

Advantages: Reduction in over-fitting and random forest classifier is more accurate than decision trees in most cases.

Disadvantages: Slow real time prediction, difficult to implement, and complex algorithm.

Accuracy: 83.57142857142857

FNR: 65.38461538461539

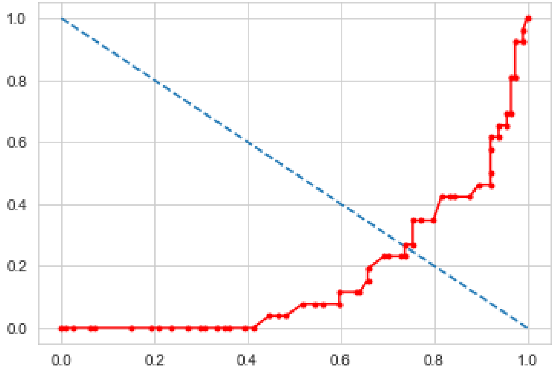
Defaulted: 15

Non-defaulted: 125

Confusion Matrix:

108	6
17	9

ROC Curve:



7.2.4 Naive Bayes

Naive Bayes algorithm based on Bayes' theorem with the assumption of independence between every pair of features. Naive Bayes classifiers work well in many real-world situations such as document classification and spam filtering.

It is a classification technique based on Bayes' Theorem with an assumption of independence among predictors. 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.

For example, a fruit may be considered to be an apple if it is red, round, and about 3 inches in diameter. Even if these features depend on each other or upon the existence of the other features, all of these properties independently contribute to the probability that this fruit is an apple and that is why it is known as 'Naive'.

Bayes theorem provides a way of calculating posterior probability P(c|x) from P(c), P(x) and P(x|c). Look at the equation below:

$$\underset{\text{Posterior Probability}}{P(c\,|\,x)} = \frac{\overset{\text{Likelihood}}{P(x\,|\,c)}\,\overset{\text{Class Prior Probability}}{P(c)}}{\underset{\text{Predictor Prior Probability}}{P(x)}}$$

$$P(c\,|\,X) = P(x_1\,|\,c)\times P(x_2\,|\,c)\times\cdots\times P(x_n\,|\,c)\times P(c)$$

Above,

- *P*(*c*|*x*) is the posterior probability of *class* (c, *target*) given *predictor* (x, *attributes*).
- *P*(*c*) is the prior probability of *class*.
- *P*(*x*|*c*) is the likelihood which is the probability of *predictor* given *class*.
- *P*(*x*) is the prior probability of *predictor*.

Advantages: This algorithm requires a small amount of training data to estimate the necessary parameters. Naive Bayes classifiers are extremely fast compared to more sophisticated methods.

Disadvantages: Naive Bayes is is known to be a bad estimator.

Accuracy: 81.42857142857143

FNR: 76.92307692307692

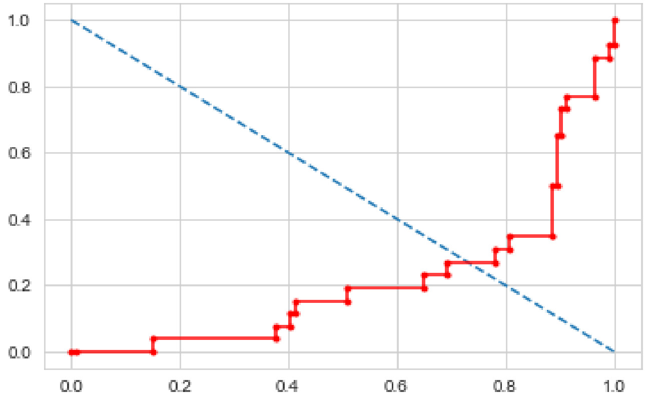
Defaulted: 12

Non-defaulted: 128

Confusion Matrix:

108 6
20 6

ROC Curve:



Conclusion: By using all the above modal here, the results we get this table of results

Classification Algorithms (Models)	Accuracy
Logistic regression	84.20
Decision Tree	74.23
Random Forest	83.57
Naive Bayes	81.42
XGB	81.42