BanK Loan Default Case

by Shubh Gupta

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.

Dataset :-

The dataset is having 850 observations and 8 variables.

	age	ed	employ	address	income	debtinc	creddebt	othdebt	default
0	41	3	17	12	176	9.3	11.359392	5.008608	1.0
1	27	1	10	6	31	17.3	1.362202	4.000798	0.0
2	40	1	15	14	55	5.5	0.856075	2.168925	0.0
3	41	1	15	14	120	2.9	2.658720	0.821280	0.0
4	24	2	2	0	28	17.3	1.787436	3.056564	1.0

Missing Value Analysis :-

	count
age	0
ed	0
employ	0
address	0
income	0
debtinc	0
creddebt	0
othdebt	0
default	150

Observation :-

In all the 9 variables, we have 150 missing values in the default predictor which is our target or dependent variable. Since imputing data in a target variable could result into a negative impact on our model, so we will drop observations associated with these 150 missing values.

Missing Value Analysis on the New Dataset :-

	count
age	0
ed	0
employ	0
address	0
income	0
debtinc	0
creddebt	0
othdebt	0
default	0

Observation :-

Now we have a new dataset with 700 observations and 9 variables with no missing values.

Splitting the Dataset into Train and Test :-

Train Data

	age	ed	employ	address	income	debtinc	creddebt	othdebt	default
0	41	3	17	12	176	9.3	11.359392	5.008608	1.0
1	27	1	10	6	31	17.3	1.362202	4.000798	0.0
2	40	1	15	14	55	5.5	0.856075	2.168925	0.0
4	24	2	2	0	28	17.3	1.787436	3.056564	1.0
6	39	1	20	9	67	30.6	3.833874	16.668126	0.0

Test Data

	age	ed	employ	address	income	debtinc	creddebt	othdebt	default
3	41	1	15	14	120	2.9	2.658720	0.821280	0.0
5	41	2	5	5	25	10.2	0.392700	2.157300	0.0
7	43	1	12	11	38	3.6	0.128592	1.239408	0.0
15	36	2	9	6	49	8.6	0.817516	3.396484	1.0
17	43	1	23	19	72	7.6	1.181952	4.290048	0.0

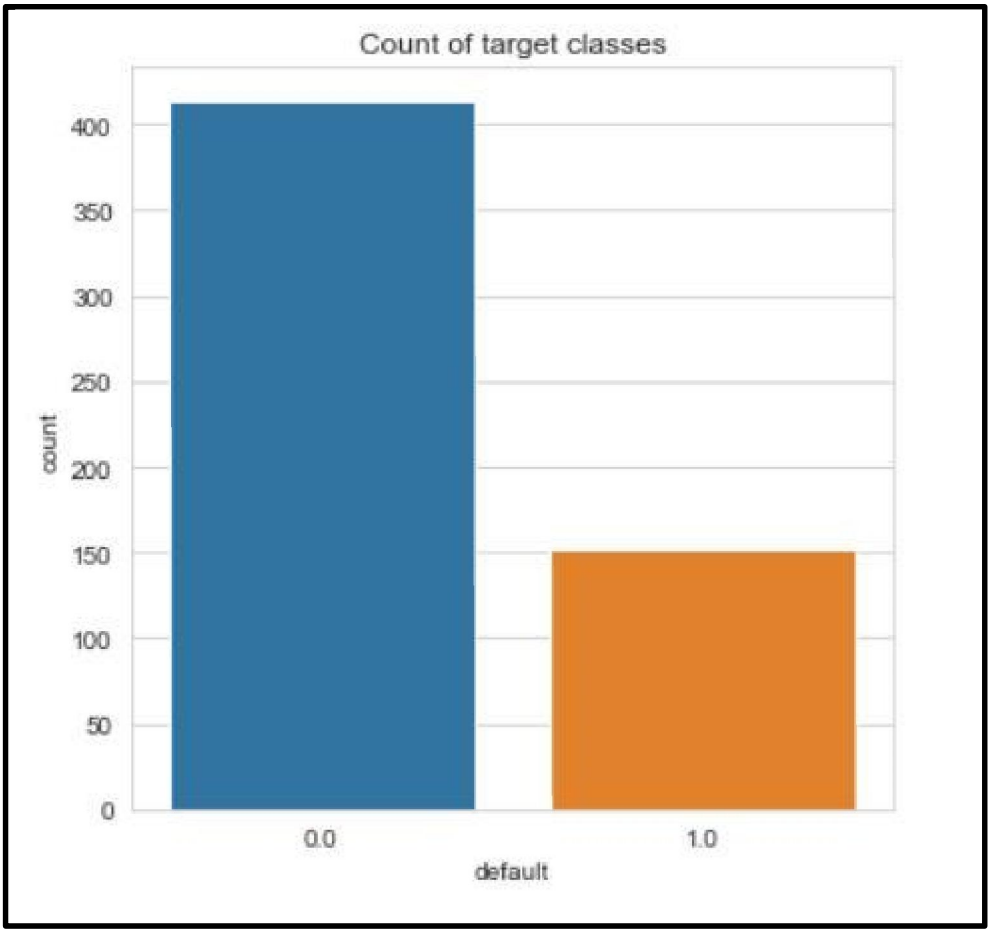
Observation :-

Through splitting the data, we have now 560 observations in the train data and 140 observations in the test data.

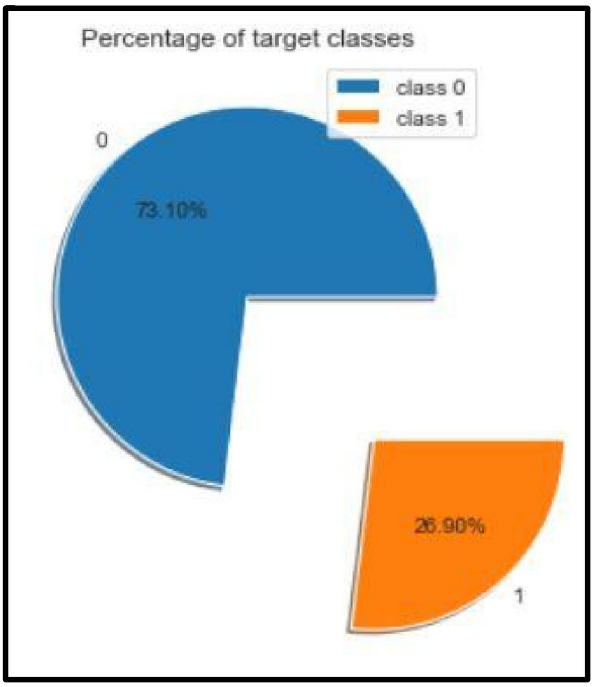
Distribution of Target Variable :-

We will check the distribution of the target variable by plotting a bar graph and pie chart.

Bar Graph



Pie Chart



Observations :-

1. The dataset is highly imbalanced.
2. The number of data points in class 0 is approximately 73%.
3. The number of data points in class 1 is approximately 27%.
4. The number of customers that will have a default status are comparatively less as compared to those who will have a non-default status.

Splitting the Train Data :-

Splitting the train data into dependent and independent variables to check the multicollinearity between the independent variables.

Independent Variables

	age	ed	employ	address	income	debtinc	creddebt	othdebt
0	41	3	17	12	176	9.3	11.359392	5.008608
1	27	1	10	6	31	17.3	1.362202	4.000798
2	40	1	15	14	55	5.5	0.856075	2.168925
4	24	2	2	0	28	17.3	1.787436	3.056564
6	39	1	20	9	67	30.6	3.833874	16.668126

Dependent Variable

```
0    1.0
1    0.0
2    0.0
4    1.0
6    0.0
```

Multicollinearity Analysis :-

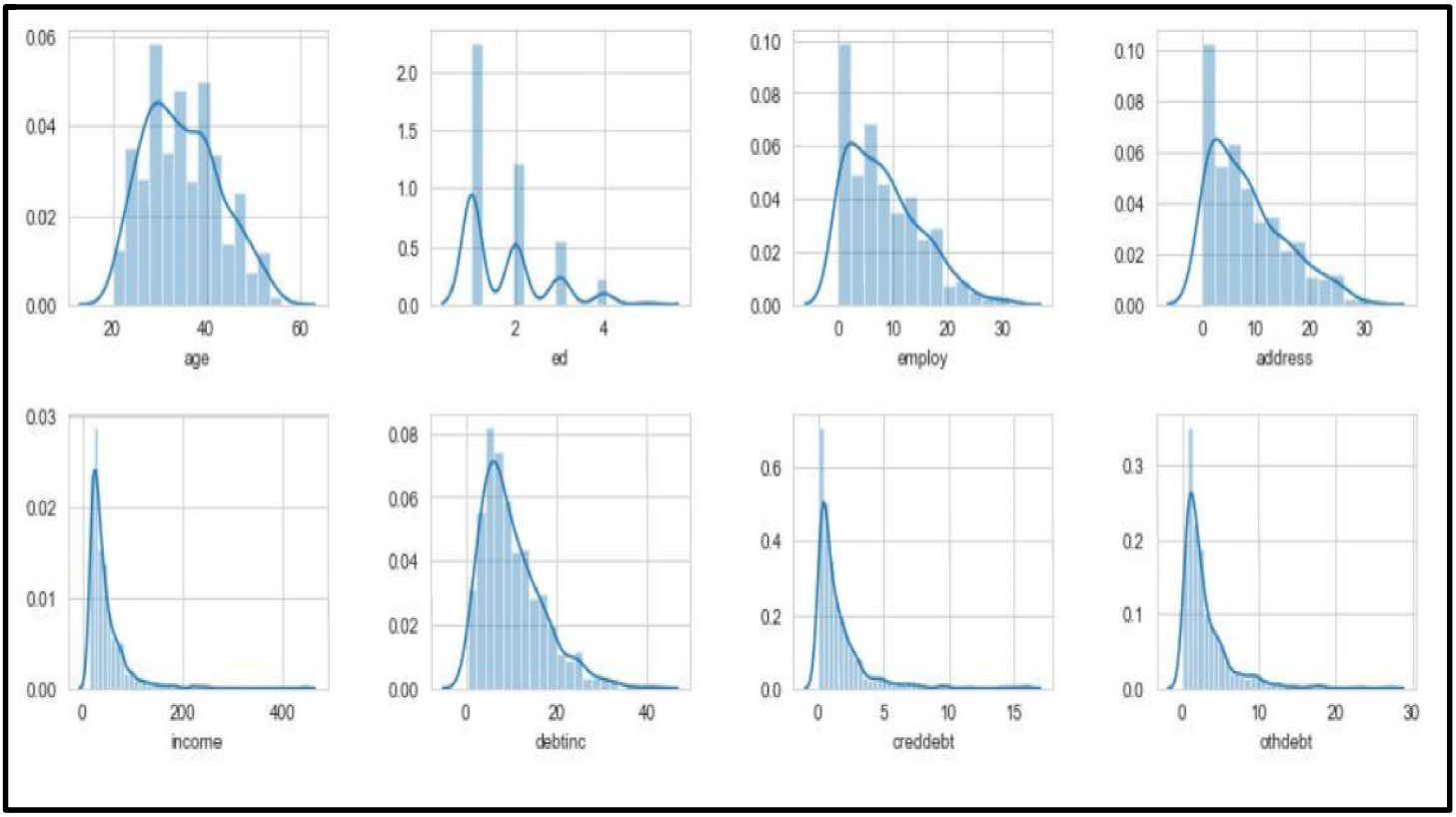
We will calculate VIF(Variance Inflation Factor) for each independent variable to check the existence of multicollinearity.

	features	vif
0	age	17.562240
1	ed	5.112207
2	employ	5.576546
3	address	3.569253
4	income	10.524638
5	debtinc	8.761947
6	creddebt	4.116065
7	othdebt	6.322753

Observation :-

Since no two variables have the same variance inflation factor. So we can continue with our modelling with all the eight independent features.

Distribution of Independent Variables :-

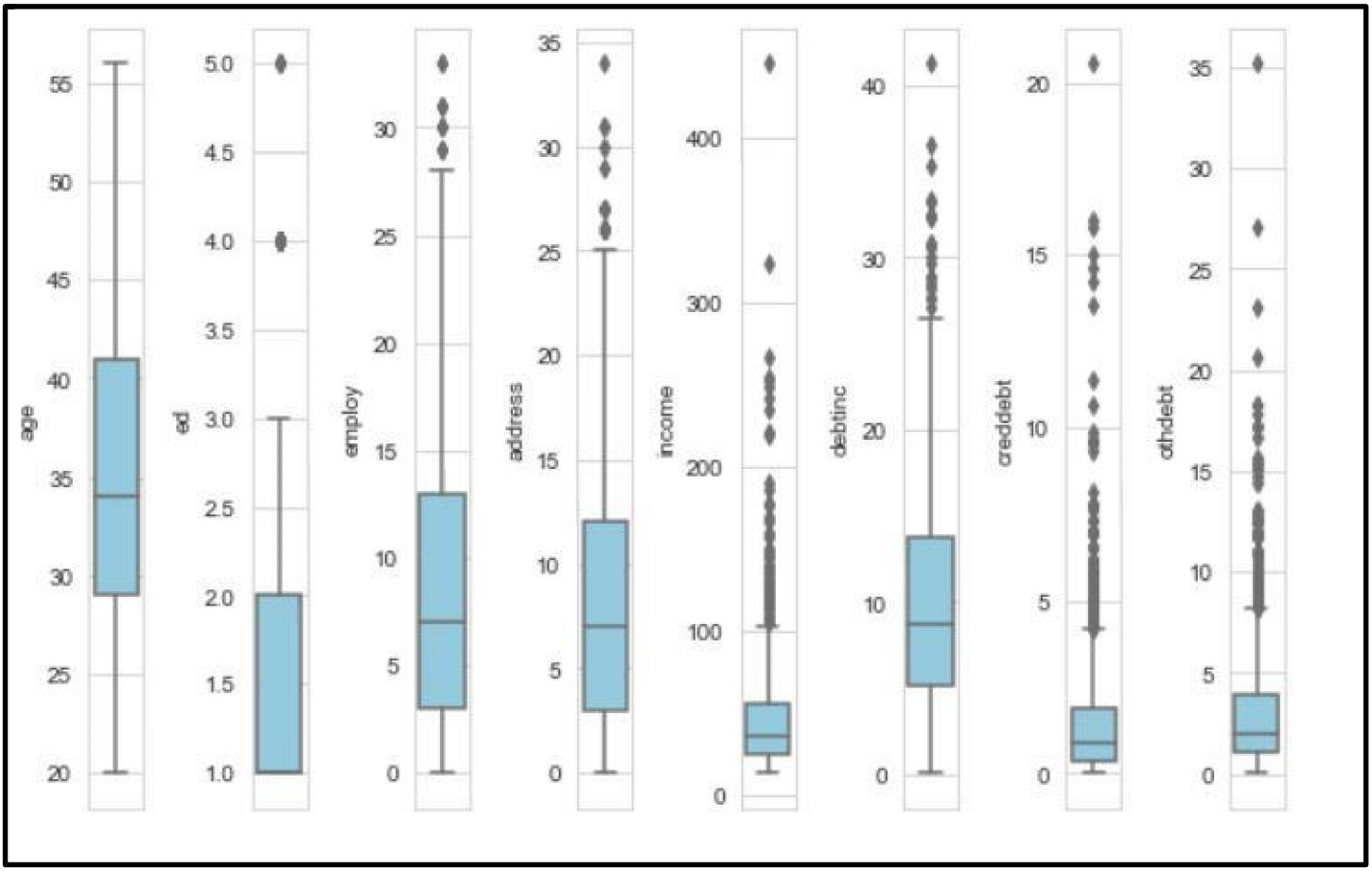


Observation :-

From the above plots, we can conclude that none of the independent variables are normally distributed. All the eight predictors are positive or right skewed.

Outlier Analysis :-

We can check outliers in the independent variables by plotting box plots.



Removing Outliers :-

We will calculate upper fence and lower fence based on IQR(Interquartile Range).
Data points which are falling below the lower fence and above the upper fence will be declared as outliers.

```
age : iqr : 11.0 : mimimum : 12.5 : maximum : 56.5
ed : iqr : 1.0 : mimimum : -0.5 : maximum : 3.5
employ : iqr : 10.0 : mimimum : -12.0 : maximum : 28.0
address : iqr : 10.0 : mimimum : -12.0 : maximum : 28.0
income : iqr : 30.0 : mimimum : -21.0 : maximum : 99.0
debtinc : iqr : 8.4 : mimimum : -7.200000000000001 : maximum : 26.400000000000002
creddebt : iqr : 1.5448439999999999 : mimimum : -1.946196 : maximum : 4.23318
othdebt : iqr : 2.797054 : mimimum : -3.1472350000000007 : maximum : 8.040981
```

Observation :-

There are 176 outliers in the dataset.

Missing Values Imputation :-

First we will replace all the outliers with the Null values and then replace all the null values with the mean of that particular variable.

age	0.000000
ed	5.840708
employ	0.707965
address	0.530973
income	6.725664
debtinc	2.654867
creddebt	7.433628
othdebt	7.256637

Standardization :-

We will use Standardization on independent variables for feature scaling in order to scale all the variables of both train and test data to have a mean of 0 and variance or standard deviation of 1.

Model Training :-

We will use following three models for training the dataset in both R and Python and then compare the performance of these models to select the best one:-

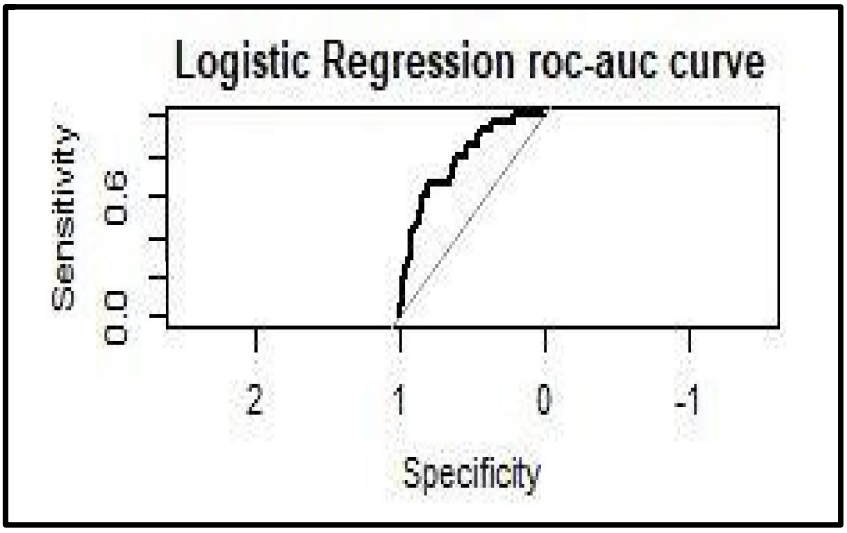
1. *Logistic Regression*
2. *Decision Tree*
3. *Random Forest*

Performance Metrics :-

We will check the performance of each model with the help of following eight performance metrics :-

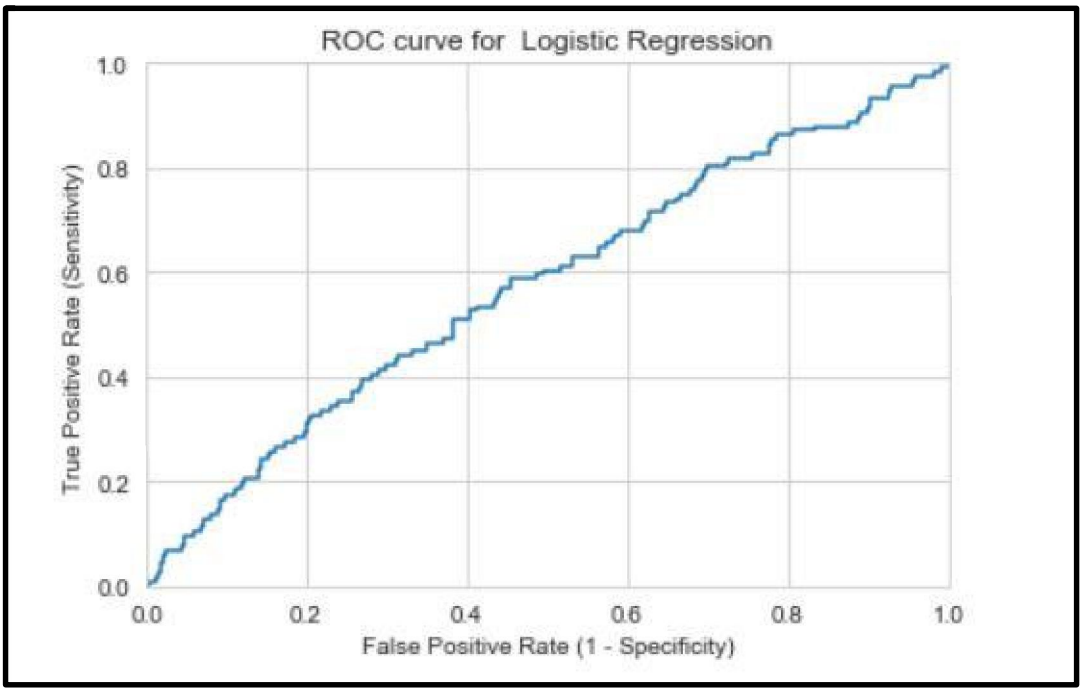
1. *Accuracy*
2. *Recall*
3. *Precision*
4. *Specificity*
5. *False Positive Rate*
6. *False Negative Rate*
7. *F1 Score*
8. *ROC-AUC curve*

Logistic Regression in R :-



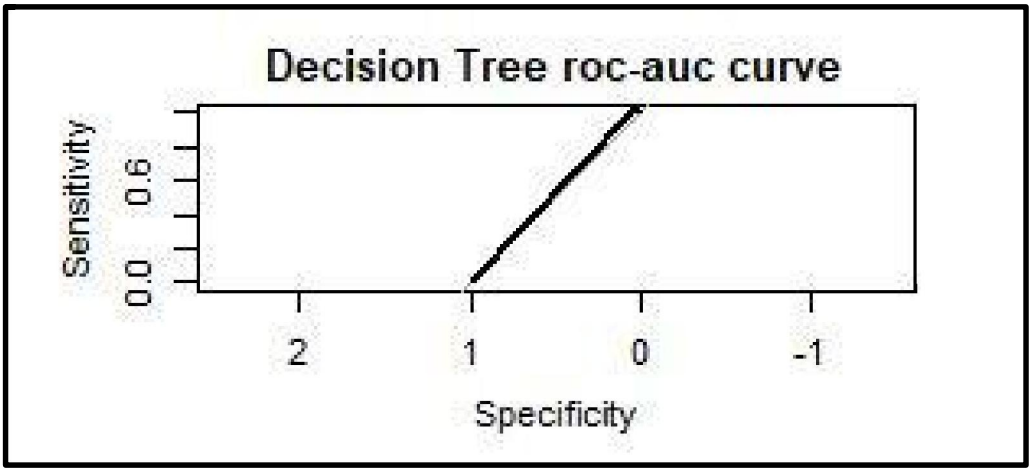
Evaluation Metrics	*Percentage Values*
Accuracy	37.85%
Recall	100%
Precision	13%
Specificity	100%
FPR	0%
FNR	87%
F1 Score	23.01%
AUC	77.76%

Logistic Regression in Python :-



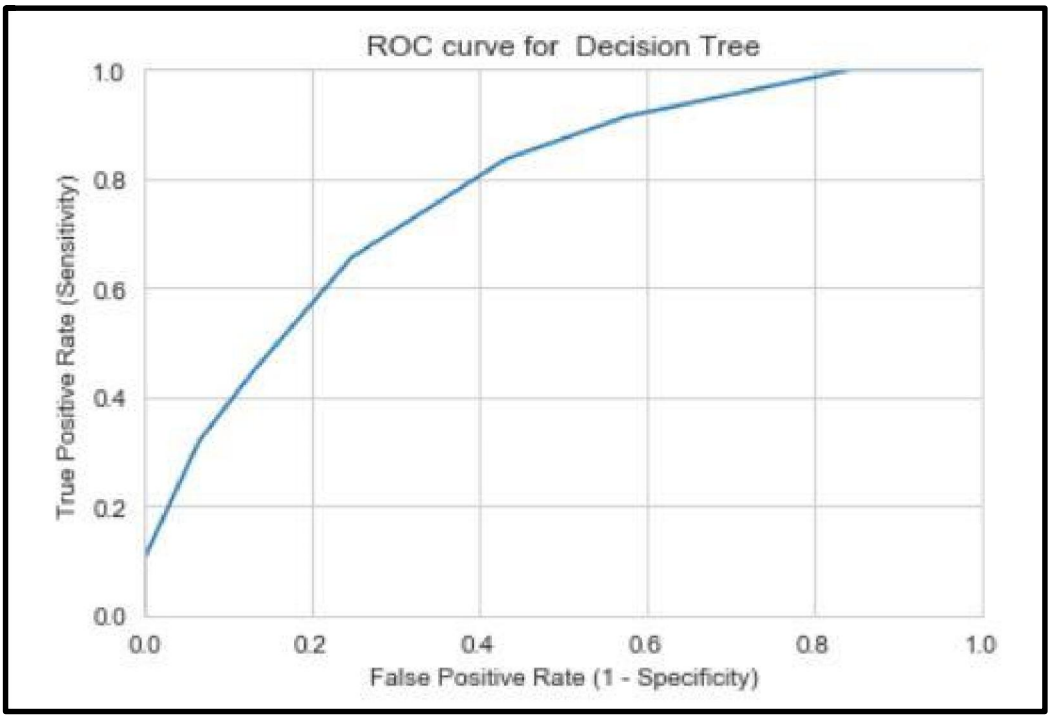
Evaluation Metrics	***Percentage Values***
Accuracy	56.09%
Recall	53.44%
Precision	30.09%
Specificity	57.01%
FPR	42.98%
FNR	46.55%
F1 Score	38.5%
AUC	57.4%

Decision Tree in R :-



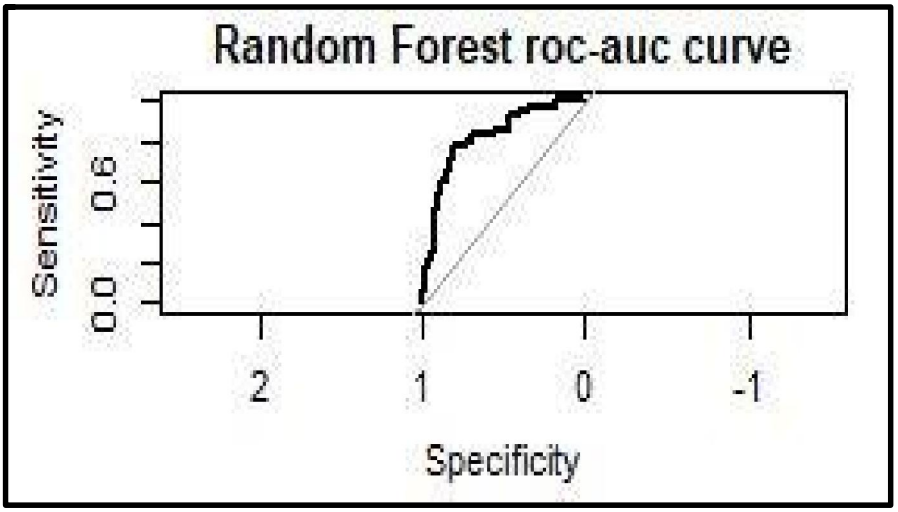
Evaluation Metrics	*Percentage Values*
Accuracy	32.14%
Recall	100%
Precision	5%
Specificity	100%
FPR	0%
FNR	95%
F1 Score	9.52%
AUC	52.5%

Decision Tree in Python :-



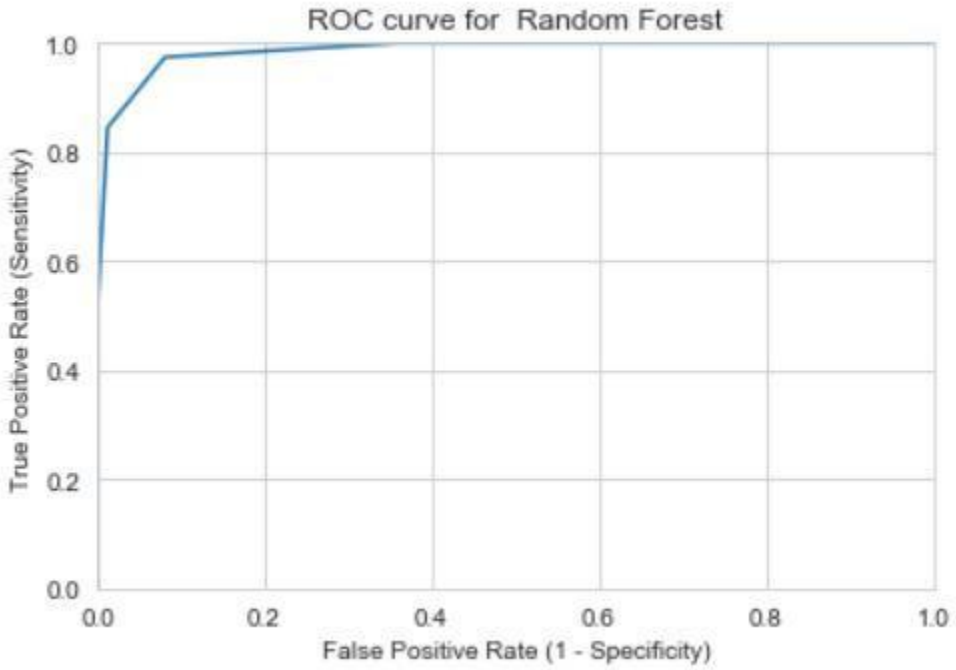
Evaluation Metrics	*Percentage Values*
Accuracy	72.72%
Recall	65.51%
Precision	47.79%
Specificity	75.22%
FPR	24.77%
FNR	34.48%
F1 Score	55.27%
AUC	77.9%

Random Forest in R :-



Evaluation Metrics	***Percentage Values***
Accuracy	79.28%
Recall	82.56%
Precision	90%
Specificity	52.5%
FPR	47.5%
FNR	10%
F1 Score	86.12%
AUC	82.15%

Random Forest in Python :-



Evaluation Metrics	***Percentage Values***
Accuracy	95.12%
Recall	84.48%
Precision	96.07%
Specificity	98.8%
FPR	1.19%
FNR	15.51%
F1 Score	89.9%
AUC	98.6%

Models Comparison in R :-

	Accuracy	*Recall*	*Precision*	*Specificity*	*FPR*	*FNR*	*F1 Score*	*AUC*
Logistic Regression	0.3785	1	0.13	1	0	0.87	0.2301	0.7776
Decision Tree	0.3214	1	0.05	1	0	0.95	0.0952	0.525
Random Forest	0.7928	0.8256	0.9	0.525	0.475	0.1	0.8612	0.8215

Models Comparison in Python :-

	Accuracy	*Recall*	*Precision*	*Specificity*	*FPR*	*FNR*	*F1 Score*	*AUC*
Logistic Regression	0.5609	0.5344	0.3009	0.5701	0.4298	0.4655	0.385	0.574
Decision Tree	0.7272	0.6551	0.4779	0.7522	0.2477	0.3448	0.5527	0.779
Random Forest	0.9512	0.8448	0.9607	0.988	0.0119	0.1551	0.899	0.986

Model Selection Parameters :-

The model should be selected based on the following parameters :-

1. *High Accuracy*
2. *High F1 Score*
3. *High AUC Score*
4. *High Recall*
5. *High Precision*
6. *High Specificity*
7. *Low FPR*
8. *Low FNR*

Freezed Model :-

Since Random Forest performs much better than other models based on the above parameters in both R and Python, we will freeze Random Forest as our final model for predicting the target class of test data.

Conclusion :-

The above freezed model can be used for ***Bank Loan Default Case*** *problem to classify* ***which customers will have a default loan application and which customers will have a non-default loan application.***