COMPREHENSIVE PROJECT REPORT

★Contents

1. Preprocessing of Data

1.1 Python packages and methods used:

1) pandas
2) numpy
3) from fancyimpute import KNN
4) import matplotlib.pyplot as plt
5) import seaborn as sns
6) from sklearn.datasets import make_classification
7) from sklearn.linear_model import LogisticRegression
8) from sklearn.model_selection import train_test_split
9) from sklearn.metrics import roc_curve
10) from sklearn.metrics import roc_auc_score
11) from matplotlib import pyplot
12) from sklearn.metrics import precision_recall_curve
13) from sklearn.metrics import auc
14) import statsmodels.api as sm
15) from sklearn.metrics import confusion_matrix
16) from sklearn.linear_model import LogisticRegression
17) import scikitplot as skplt
18) from sklearn import tree
19) from sklearn.metrics import accuracy_score
20) from sklearn.ensemble import RandomForestClassifier
21) from sklearn.neighbors import KNeighborsClassifier
22) from sklearn.preprocessing import StandardScaler
23) from sklearn.metrics import roc_auc_score
24) from sklearn.naive_bayes import GaussianNB
25) from sklearn.datasets import make_classification
26) from sklearn.ensemble import GradientBoostingClassifier
27) from xgboost import XGBClassifier
28) from sklearn.svm import SVC

1.2 R packages and methods used:

1) ggplot2
2) corrgram
3) DMwR
4) caret
5) randomforest
6) unbalanced
7) C50

8) inTrees
9) dummies
10) e1071
11) Information
12) MASS
13) rpart
14) gbm
15) ROSE
16) sampling
17) RRF
18) precrec
19) pROC
20) funModeling
21) xgboost

1.3 Inputing data to predict:

NOTE: These values have been taken from the dataset itself which results to default value=1, i.e.
Sample Data: (age:41, ed:3, employ:17, address:12, income:176, debtinc:9.3, creddebt:11.359392, othdebt:5.008608 -> default: 1)

Enter age41
Enter education3
Enter employ17
Enter address12
Enter income176
Enter debtinc9.3
Enter creddebt11.3593
Enter othdebt5.008

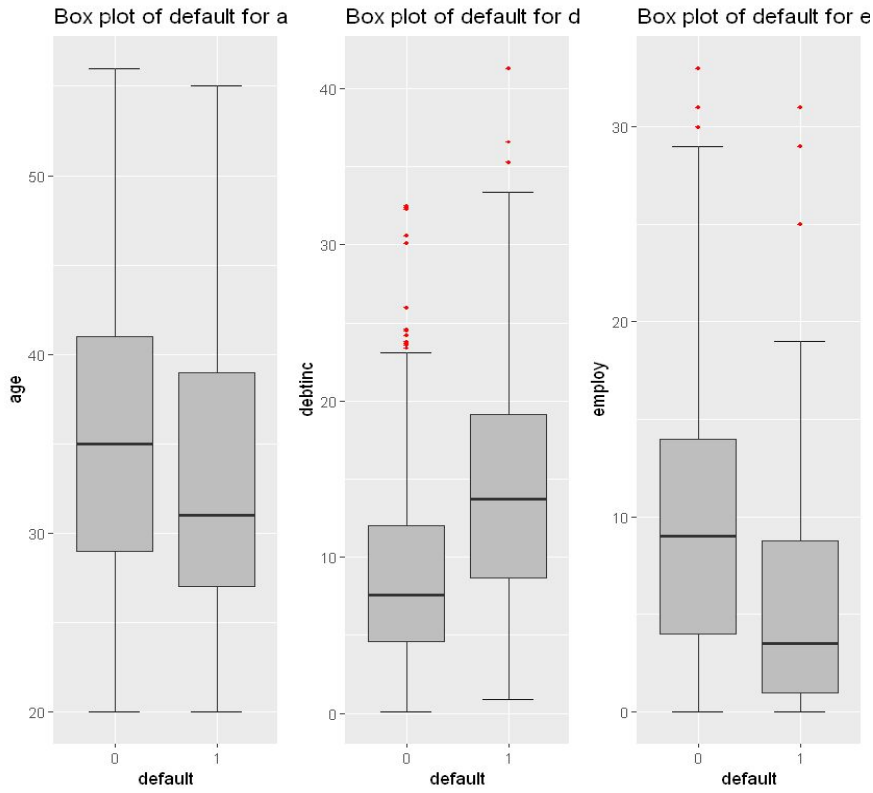
1.4 Imputing Missing Values: Imputing missing values of the "default" column's last 150 variables in the bank-loan.csv file.

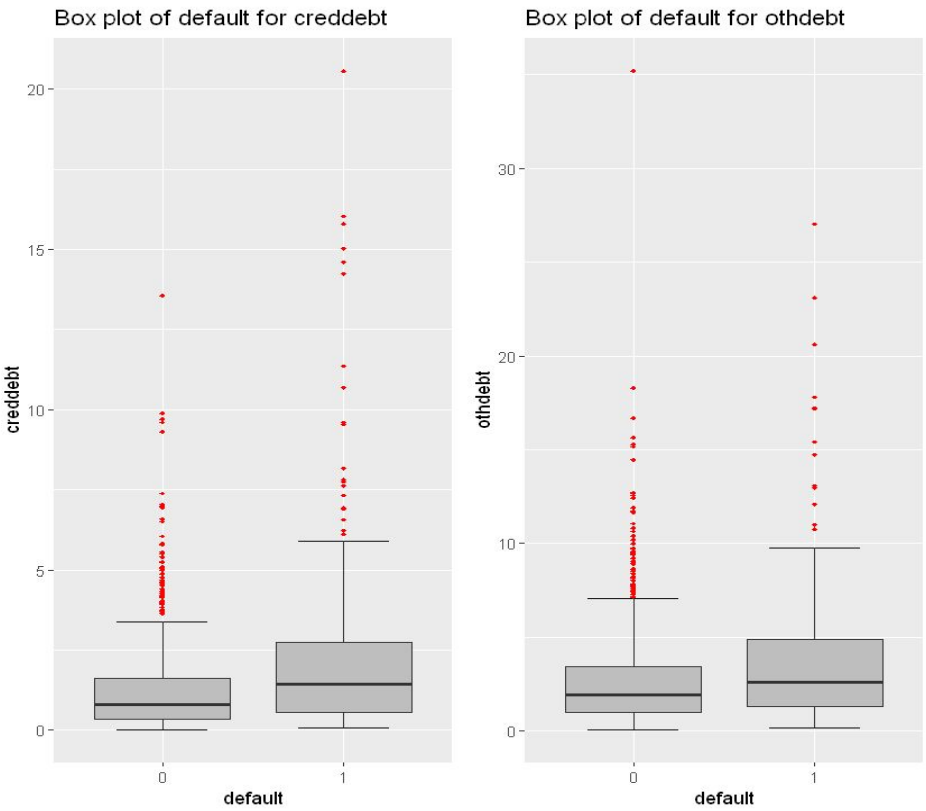
Using TensorFlow backend.

Imputing row 1/851 with 7 missing, elapsed time: 0.235

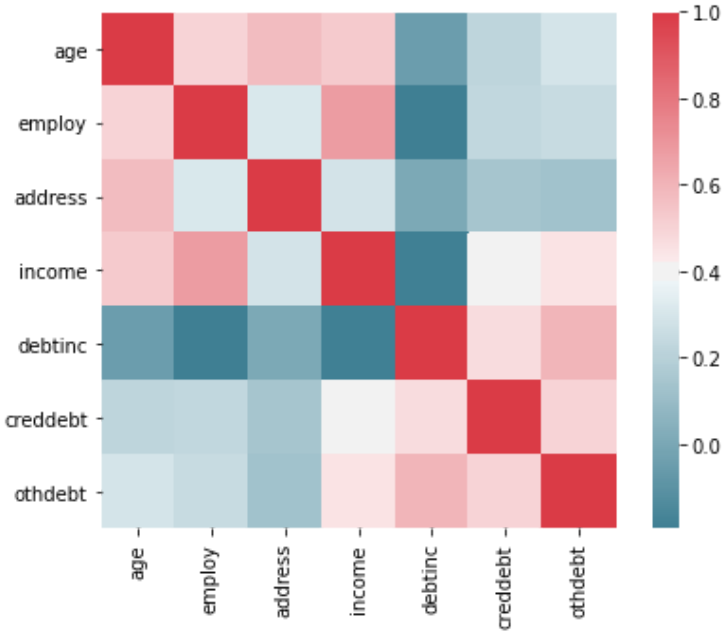
Imputing row 101/851 with 5 missing, elapsed time: 0.239
Imputing row 201/851 with 0 missing, elapsed time: 0.242
Imputing row 301/851 with 8 missing, elapsed time: 0.247
Imputing row 401/851 with 0 missing, elapsed time: 0.252
Imputing row 501/851 with 0 missing, elapsed time: 0.259
Imputing row 601/851 with 0 missing, elapsed time: 0.265
Imputing row 701/851 with 0 missing, elapsed time: 0.269
Imputing row 801/851 with 0 missing, elapsed time: 0.274

1.5 Outlier Analysis: Performing outlier analysis on data set and plotting a boxplot on the columns, and cleaning the data by replacing them with NA and them imputing missing values.

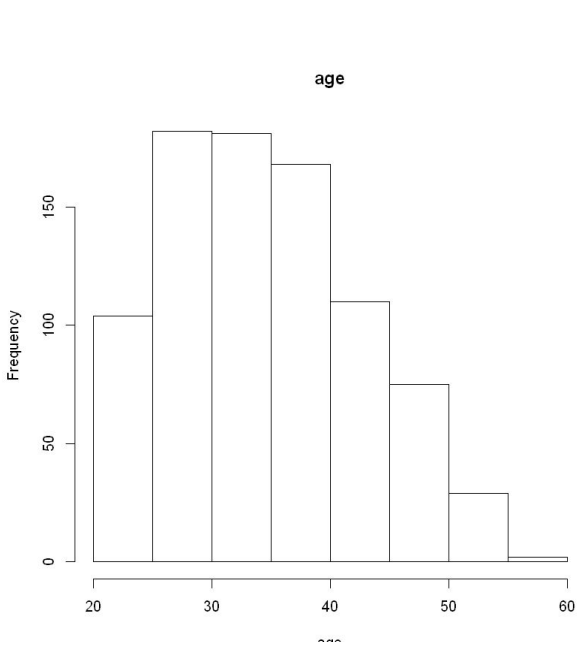


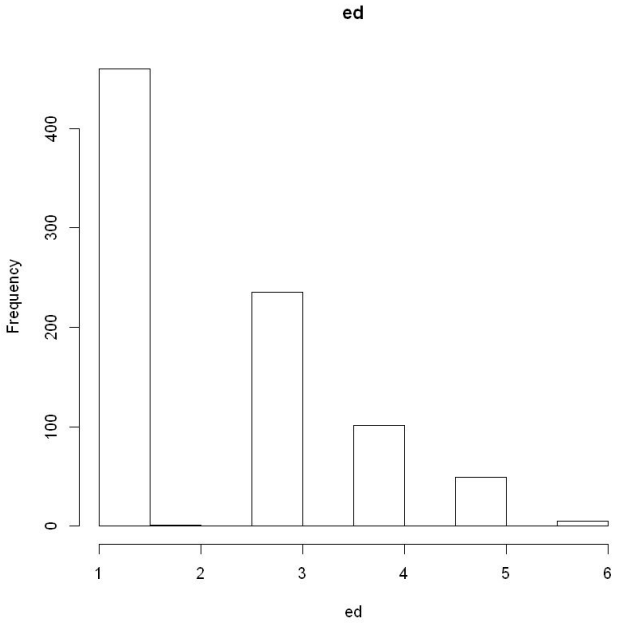


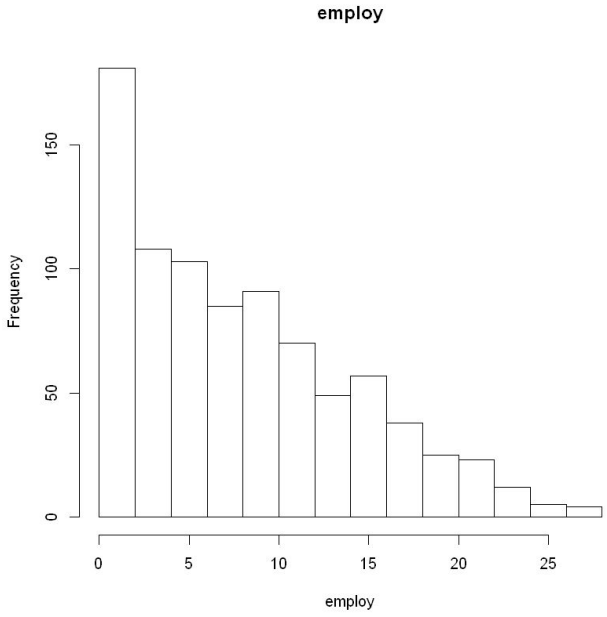
1.6 Correlation Analysis: Performing correlation analysis on data set and extracting the columns which give us meaningful data, while dropping the ones ("debtinc" column in this case) which are highly correlated.

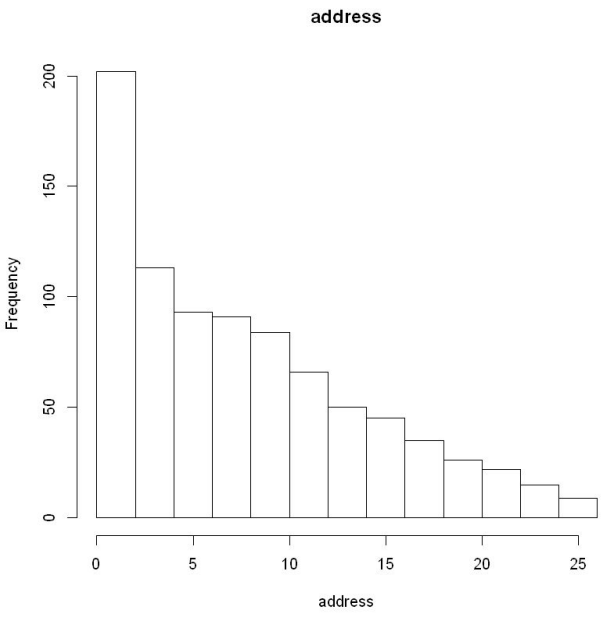


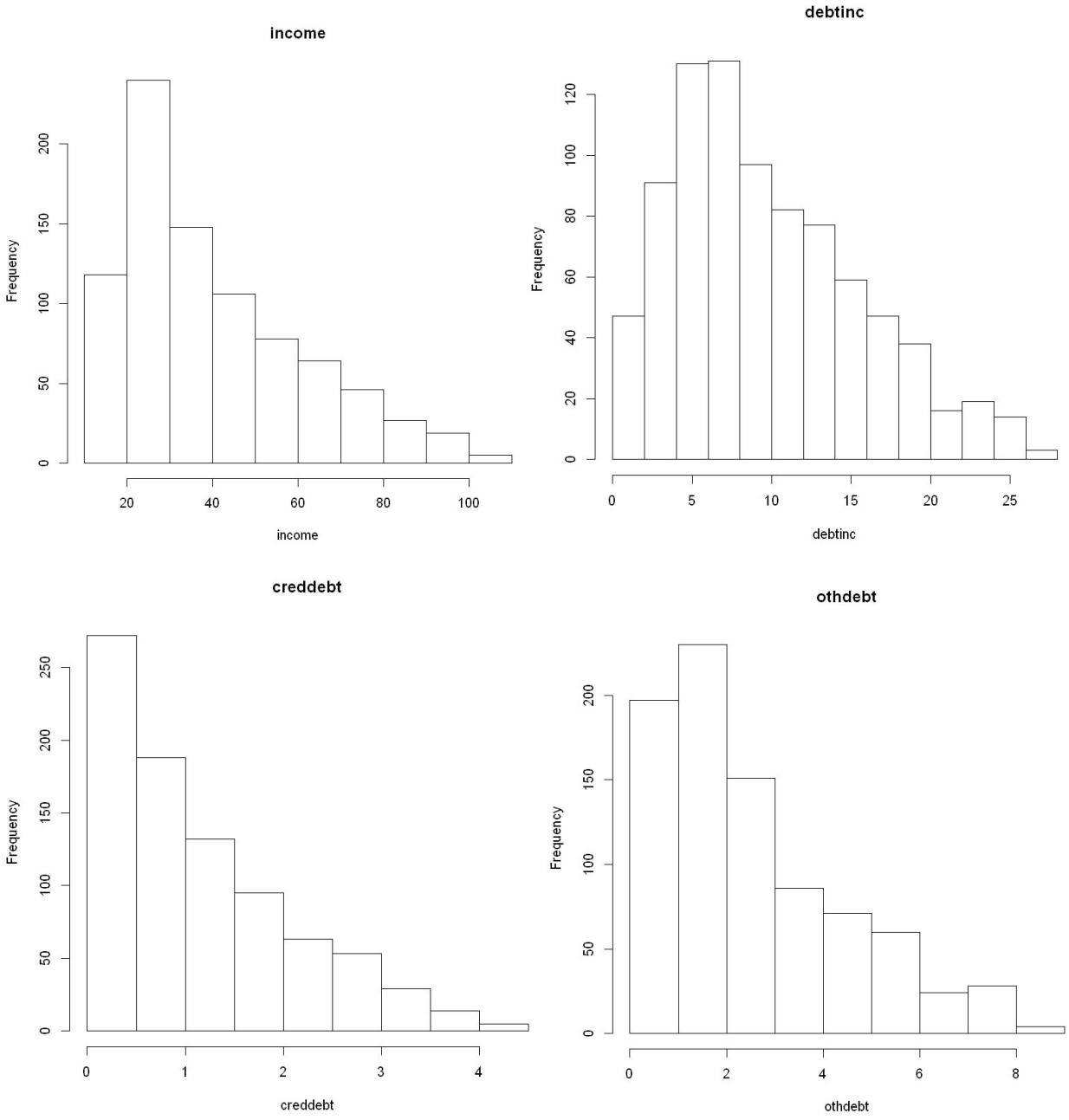
1.7 Feature Scaling: Performing feature Extraction by applying standardization on the dataset, as we can see by the below-mentioned histograms that standardization would be a better way to go with, rather than normalization in this case.











2.Model summary with different curves and charts

2.1 Multiple Logistic regression

After pre-processing of data to multiple logistic regression classifier we got the following results:

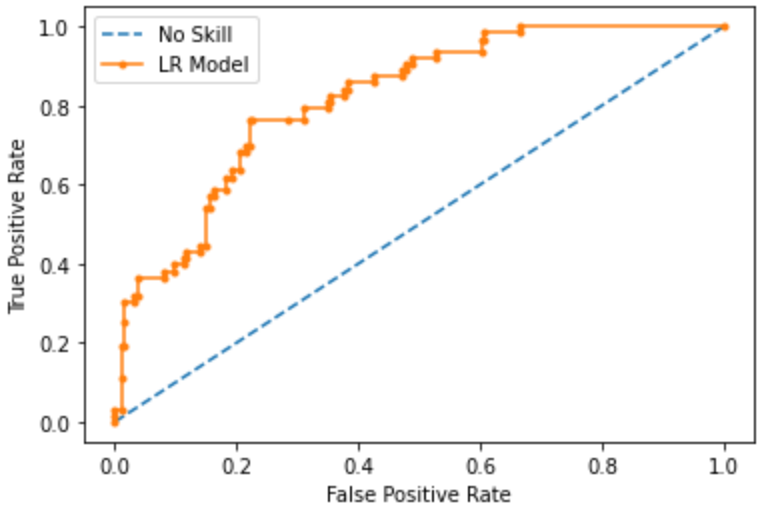
For input data:
850 0.466765
dtype: float64
Default value from Logistic Regression is 0

Confusion matrix:

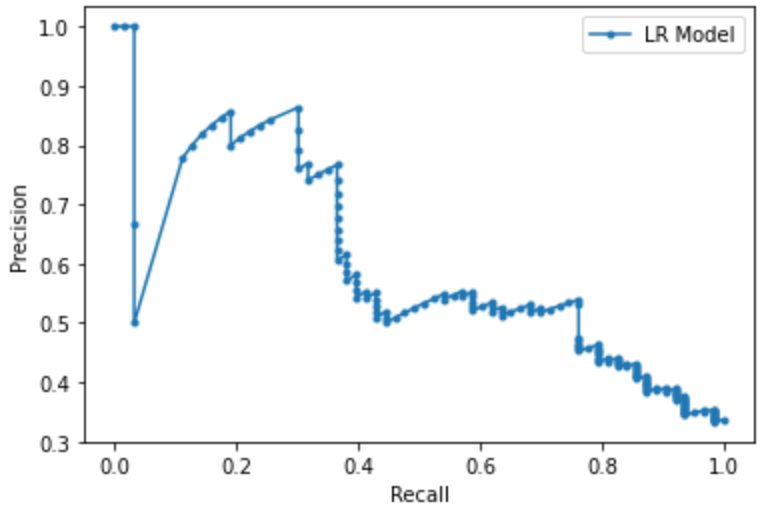
Actual_val	0	1
pred_val		
0	182	7
1	43	29

Accuracy from Logistic regression = 79.91967871485944
Recall = 40.158730158730158

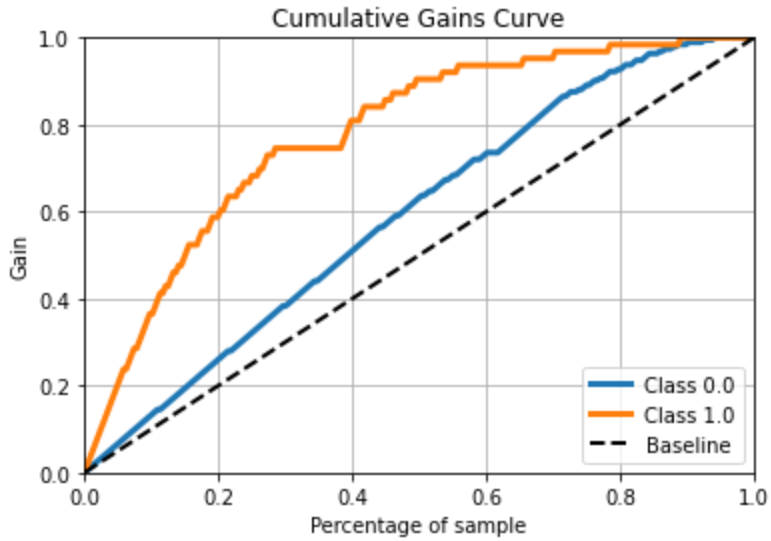
ROC-Curve:
LR Model No Skill: ROC AUC=0.500
LR Model ROC AUC=0.817



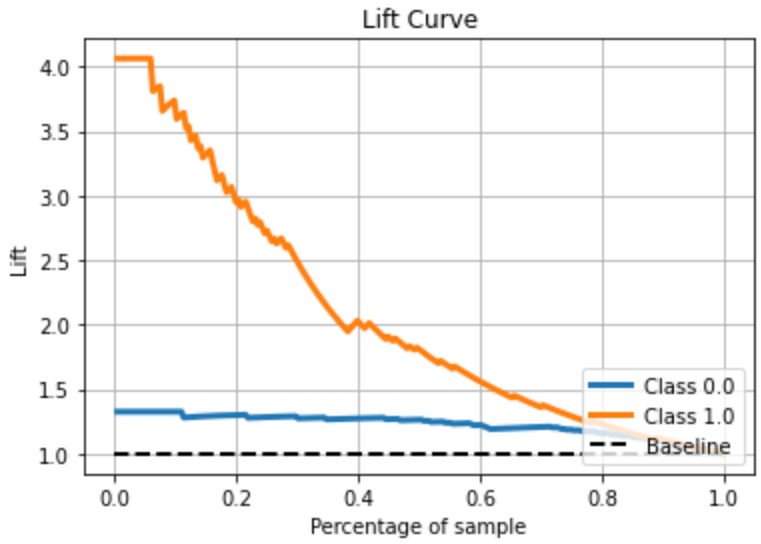
PR-Curve:
LR Model PR AUC=0.596



Gain-chart:



Lift-curve:



Conclusion of the model:

In our Bank loan default case model generated by the multiple Logistic regression classifier, we can see that it has an acceptable ROC-AUC value, also the Accuracy is quiet standard but on the other hand its PR-AUC value which tells us how the Precision and recall of the model are varying. With each other is quite low, not to mention if we can only predict 40% of the True positive cases then our model is not good enough. So, we'll **reject** this model based on its **Recall value.**

2.2 Decision Tree Model:

After pre-processing of data to Decision Tree classifier we got the following results:

For input data:

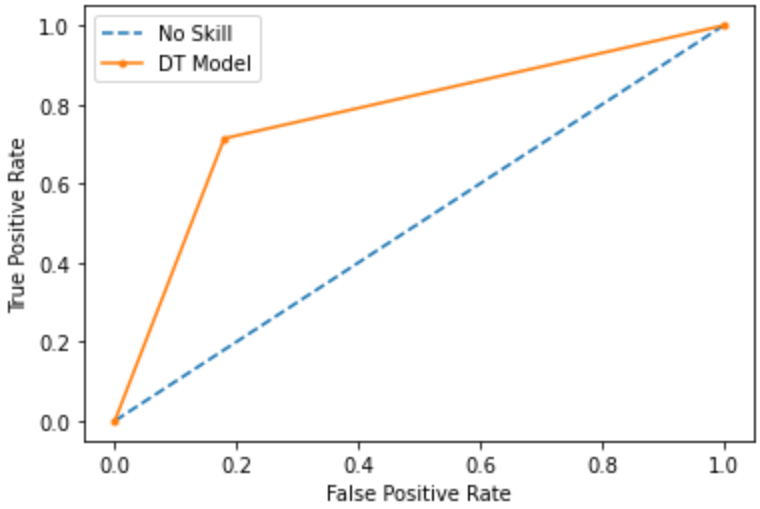
DT Model
Default value for input data is: 1

Confusion matrix:

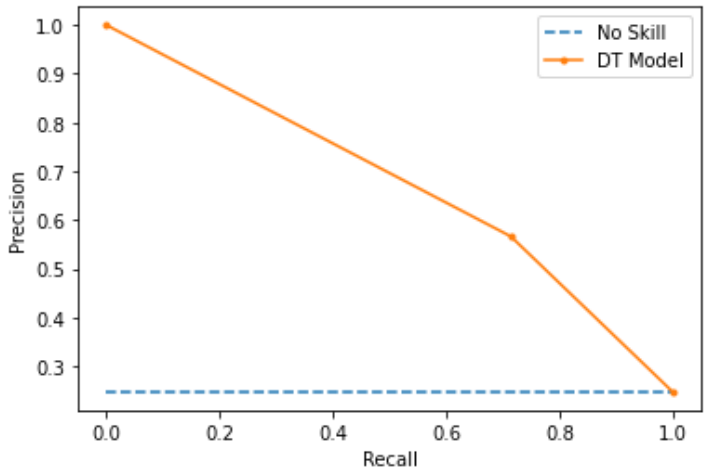
Actual_val	0	1
pred_val		
0	110	23
1	10	27

Accuracy from DT : 79.41176470588235
Recall: 72.42857142857143

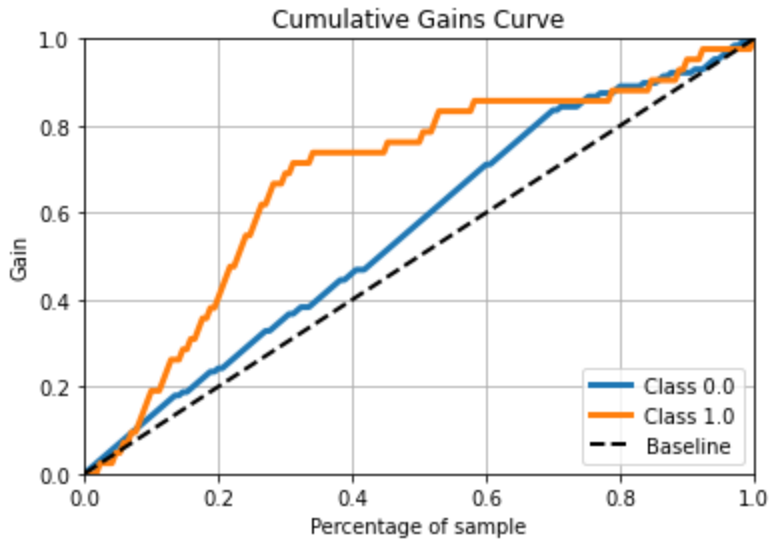
ROC-Curve:
DT Model No Skill: ROC AUC=0.500
DT Model ROC AUC=0.767



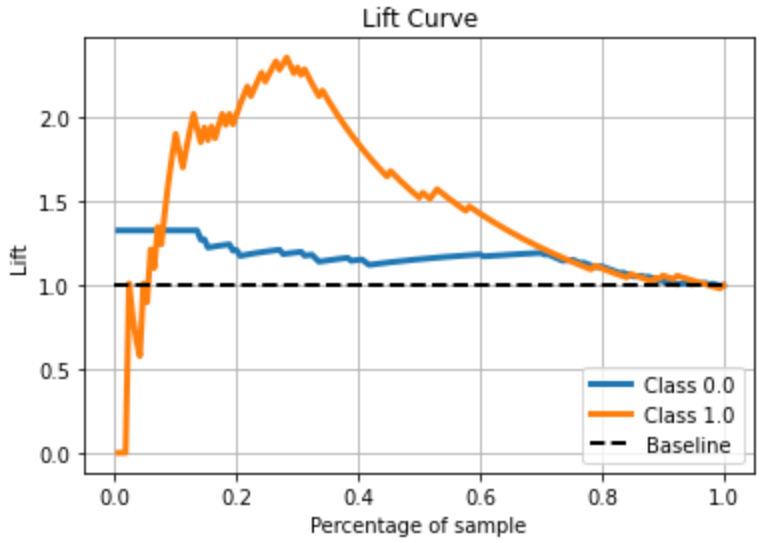
PR-Curve:
Logistic: auc=0.675



Gain-chart:



Lift-curve:



Conclusion of the model:

In our Bank loan default case model generated by the Decision tree classifier Both the accuracy and the recall value are decent enough to be accepted but contradicting to that it's PR-AUC value is quite low which is 0.675 which means that even if we are able to distinguish most of the true positive cases it would not be efficient, if we have a low budget like suppose an example where we have a budget to only reach out to 20% of our customers then even if we can distinguish most of the true positive values correctly (71% in this case), if we can't minimize our sample cases then the recall value states to nothing.

In simpler words let's suppose an example where we have 100 customers and we have a budget to reach out to only 20 of our customers now what PR-AUC value of 0.675 states that is for the true positives to be 71% we have to consider a sample set of around 67% of our total

customers as the sample set will also include the false positives. The main reason why the PR-AUC value is 0.675 even with a considerable high recall rate is because of precision. Hence even though the model has a good recall value we will **reject** it because it requires a good amount of sample cases to project 71% recall value. But, in cases where budget is not the main concern, then we can definitely cope up with the model and accept it.

2.3 Random Forest classifier:

After pre-processing of data to Random forest classifier we got the following results:

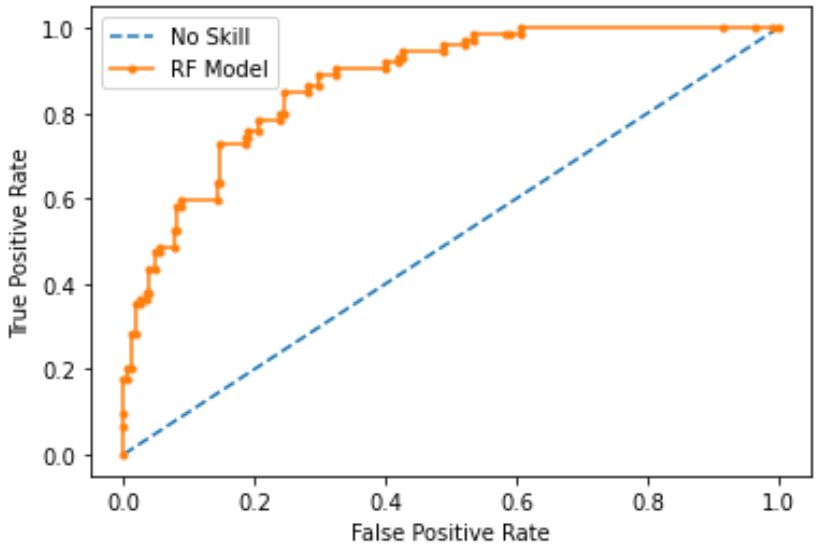
For input data:
RF Model
Default value for input data is: 1

Confusion matrix:

Actual_val	0	1
pred_val		
0	213	19
1	33	41

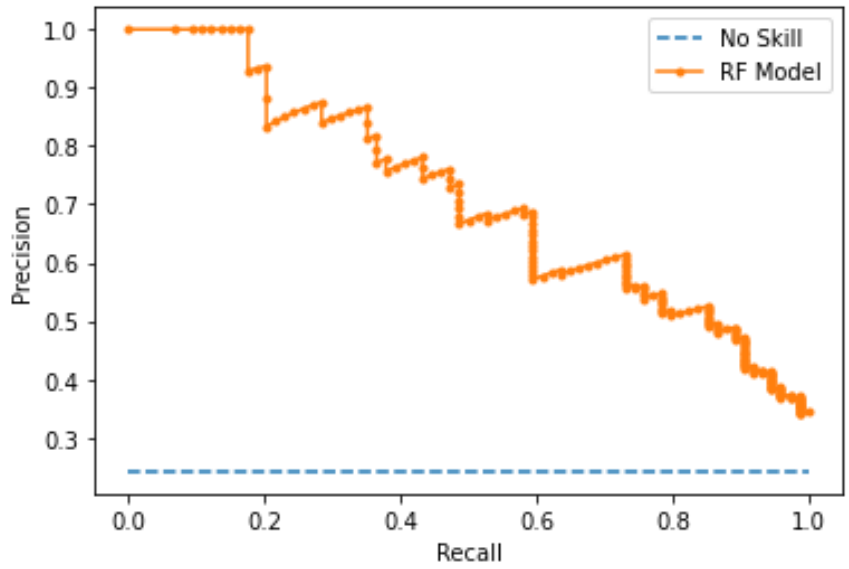
Accuracy from Random Forest: 83.00653594771242
Recall from Random Forest: 55.4054054054054

ROC-Curve:
RF Model No Skill: ROC AUC=0.500
RF Model ROC AUC=0.874

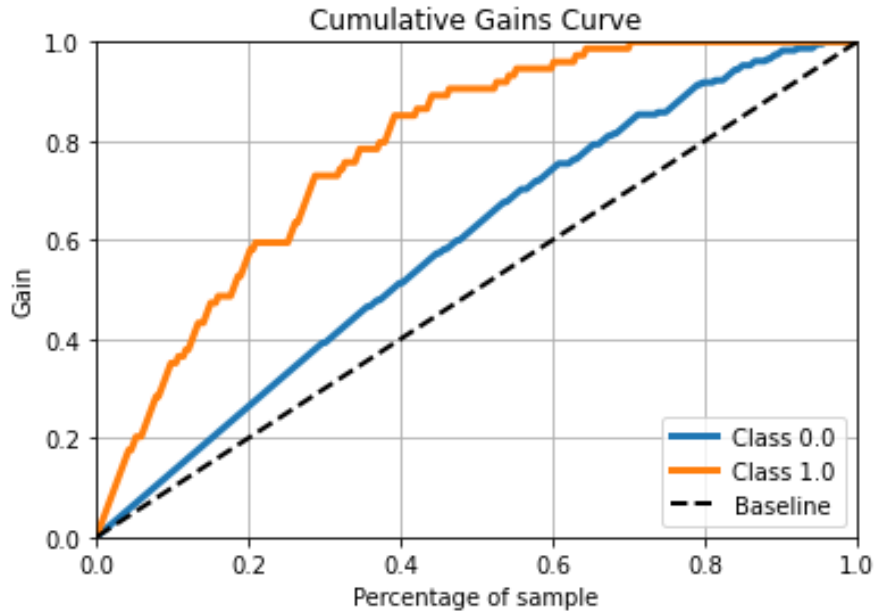


PR-Curve:

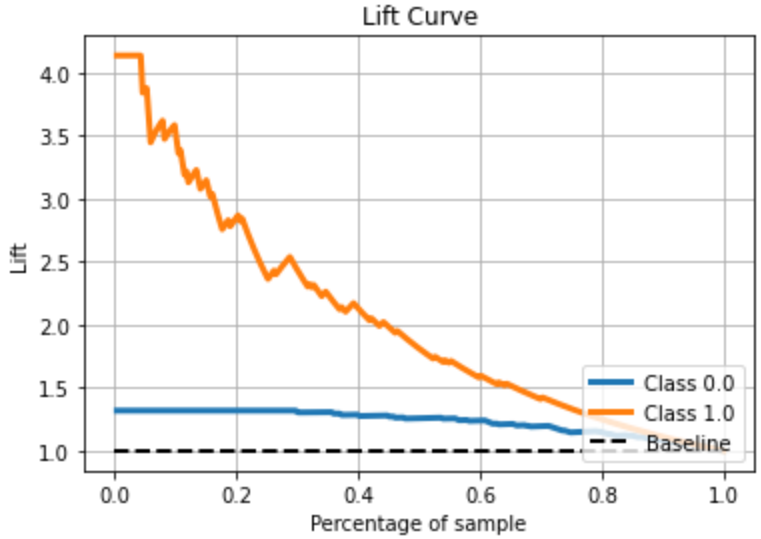
RF Model PR AUC = 0.713



Gain-chart:



Lift-curve:



Conclusion of the model:

In our Bank loan default case model generated by the Random Forest classifier, all the parameters values such as ROC-AUC value, PR-AUC value and even accuracy which is 83% is good enough to accept the model based on different parameters, but until we can't find another model which gives us better results than the random forest so we have to hold this result till then, also there is a parameter which does not meet up to our expectations i.e recall

value which is only 55%, But according to user requirements such as if you want to calculate how many customers will not take up a loan, in that case, the model can be beneficiary. Also, we can see that most of the true positives are classified at The beginning of a sample data set. So if we care about taking a minimal data set to get most of the positive values without caring about the recall then this model can be the one that we can apply for our use.

2.4 K-nearest Neighbour Model:

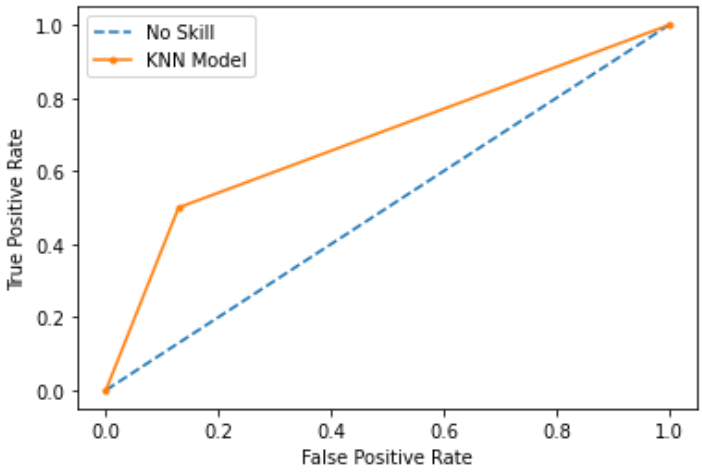
After pre-processing of data to KNN model we got the following results:

For input data:
KNN Model
Default value for input data is: 1

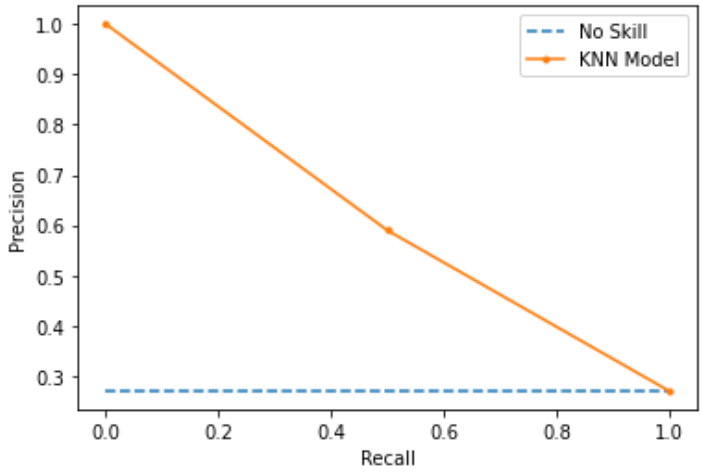
Confusion matrix:

Actual_val	0	1
pred_val		
0	108	16
1	23	23

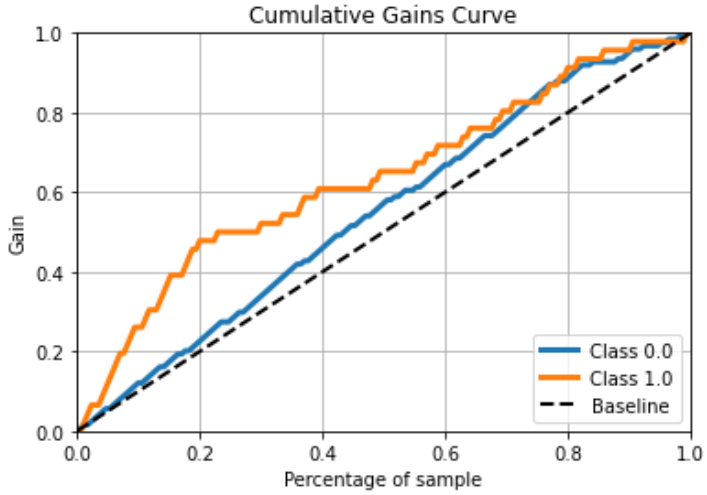
KNN accuracy: 77.05882352941177
Recall: 50.0
ROC-Curve:
KNN Model No Skill: ROC AUC=0.500
KNN Model ROC AUC=0.685



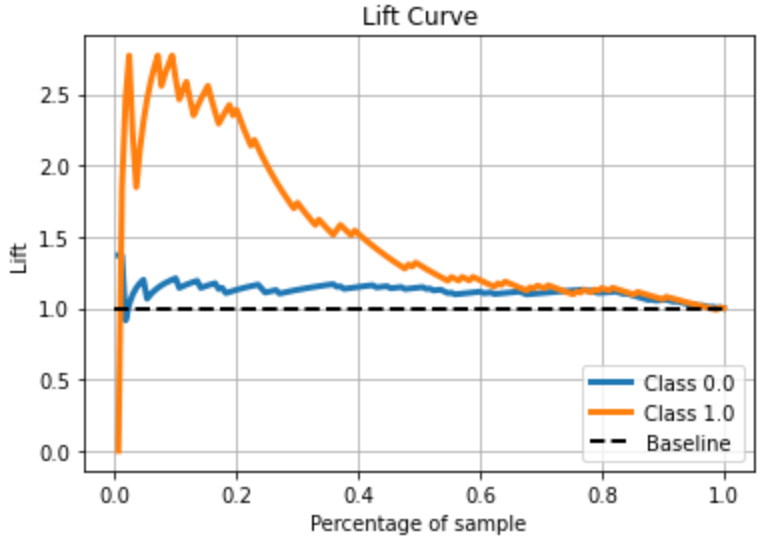
PR-Curve:
KNN Model PR AUC = 0.613



Gain-chart:



Lift-curve:



Conclusion of the model:

In our Bank loan default case model generated by the KNN model, all the parameters values such as ROC-AUC value, PR-AUC value, and even accuracy doesn't stand up to our requirement but even then we can tell advantage of the model which is the lift curve classified the most of the true positives at the beginning of the sample data set and a disadvantage is that the cumulative gains curve which tells us how much better our model is if we would have rather classified it randomly. Hence, we'll reject the model.

2.5 Naive Bayes Model:

After pre-processing of data to Naive Bayes model we got the following results:

For input data:
NB Model
Default value for input data is: 0

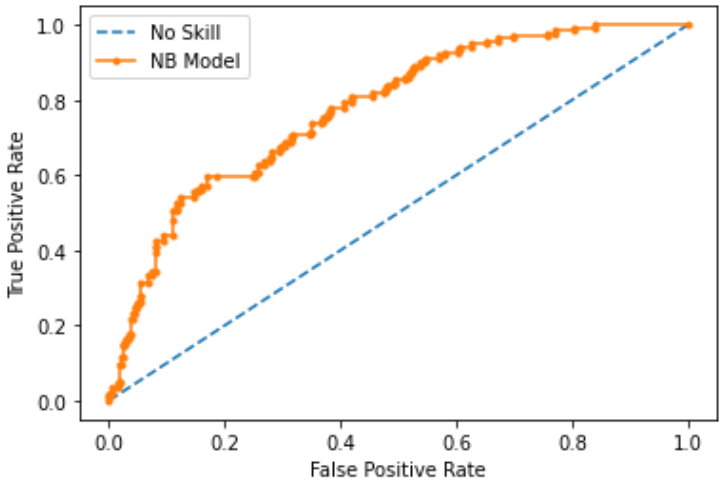
Confusion matrix:

Actual_val	0	1
pred_val		
0.0	308	108
1.0	53	84

Naive Bayes Accuracy: 70.88607594936708
Recall: 61.31386861313869

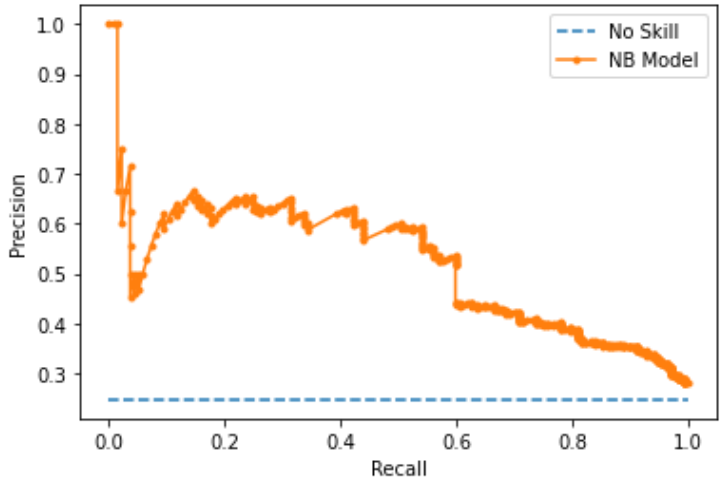
ROC-Curve:

NB Model No Skill: ROC AUC=0.500
NB Model ROC AUC=0.778

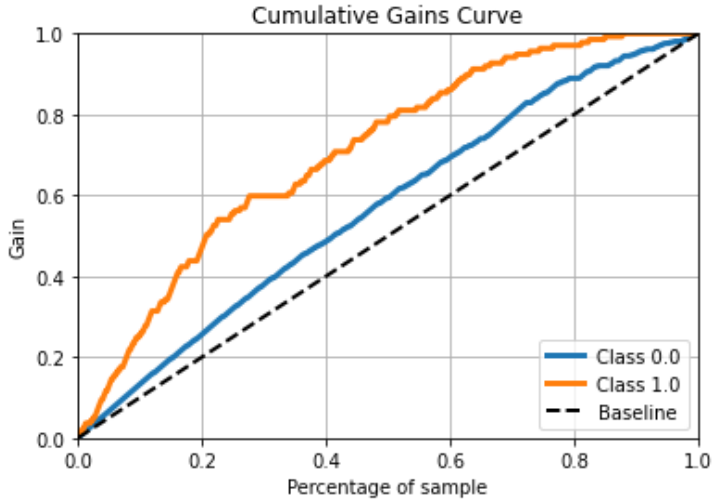


PR-Curve:

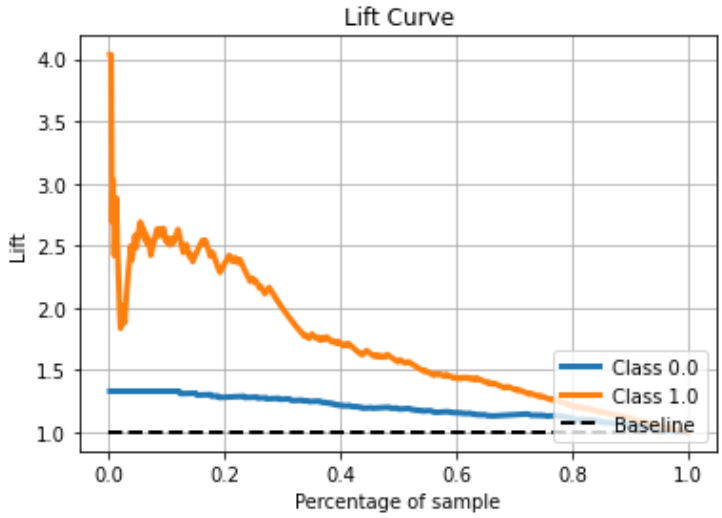
KNN Model PR AUC = 0.519



Gain-chart:



Lift-curve:



Conclusion of the model:

In our Bank loan default case model generated by the NB model, The accuracy is 70% while the record is quite impressive which is 61% considering the fact that the true positives in the actual dataset are much less than true negatives approximately in 1:6 ratio, though this doesn't change the PR-AUC value which is 0.519 which is quite similar to one of the previous models in which we have to consider a considerable amount of sample days data set to get a satisfactory amount of true positives. Hence we can reject the model on the speculation of its poor parameters excluding the recall value.

2.6 Gradient Boosting Algorithm:

After pre-processing of data to Gradient Boosting Algorithm we got the following results:

For input data:

GB Model

Default value for input data is: 1

Confusion matrix:

Actual_val	0	1
pred_val		
0.0	116	8
1.0	22	24

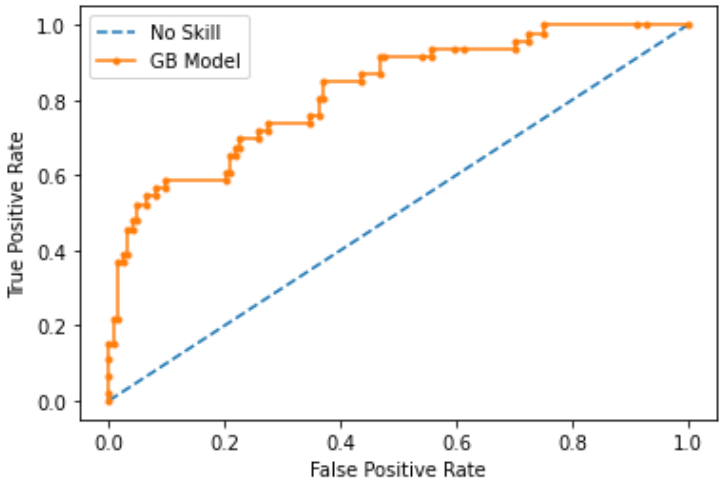
Gradient Boosting Algorithm 82.0

Recall 52.17391304347826

ROC-Curve:

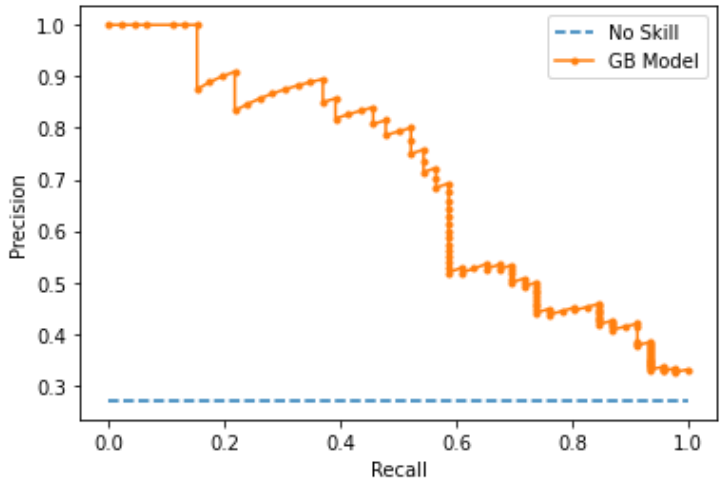
GB Model No Skill: ROC AUC=0.500

GB Model ROC AUC=0.823

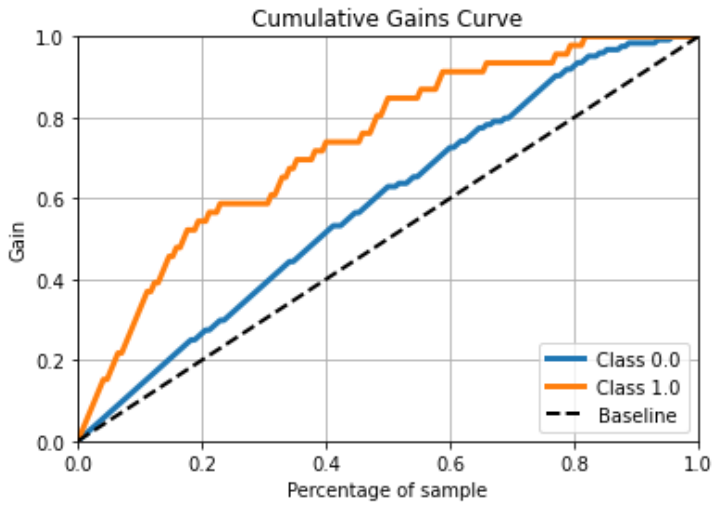


PR-Curve:

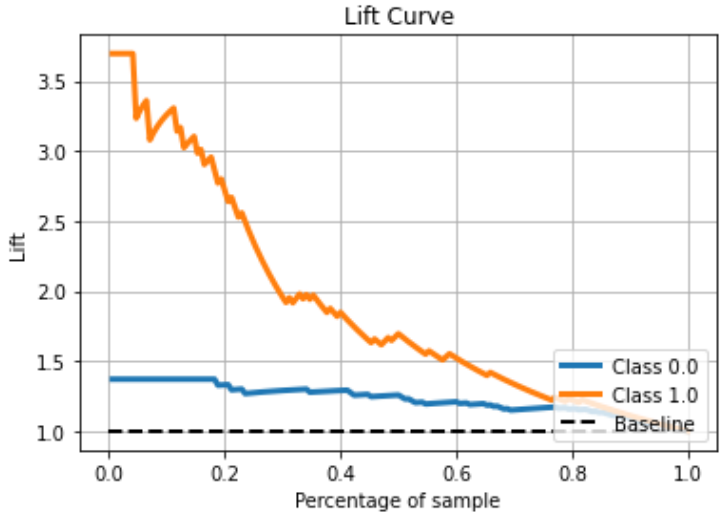
GB Model PR AUC = 0.699



Gain-chart:



Lift-curve:



Conclusion of the model:

In our Bank loan default case model generated by the Gradient Boosting Algorithm, All the parameters are satisfactory if we exclude the recall value which is 52% Until we can't find a much better model this model can work for us if we consider accuracy as a Priority because the stability of the model which is given by the ROC is 0.823, which can be considered a stable model.

2.7 Xtreme Gradient Boosting Algorithm:

After preprocessing of data to Xtreme Gradient Boosting Algorithm we got the following results:

For input data:

XGB Model

Default value for input data is: 1

Confusion matrix:

Actual_val	0	1
pred_val		
0.0	220	18
1.0	36	41

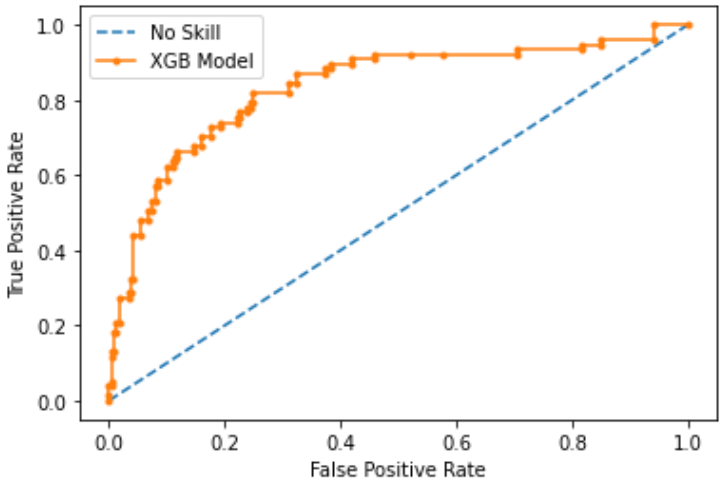
Accuracy of XGB Model = 83.0

Recall = 53.246753246753244

ROC-Curve:

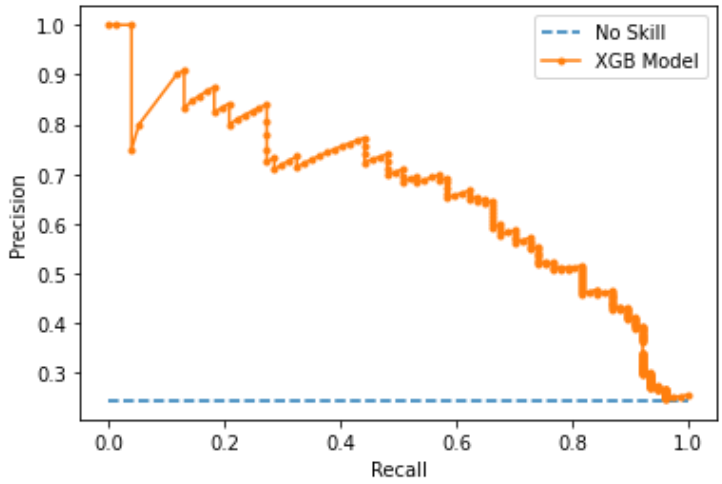
XGB Model No Skill: ROC AUC=0.500

XGB Model ROC AUC=0.836

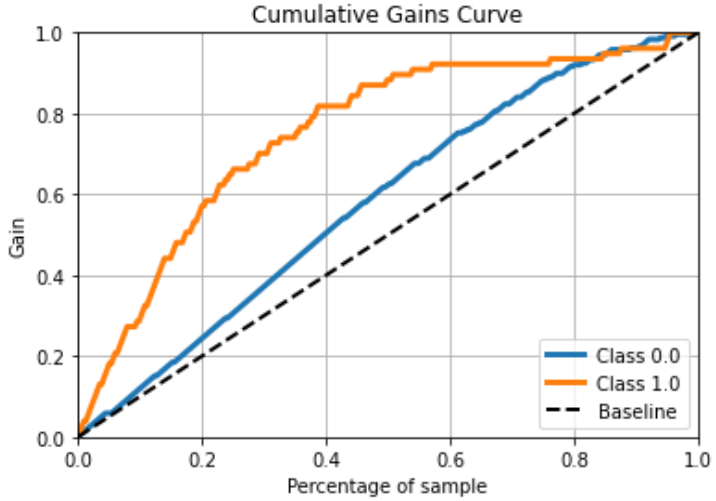


PR-Curve:

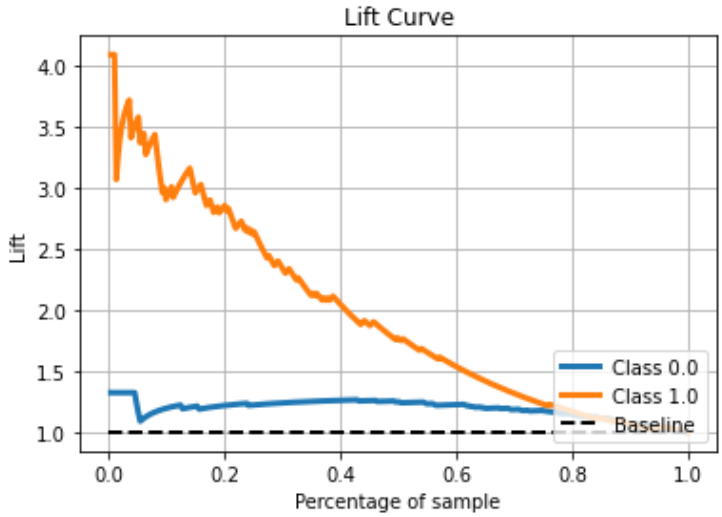
XGB Model PR AUC = 0.663



Gain-chart:



Lift-curve:



Conclusion of the model:

In our Bank loan default case model generated by the Xtreme Gradient Boosting Algorithm, All the parameters are quite standard which we can tell from the values of Recall (53.2%) and PR-AUC value (0.663), the only parameters worth consideration can be the accuracy (83%) and ROC-AUC value (0.836). Hence, since we have better models that gave much better results, we can **reject** the model.

2.8 Support Vector Machine:

After preprocessing of data to Support Vector Machine Model we got the following results:

For input data:

SVM Model

Default value for input data is: 1

Confusion matrix:

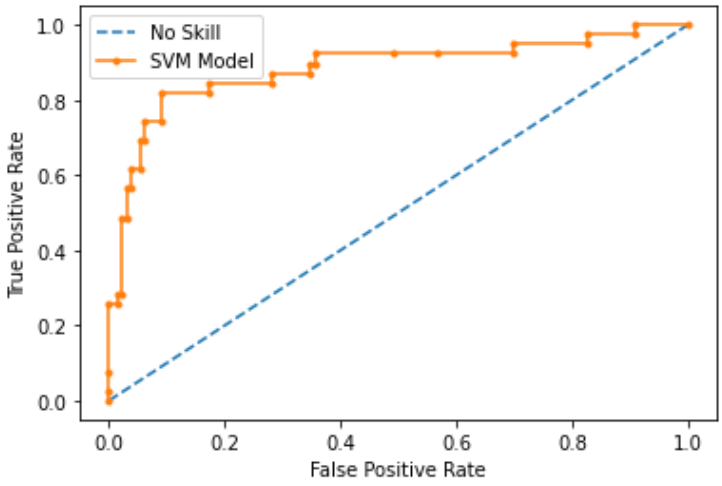
Actual_val	0	1
pred_val		
0	125	7
1	14	25

Accuracy of SVM Model 88.0

Recall: 64.1025641025641

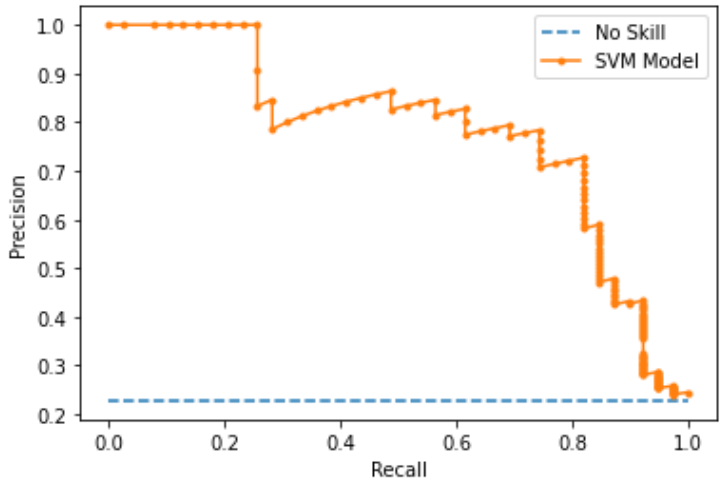
ROC-Curve:

SVM Model ROC AUC=0.884

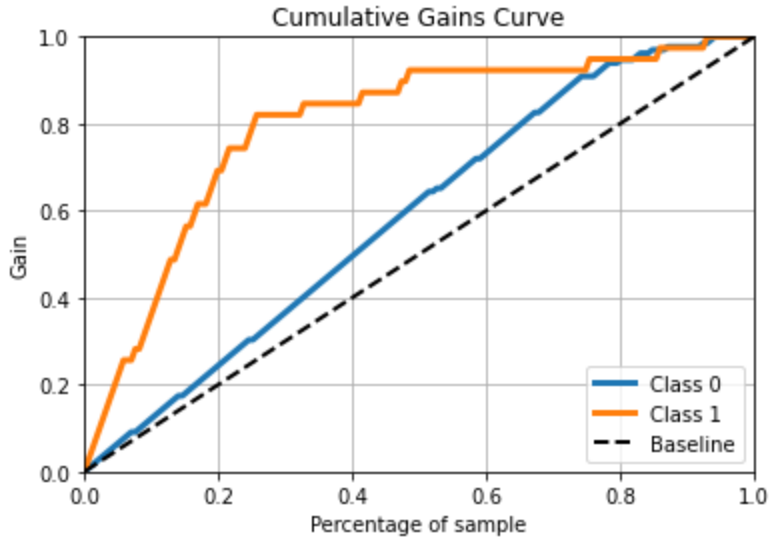


PR-Curve:

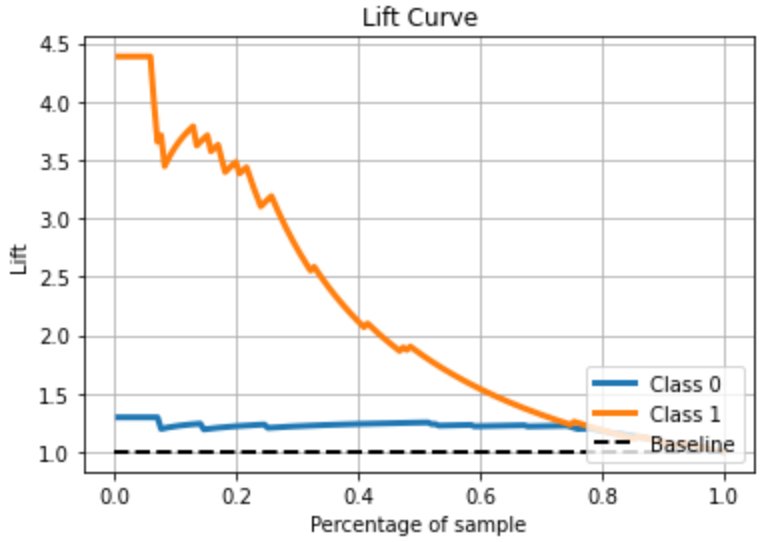
XGB Model PR AUC = 0.779



Gain-chart:



Lift-curve:



Conclusion of the model:

In our Bank loan default case model generated by the Support Vector Machine Model, all the parameters stand out when compared to the rest of the models, that at least had one or two disadvantages. But in this model, we not only have good accuracy (88%) but also the Recall is good enough to accept the model, not to mention its ROC-AUC value (0.884).

3. Conclusion -

The conclusion to the project report is that for better performance in every aspect namely:

1) Recall
2) PR-AUC value
3) ROC-AUC value
4) Accuracy

If considered as criteria, then we can accept the SVM Model as our classifying model.

----------Thank you for your time to read this comprehensive report ----------