MACHINE LEARNING AND DATA MINING PROJECT

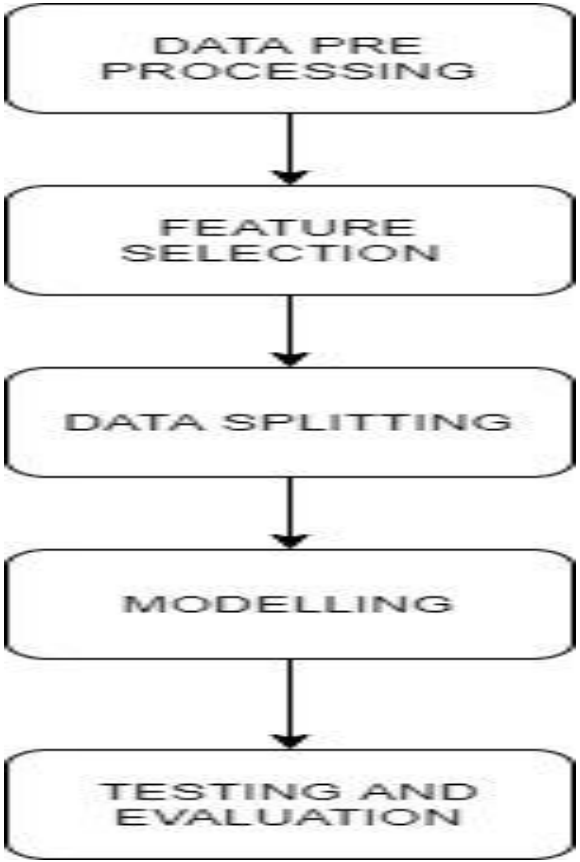
VEHICLE LOAN DEFAULT PREDICTION

REGISTRATION NUMBER	NAME
BL.EN.U4CSE17072	SARATH CHANDRA .M
BL.EN.U4CSE17076	M.M. SHOBANDARI
BL.EN.U4CSE17121	SUNIL CHOWDARY .B

ABSTRACT

- Financial institutions incur significant losses due to the default of vehicle loans. This has led to the tightening up of vehicle loan underwriting and increased vehicle loan rejection rates.
- The need for a better credit risk scoring model is also raised by these institutions. This warrants a study to estimate the determinants of vehicle loan default.
- A financial institution has hired you to accurately predict the probability of loanee/borrower defaulting on a vehicle loan in the first EMI (Equated Monthly Installments) on the due date.
- Doing so will ensure that clients capable of repayment are not rejected and important determinants can be identified which can be further used for minimising the default rates.

IMPLEMENTATION



MODELS :

- RANDOM FORESTS
- LOGISTIC REGRESSION
- DECISION TREE
- NAIVE BAYES

IMPORTED LIBRARIES

train_2_ × test_2_ × X0 × X

Source on Save

```
library(caret)
library(Boruta)
library(mlbench)
library(randomForest)
require(pROC)
library(MLmetrics)
library(e1071)
library(pscl)
library(rpart.plot)
```

SELECTING FEATURES USING BORUTA

train_2_ | test_2_ | X0 | X1 | final | Untitled4 | new | Untitled5* | Untitled6* | Untitled7* | new_1_

Source on Save | Run | Source

```r
#random forest

# getNonRejectedFormula(select)

fpart1$LOAN_DEFAULT=as.factor(fpart1$LOAN_DEFAULT)
set.seed(222)

#data splitting
ind<-sample(2,nrow(fpart1),replace=T,prob=c(0.7,0.3))
trainfin<-fpart1[ind==1,]
test<-fpart1[ind==2,]
##############################################################
#all variables
set.seed(333)
rf60<-randomForest(LOAN_DEFAULT~., data=trainfin)
rf60
p<-predict(rf60,test,type="class")
print(p)
confusionMatrix(p,test$LOAN_DEFAULT)

require(pROC)
point<-predict(rf60,test,type="prob")
ROC_rfad <- roc(test$LOAN_DEFAULT, point[,2])
ROC_rf_aucrt <- auc(ROC_rfad)
plot(ROC_rfad, col = "green", main = "ROC For Random Forest (GREEN)")
paste("Accuracy % of random forest: ", mean(test$LOAN_DEFAULT == round(point[,2], digits = 0)))
paste("Area under curve of random forest: ", ROC_rf_aucrt)
library(MLmetrics)
F1_Score(y_true=test$LOAN_DEFAULT,y_pred=p)
precision(test$LOAN_DEFAULT,p)
```

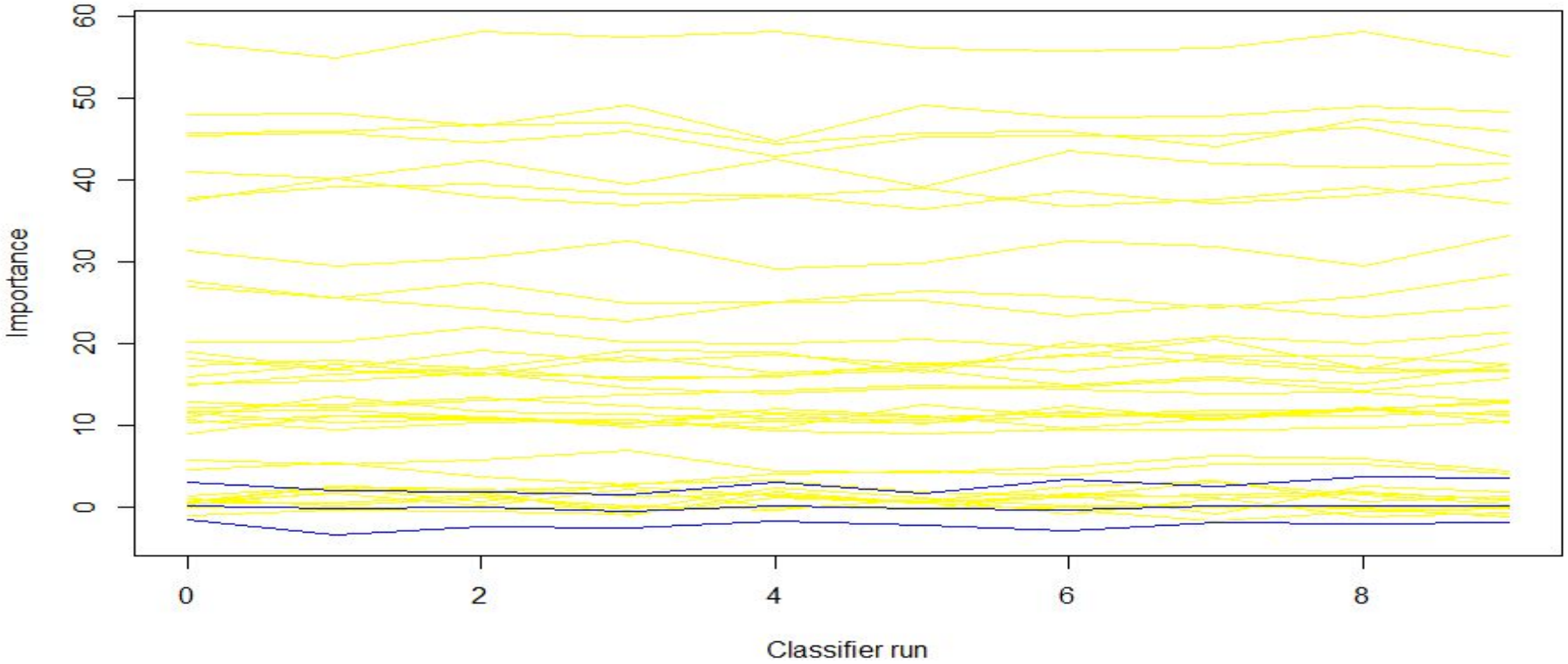
112:32 (Top Level) R Script

Console | Terminal | Jobs

M:/mldm/classprograms/

```
                          Overall
PERFORM_CNS_SCORE          100.00
PERFORM_CNS_SCORE_DESCRIPTION  98.84
LTV                         46.32
DISBURSED_AMOUNT            46.21
PRI_SANCTIONED_AMOUNT       28.24
PRI_DISBURSED_AMOUNT        27.47
PRI_CURRENT_BALANCE         24.56
STATE_ID                    22.29
EMPLOYMENT_TYPE              0.00
PAN_FLAG                     0.00
ASSET_COST                   0.00
DRIVING_FLAG                 0.00
CREDIT_HISTORY_LENGTH        0.00
CURRENT_PINCODE_ID           0.00
VOTERID_FLAG                 0.00
NO_OF_INQUIRIES              0.00
SEC_OVERDUE_ACCTS            0.00
AADHAR_FLAG                  0.00
PRI_ACTIVE_ACCTS             0.00
SEC_INSTAL_AMT               0.00
> part4<-part0[1:40000,]
> part5<-part1[1:40000,]
```

SELECTING FEATURES USING BORUTA

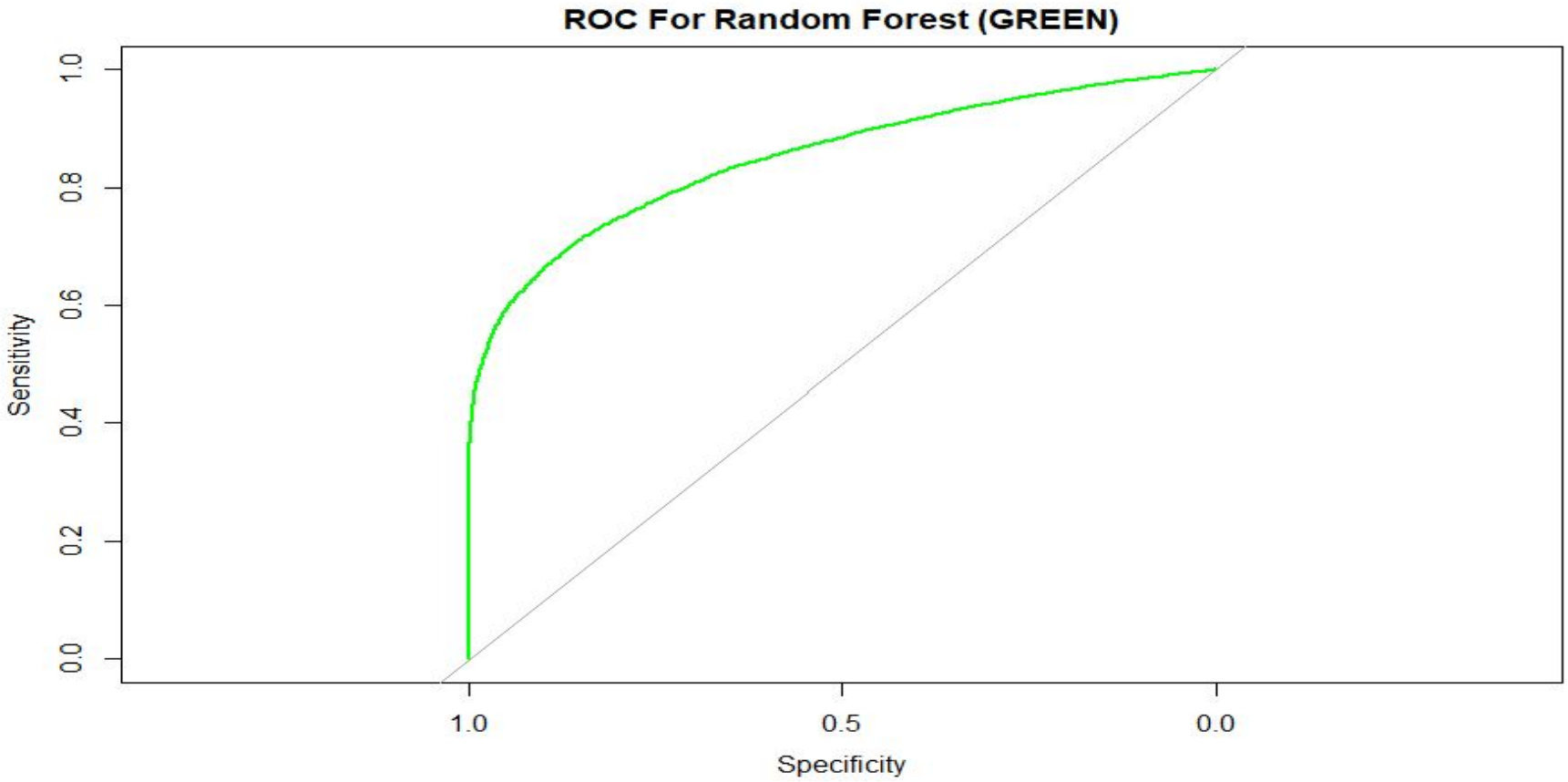


RANDOM FOREST WITH ALL FEATURES

```r
p<-predict(rf60,test,type="class")
print(p)
confusionMatrix(p,test$LOAN_DEFAULT)

require(pROC)
point<-predict(rf60,test,type="prob")
ROC_rfad <- roc(test$LOAN_DEFAULT, point[,2])
ROC_rf_aucrt <- auc(ROC_rfad)
plot(ROC_rfad, col = "green", main = "ROC For Random Forest (GREEN)")
paste("Accuracy % of random forest: ", mean(test$LOAN_DEFAULT == round(point[,2], digits = 0)))
paste("Area under curve of random forest: ", ROC_rf_aucrt)
library(MLmetrics)
F1_Score(y_true=test$LOAN_DEFAULT,y_pred=p)
precision(test$LOAN_DEFAULT,p)
recall(test$LOAN_DEFAULT,p)

#################################################################
#27 variables
rf27<-randomForest(LOAN_DEFAULT~ DISBURSED_AMOUNT + ASSET_COST + PERFORM_CNS_SCORE +
                     PRI_CURRENT_BALANCE + PRI_SANCTIONED_AMOUNT + PRI_DISBURSED_AMOUNT +
                     PRIMARY_INSTAL_AMT + LTV + BRANCH_ID + SUPPLIER_ID + MANUFACTURER_ID +
                     CURRENT_PINCODE_ID + EMPLOYMENT_TYPE + STATE_ID + EMPLOYEE_CODE_ID +
                     AADHAR_FLAG + PAN_FLAG + VOTERID_FLAG + DRIVING_FLAG + PERFORM_CNS_SCORE_DESCRIPTION +
                     PRI_NO_OF_ACCTS + PRI_ACTIVE_ACCTS + PRI_OVERDUE_ACCTS +
                     NEW_ACCTS_IN_LAST_SIX_MONTHS + AVERAGE_ACCT_AGE + CREDIT_HISTORY_LENGTH +
                     NO_OF_INQUIRIES , data=trainfin)
rf27

library(e1071)

#testing
p1<-predict(rf27,test,type="class")
```

```
M:/mldm/classprograms/
         Detection Rate : 0.4359
   Detection Prevalence : 0.5911
      Balanced Accuracy : 0.7818

       'Positive' Class : 0

> require(pROC)
> point<-predict(rf60,test,type="prob")
> ROC_rfad <- roc(test$LOAN_DEFAULT, point[,2])
Setting levels: control = 0, case = 1
Setting direction: controls < cases
> ROC_rf_aucrt <- auc(ROC_rfad)
> plot(ROC_rfad, col = "green", main = "ROC For Random Forest (GREEN)")
> paste("Accuracy % of random forest: ", mean(test$LOAN_DEFAULT == round(point[,2], digits = 0)))
[1] "Accuracy % of random forest:  0.781572809063645"
> paste("Area under curve of random forest: ", ROC_rf_aucrt)
[1] "Area under curve of random forest:  0.8514559690438 5"
> library(MLmetrics)
> F1_Score(y_true=test$LOAN_DEFAULT,y_pred=p)
[1] 0.7997096
> precision(test$LOAN_DEFAULT,p)
[1] 0.8735392
> recall(test$LOAN_DEFAULT,p)
[1] 0.7373873
>
```

RANDOM FOREST WITH ALL FEATURES

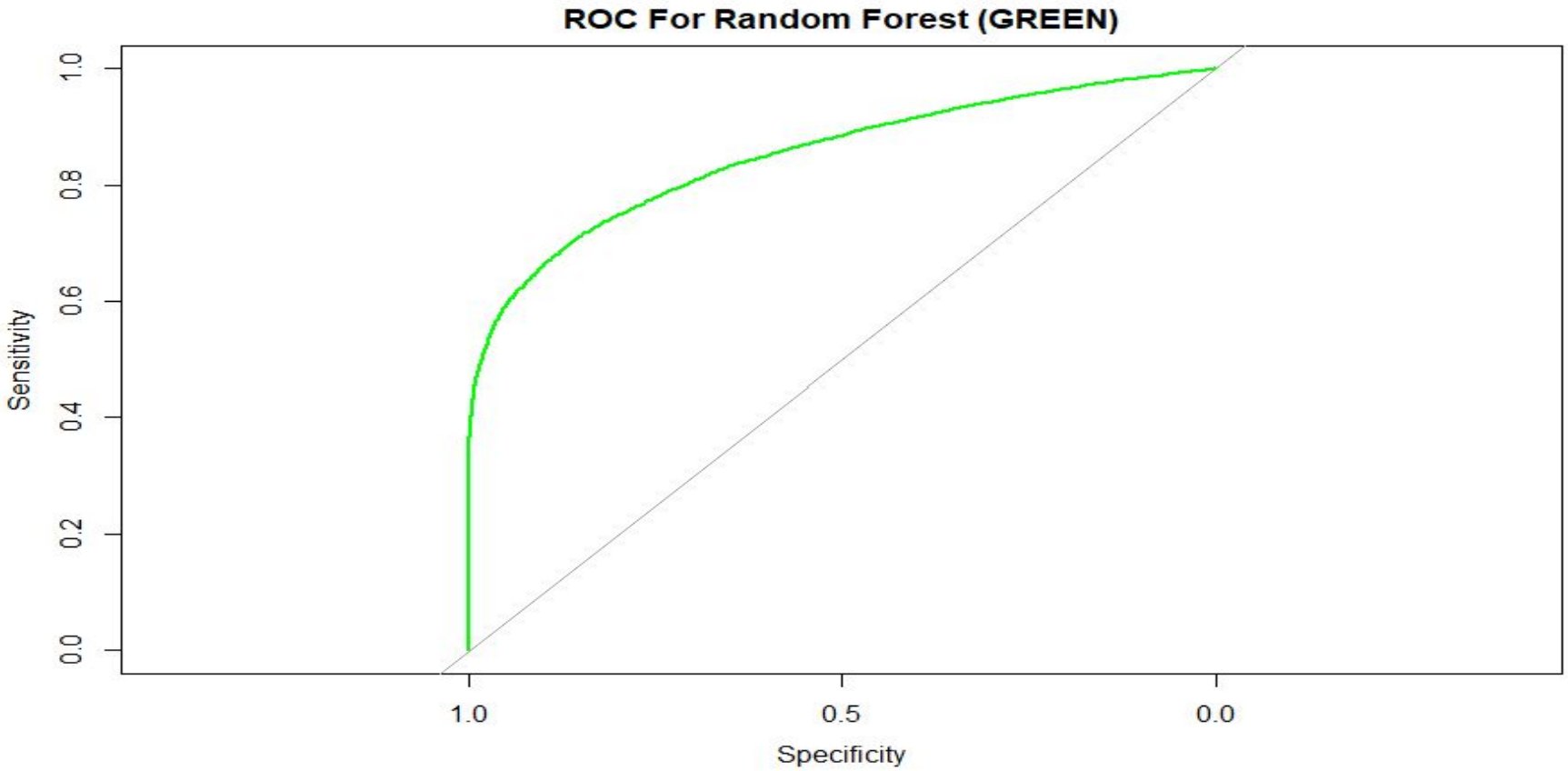


RANDOM FOREST WITH 27 SELECTED FEATURES

```
                        NO_OF_INQUIRIES , data=traintin)
rf27

library(e1071)

#testing
p1<-predict(rf27,test,type="class")
confusionmatrix(p1,test$LOAN_DEFAULT)

require(pROC)
p3<-predict(rf27,test,type="prob")
ROC_rf <- roc(test$LOAN_DEFAULT, p3[,2])
ROC_rf_auc <- auc(ROC_rf)
plot(ROC_rf, col = "green", main = "ROC For Random Forest (GREEN)")
paste("Accuracy % of random forest: ", mean(test$LOAN_DEFAULT == round(p3[,2], digits = 0)))
paste("Area under curve of random forest: ", ROC_rf_auc)
library(MLmetrics)
F1_Score(y_true=test$LOAN_DEFAULT,y_pred=p1)
precision(test$LOAN_DEFAULT,p1)
recall(test$LOAN_DEFAULT,p1)

##########################################################
##20 variables
rf20<-randomForest(LOAN_DEFAULT~PERFORM_CNS_SCORE      +
                        PERFORM_CNS_SCORE_DESCRIPTION    +
                        LTV                              +
                        DISBURSED_AMOUNT                 +
                        PRI_SANCTIONED_AMOUNT            +
                        PRI_DISBURSED_AMOUNT             +
                        PRI_CURRENT_BALANCE              +
                        STATE_ID                         +
                        EMPLOYMENT_TYPE                  +
```

```
M:/mldm/classprograms/
> confusionmatrix(p1,test$LOAN_DEFAULT)
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 10444  3618
         1  1536  8410

               Accuracy : 0.7853
                 95% CI : (0.7801, 0.7905)
    No Information Rate : 0.501
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5708

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.8718
            Specificity : 0.6992
         Pos Pred Value : 0.7427
         Neg Pred Value : 0.8456
             Prevalence : 0.4990
         Detection Rate : 0.4350
   Detection Prevalence : 0.5857
      Balanced Accuracy : 0.7855
```

RANDOM FOREST WITH 27 SELECTED FEATURES

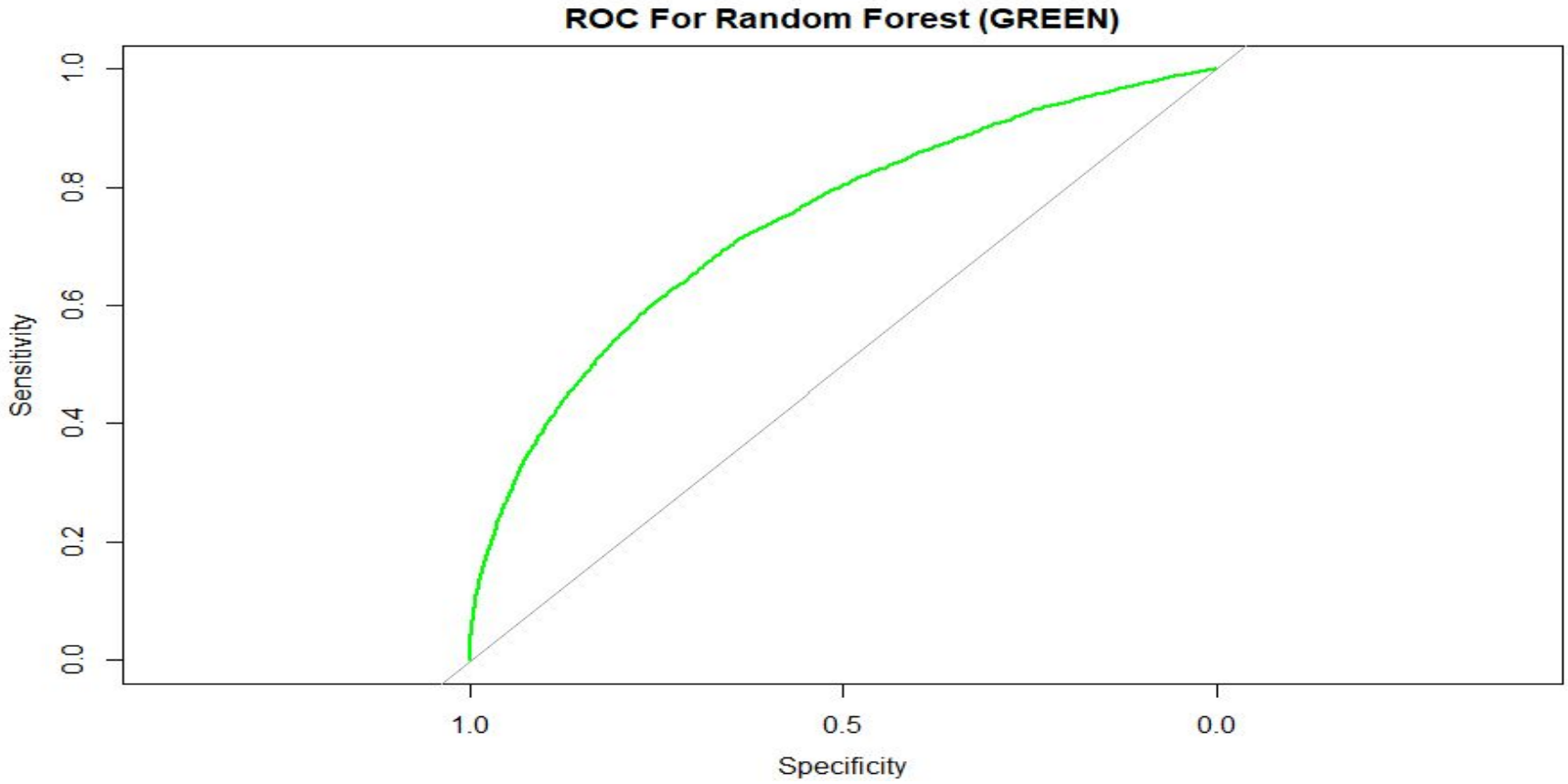


RANDOM FOREST WITH 20 SELECTED FEATURES

```
                                PRI_ACTIVE_ACCTS                    +
                                SEC_INSTAL_AMT                    , data=trainfin)
rf20

#testing
p2<-predict(rf20,test,type="class")
confusionMatrix(p2,test$LOAN_DEFAULT)

library(ROCR)

#metrics
require(pROC)
point2<-predict(rf20,test,type="prob")
ROC_rfas <- roc(test$LOAN_DEFAULT, point2[,2])
ROC_rf_aucas <- auc(ROC_rfas)
plot(ROC_rfas, col = "green", main = "ROC For Random Forest (GREEN)")
paste("Accuracy % of random forest: ", mean(test$LOAN_DEFAULT == round(point2[,2], digits = 0)))
paste("Area under curve of random forest: ", ROC_rf_aucas)
library(MLmetrics)
F1_Score(y_true=test$LOAN_DEFAULT,y_pred=p2)
precision(test$LOAN_DEFAULT,p2)
recall(test$LOAN_DEFAULT,p2)



##############################################################################


#logistic regression
```

```
> library(ROCR)
warning message:
package 'ROCR' was built under R version 4.0.3
>
> #metrics
> require(pROC)
> point2<-predict(rf20,test,type="prob")
> ROC_rfas <- roc(test$LOAN_DEFAULT, point2[,2])
Setting levels: control = 0, case = 1
Setting direction: controls < cases
> ROC_rf_aucas <- auc(ROC_rfas)
> plot(ROC_rfas, col = "green", main = "ROC For Random Forest (GREEN)")
> paste("Accuracy % of random forest: ", mean(test$LOAN_DEFAULT == round(point2[,2], digits = 0)))
[1] "Accuracy % of random forest:  0.677149283721 43"
> paste("Area under curve of random forest: ", ROC_rf_aucas)
[1] "Area under curve of random forest:  0.739740560145415"
> library(MLmetrics)
> F1_Score(y_true=test$LOAN_DEFAULT,y_pred=p2)
[1] 0.684642
> precision(test$LOAN_DEFAULT,p2)
[1] 0.7028381
> recall(test$LOAN_DEFAULT,p2)
[1] 0.6674065
>
```

RANDOM FOREST WITH 20 SELECTED FEATURES



RESULTS OF ALL IMPLEMENTATIONS

ACCURACY :

MODEL	ALL FEATURES	27 FEATURES	20 FEATURES
RANDOM FOREST	78.17	78.53	67.69
LOGISTIC REGRESSION	66.05	66.05	61.72
DECISION TREE	67.04	67.04	62.45
NAIVE BAYES	64.51	58.16	58.18

RESULTS OF ALL IMPLEMENTATIONS

AUC :

MODEL	ALL FEATURES	27 FEATURES	20 FEATURES
RANDOM FOREST	85.14	85.42	73.97
LOGISTIC REGRESSION	66.05	66.05	61.72
DECISION TREE	69.71	67.90	63.58
NAIVE BAYES	70.30	70.99	62.21

RESULTS OF ALL IMPLEMENTATIONS

F1 SCORE :

MODEL	ALL FEATURES	27 FEATURES	20 FEATURES
RANDOM FOREST	0.7997	0.8020	0.6800
LOGISTIC REGRESSION	0.6679	0.6678	0.6104
DECISION TREE	0.7305	0.7305	0.6091
NAIVE BAYES	0.6494	0.6967	0.6971

REFERENCES

- https://rpubs.com/Monesh_23992/Bank_Loan_Default - Reference model for the whole project
- https://www.analyticsvidhya.com/blog/2016/03/select-important-variables-boruta-package/ - A description for boruta feature selection
- https://towardsdatascience.com/random-forest-in-r-f66adf80ec9 - Used for implementing random forest.