Credit Scoring in R

Sovik Kumar Nath

INTRODUCTION

Reading the document

There are three components of the document in terms of reading.
(i) Plain text without boxes - In this, descriptions and analysis are written in pain text.
(ii) Code in greenish text boxes - In these, the r codes and models are written.
(iii) Text in white boxes - These shows the results from the computation of the codes.

DESCRIPTION OF THE DATASETS

The first column in the dataset has output of response variable. The output variable in this dataset corresponds to creditability or defaults. 1 corresponds to good creditability i.e. no defaults and 0 corresponds to bad creditability i.e. defaults.

The dataset is comprisd to total 20 attributes or variables, 13 of which are qualitative and 7 are numerical. The qualitative attributes have certain number of classes associated with them.

Variables - Qualitative

Attribute 1: Status of existing checking account, 4 classes
Attribute 3: Credit history, 4 classes
Attribute 4: Purpose, 11 classes
Attribute 6: Savings account/bonds, 5 classes
Attribute 7: Present employment since, 5 classes
Attribute 9: Personal status and sex, 5 classes
Attribute 10: Other debtors / guarantors, 3 classes
Attribute 12: Property, 4 classes
Attribute 14: Other installment plans, 3 classes
Attribute 15: Housing, 3 classes
Attribute 17: Job, 4 classes
Attribute 19: Telephone, 2 classes
Attribute 20: Foreign worker, 2 classes

Variables - Numerical

Attribute 2: Duration in month Attribute 5: Credit amount
Attribute 8: Installment rate in percentage of disposable income
Attribute 11: Present residence since
Attribute 13: Age in years
Attribute 16: Number of existing credits at this bank
Attribute 18: Number of people being liable to provide maintenance for

DATA PREPARATION

```
### -------------------------------READING DATA----------------------------------

data<-read.csv("german_credit.csv")
data2 <- data
hdname <- c("Creditability", "P1", "P2", "P3", "P4","P5","P6", "P7", "P8", "P9", "P10", "P11", "P12", "P
colnames(data2) <- hdname
```

For the sake of ease of handling, the variables are renamed as follows.

OUTPUT VARIABLE <- Creditability
P1 <- Account Balance
P2 <- Duration of Credit (month)
P3 <- Payment Status of Previous Credit
P4 <- Purpose
P5 <- Credit Amount
P6 <- Value Savings/Stocks
P7 <- Length of current employment
P8 <- Instalment per cent
P9 <- Sex & Marital Status
P10 <- Guarantors
P11 <- Duration in Current address
P12 <- Most valuable available asset
P13 <- Age (years)
P14 <- Concurrent Credits
P15 <- Type of apartment
P16 <- No of Credits at this Bank
P17 <- Occupation
P18 <- No of dependents
P19 <- Telephone
P20 <- Foreign Worker

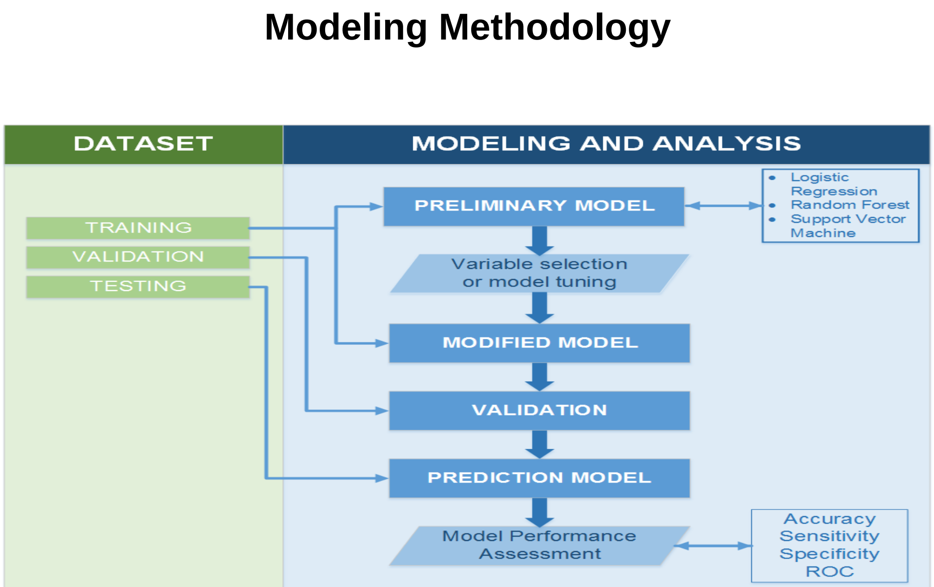
The data is splitted into three groups for the training, validating and testing the models.

```
### --------------------------------SPLITTING DATA-------------------------------
set.seed(1111)
d = sort(sample(nrow(data2), nrow(data2)*.8))
train <- data2[d,]
data3 <- data2[-d,]
d2 = sort(sample(nrow(data3), nrow(data3)*.7))
test <- data3[d2,]
cvalid <- data3[-d2,]
```

MODELING AND ANALYSIS

Three types of models are used for analysing the data and building predictive models for credit scoring, logistic regression, random forests and support vector machines. For each of the methods, the following methoodology is used. First, a preliminary model is build either to find out the significant or important variables or tuning parameters. the results of this preliminary model is analysed. Based on this analysis the new model is build using the training dataset. This model is then cross validated using the validation dataset. Once the model has been developed, it is used to predict using the test dataset. After these three stages, the performance of the models are assessed. Different performance assessment metrices are used which are accuracy, sensitivity, specificity and ROC. The area under curce (AUC) for ROC is calculated and the ROC curve is plotted. These form the four stages of indivitual modeling methods.

```r
library(png)
model_img <- readPNG("Modeling.png")
plot(c(100, 2100), c(300, 1820), type = "n", xlab = "", ylab = "", axes=FALSE, main="Modeling Methodolog
rasterImage(model_img, 100, 300, 2100, 1820, interpolate = TRUE)
```



Once the different stages of indivitual modeling methods are completed, the performance of the three methods, logistic regression, random forest and support vector machines for credit scoring, are assessed using the four performance assessment metrices.

LOGISTIC REGRESSION

```r
### ------------------------RUNNING LOGISTIC REGRESSION MODEL-----------------------

#LOGISTIC REGRESSION MODEL

m_logreg <-glm(formula = Creditability ~ ., data=train,family=binomial())

summary(m_logreg)
```

```
##
## Call:
## glm(formula = Creditability ~ ., family = binomial(), data = train)
##
```

```
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5734  -0.7588   0.4439   0.7537   1.8441
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.822e+00  1.111e+00  -3.441 0.000580 ***
## P1           5.677e-01  7.905e-02   7.181 6.92e-13 ***
## P2          -3.051e-02  9.874e-03  -3.090 0.002002 **
## P3           3.922e-01  9.904e-02   3.960 7.50e-05 ***
## P4           4.185e-02  3.368e-02   1.243 0.213973
## P5          -3.674e-05  4.484e-05  -0.819 0.412560
## P6           2.299e-01  6.510e-02   3.532 0.000413 ***
## P7           1.367e-01  8.068e-02   1.694 0.090216 .
## P8          -3.017e-01  9.277e-02  -3.253 0.001143 **
## P9           2.781e-01  1.285e-01   2.164 0.030452 *
## P10          3.998e-01  1.944e-01   2.057 0.039698 *
## P11         -4.624e-02  8.673e-02  -0.533 0.593931
## P12         -2.714e-01  1.040e-01  -2.609 0.009087 **
## P13          1.289e-02  9.026e-03   1.428 0.153258
## P14          3.080e-01  1.220e-01   2.525 0.011555 *
## P15          3.680e-01  1.850e-01   1.990 0.046635 *
## P16         -2.801e-01  1.824e-01  -1.535 0.124665
## P17          8.447e-02  1.541e-01   0.548 0.583627
## P18         -5.918e-02  2.563e-01  -0.231 0.817418
## P19          2.426e-01  2.066e-01   1.174 0.240414
## P20          5.269e-01  6.384e-01   0.825 0.409150
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 984.07  on 799  degrees of freedom
## Residual deviance: 767.61  on 779  degrees of freedom
## AIC: 809.61
##
## Number of Fisher Scoring iterations: 5
```

Variable Selection

The logistic regression model for all the variables was computed to find the significant variables based on p-values. The low p-value corresponds to the variable that contributes significantly to the model. Based on this, 10 variables were found significant, P1, P2, P3, P5, P6, P7, P8, P10, P14, P19. These corresponds to account balance, duration of credit, payment status of previous credit, credit amount, value savings/ stocks, length of current employment, instalment percent, guarantors, concurrent credits and telephone. these 10 variables were used to build the new logistic regression model for credit scoring.

```r
#MODIFIED LOGISTIC REGRESSION MODEL
m_logreg <-glm(formula = Creditability ~ P1 + P2 + P3 + P6 + P7 + P8 + P9 + P10 + P12 + P14 + P15, data=

#CROSS VALIDATION MODEL
library(boot)
cv_logreg <- cv.glm(data=train, glmfit=m_logreg, K=10)
```

```
#PREDICTION MODEL
library(ROCR)
```

```
## Loading required package: gplots
##
## Attaching package: 'gplots'
##
## The following object is masked from 'package:stats':
##
##     lowess
```

```
#score test data set
test1=test[,-1]
test_score<-predict(m_logreg,type='response',newdata=test)
```

The prediction of the logistic regression model was examined using confusion matrix, accuracy, sensitivity, specificity and ROC curve. The Area under the curve (AUC) for ROC was calculated. A higher AUC signifies a better prediction model based on test data.The calculation of these metrices are shown below.

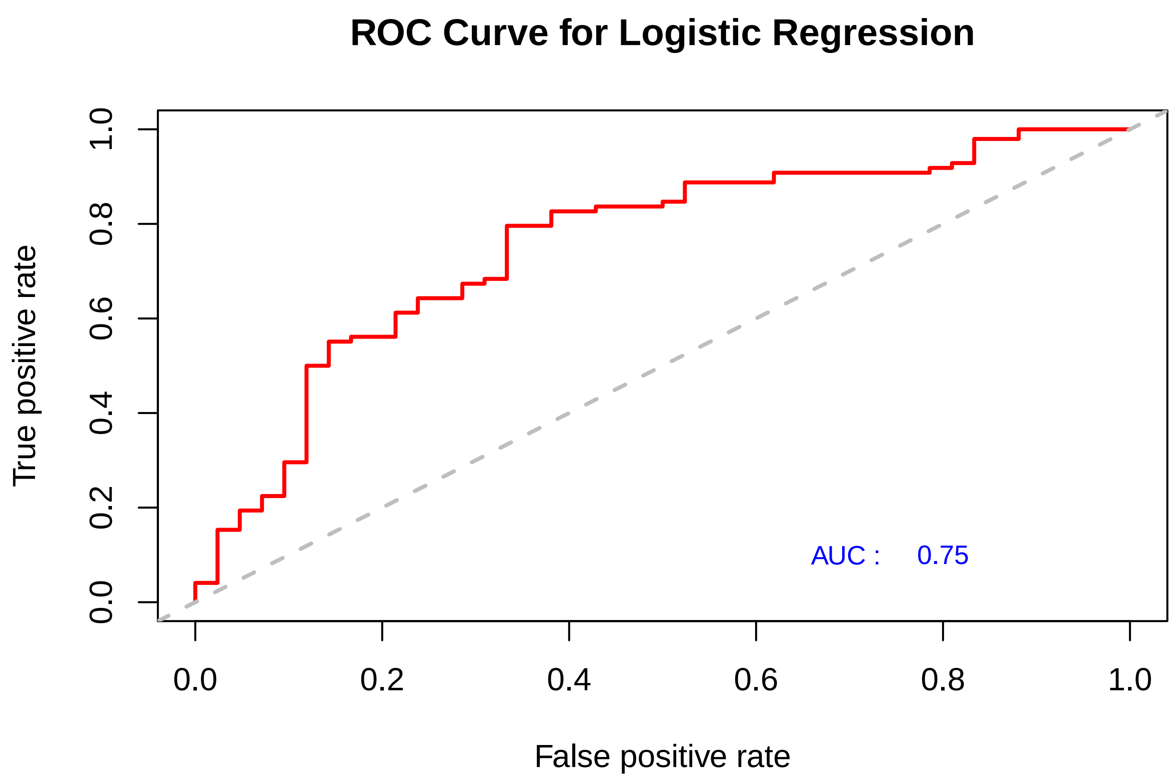
```
###--------------------------PERFORMANCE MESUREMENT---------------------------------

glm.pred=rep("0" ,nrow(test))
glm.pred[test_score>.5]="1"
tab_LR <- table(glm.pred,test[,1])
tab_LR
```

```
##
## glm.pred  0  1
##        0 23 16
##        1 19 82
```

```
acc_lr <-round((tab_LR[1,1] + tab_LR[2,2])/(tab_LR[1,1] + tab_LR[1,2] + tab_LR[2,1] + tab_LR[2,2]), dig
sen_lr <-round((tab_LR[2,2])/(tab_LR[2,1] + tab_LR[2,2]), digits=2)
spec_lr <-round((tab_LR[1,1])/(tab_LR[1,1] + tab_LR[1,2]), digits=2)

PM_lr <- data.frame(matrix(nrow = 3, ncol = 2))
PM_lr[1:3,1] <- c("Accuracy", "Sensitivity", "Specificity")
PM_lr[1:3,2] <- c(acc_lr*100, sen_lr*100, spec_lr*100)

colnames(PM_lr) <- c("Metrics", "Values")
library(knitr)
kable(PM_lr)
```

Metrics	Values
Accuracy	75
Sensitivity	81
Specificity	59

```r
pred.df <- data.frame(glm.pred,test[,1])
pred_logreg <- prediction(test_score,test[,1])
perf_logreg <- performance(pred_logreg,"tpr","fpr")
AUC_temp_logreg <- performance(pred_logreg,"auc")
AUC_logreg <- as.numeric(AUC_temp_logreg@y.values)
AUC_logreg <- round(AUC_logreg, 2)

#plot(perf_logreg,col=2,lwd=2)
plot(perf_logreg,main="ROC Curve for Logistic Regression",col=2,lwd=2)
abline(a=0,b=1,lwd=2,lty=2,col="gray")
text( x = 0.7, y = 0.1, "AUC : ", cex = 0.8, col = "blue" )
text( x = 0.8, y = 0.1, labels = AUC_logreg, cex = 0.8, col = "blue" )
```



RANDOM FOREST

```r
### ------------------------RUNNING RAMDOM FOREST MODEL--------------

#Random Forest
library(randomForest)
```

```
## randomForest 4.6-10
## Type rfNews() to see new features/changes/bug fixes.
```

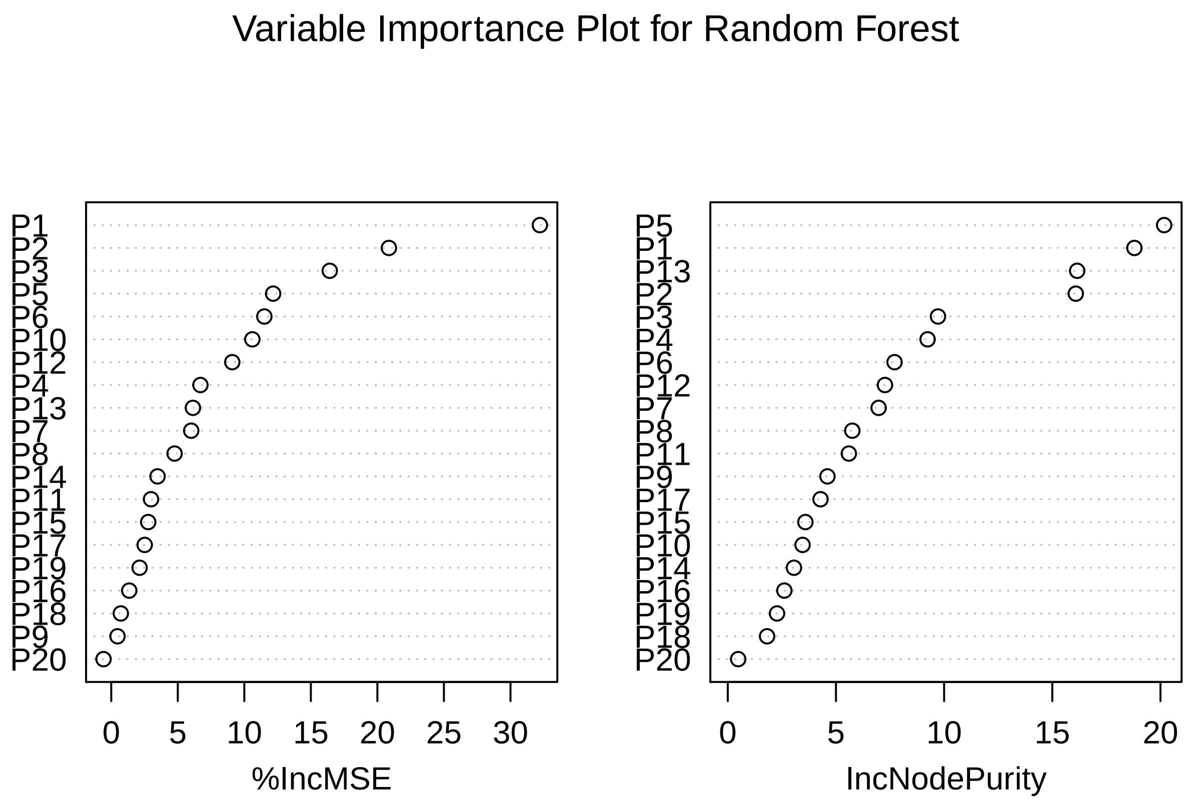
```r
m_rf <- randomForest(Creditability ~ ., data=train, importance=TRUE)
```

```
## Warning in randomForest.default(m, y, ...): The response has five or fewer
## unique values. Are you sure you want to do regression?
```

```r
round(importance(m_rf), 2)
```

```
##     %IncMSE IncNodePurity
## P1    32.21         18.80
## P2    20.86         16.09
## P3    16.42          9.72
## P4     6.70          9.24
## P5    12.17         20.17
## P6    11.50          7.71
## P7     6.01          6.97
## P8     4.76          5.76
## P9     0.47          4.61
## P10   10.60          3.46
## P11    2.99          5.60
## P12    9.09          7.27
## P13    6.14         16.15
## P14    3.48          3.06
## P15    2.78          3.59
## P16    1.35          2.61
## P17    2.51          4.29
## P18    0.72          1.82
## P19    2.13          2.28
## P20   -0.58          0.48
```

```r
varImpPlot(m_rf, main="Variable Importance Plot for Random Forest", sort=TRUE)
```



Variable Selection

Based on the variable importance values and variable importance plots for random forest, six variables were found to have importance value greater than 10%. They are P1, P2, P3, P5, P6 and P10. This corresponds to the predictors account balance, duration of credit (months), payment status of previous credit, credit amount, value savings/ stock and guarantors. These six variables were the used to construct the modified random forest model for credit scoring.

```
m_rf <- randomForest(Creditability ~ P1 + P2 + P3 + P5 + P6 + P10, data=train, importance=TRUE)
```

```
## Warning in randomForest.default(m, y, ...): The response has five or fewer
## unique values. Are you sure you want to do regression?
```

```
#PREDICTION MODEL
#score test data set
test_score_rf<-predict(m_rf,type='response',newdata=test1)
```

The prediction of the random forest model was examined using confusion matrix, accuracy, sensitivity, specificity and ROC curve. The Area under the curve (AUC) for ROC was calculated. A higherAUC signifies a better prediction model based on test data. The calculation of these metrices are shown below.

```
###--------------------------PERFORMANCE MESUREMENT---------------------------------

rf.pred=rep("0" ,nrow(test))
```

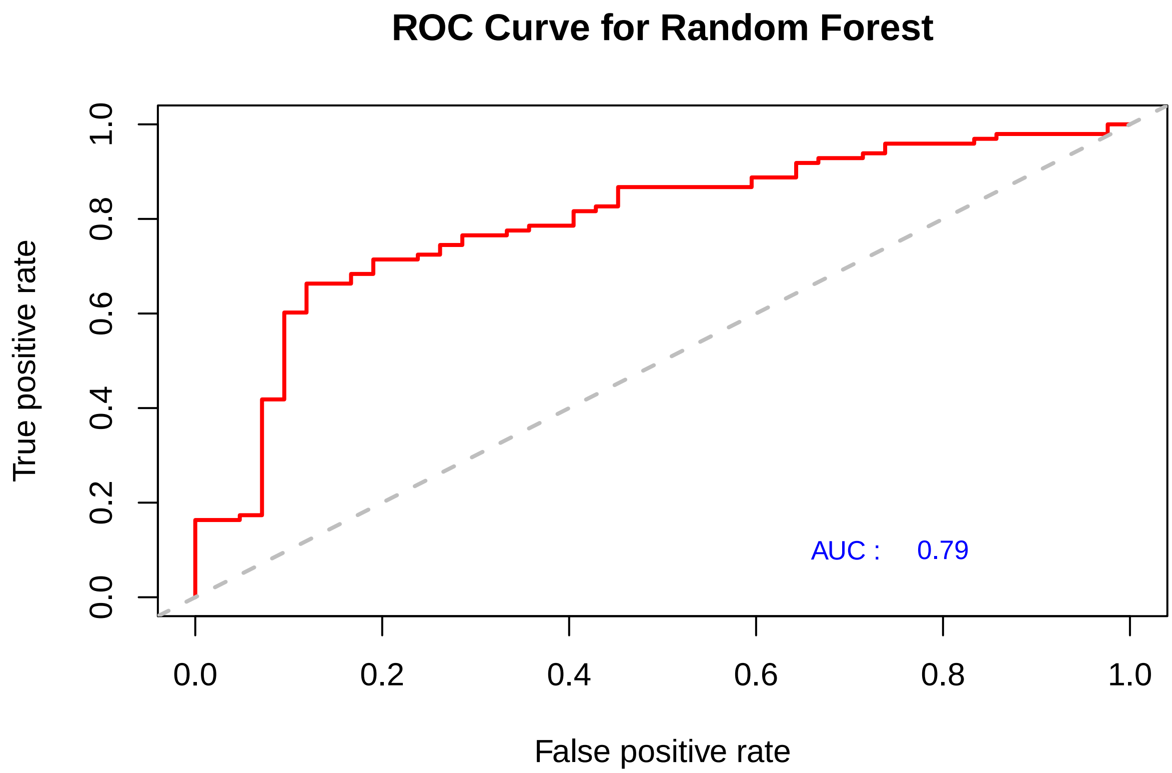
```r
rf.pred[test_score_rf>.5]="1"
tab_rf <- table(rf.pred,test[,1])
tab_rf
```

```
##
## rf.pred  0  1
##       0 22 13
##       1 20 85
```

```r
acc_rf <-round((tab_rf[1,1] + tab_rf[2,2])/(tab_rf[1,1] + tab_rf[1,2] + (tab_rf[2,1] + tab_rf[2,2])), di
sen_rf <-round((tab_rf[2,2])/(tab_rf[2,1] + tab_rf[2,2]), digits=2)
spec_rf <-round((tab_rf[1,1])/(tab_rf[1,1] + tab_rf[1,2]), digits=2)

PM_rf <- data.frame(matrix(nrow = 3, ncol = 2))
PM_rf[1:3,1] <- c("Accuracy", "Sensitivity", "Specificity")
PM_rf[1:3,2] <- c(acc_rf*100, sen_rf*100, spec_rf*100)

colnames(PM_rf) <- c("Metrics", "Values")
kable(PM_rf)
```

Metrics	Values
Accuracy	76
Sensitivity	81
Specificity	63

```r
pred.df <- data.frame(rf.pred,test[,1])
pred_rf <- prediction(test_score_rf,test[,1])
perf_rf <- performance(pred_rf,"tpr","fpr")
AUC_temp_rf <- performance(pred_rf,"auc")
AUC_rf <- as.numeric(AUC_temp_rf@y.values)
AUC_rf <- round(AUC_rf, 2)

plot(perf_rf,main="ROC Curve for Random Forest",col=2,lwd=2)
abline(a=0,b=1,lwd=2,lty=2,col="gray")
text( x = 0.7, y = 0.1, "AUC : ", cex = 0.8, col = "blue" )
text( x = 0.8, y = 0.1, labels = AUC_rf, cex = 0.8, col = "blue" )
```



SUPPORT VECTOR MACHINE

```
### -------------------------RUNNING SUPPORT VECTOR MACHINE--------------

library(e1071)
m_svm <- tune.svm(Creditability ~ ., data=train, gamma=10^(-3:-1), cost=10^(1:2))
summary(m_svm)
```

```
##
## Parameter tuning of 'svm':
##
## - sampling method: 10-fold cross validation
##
## - best parameters:
##  gamma cost
##  0.001  100
##
## - best performance: 0.1832926
##
## - Detailed performance results:
##   gamma cost     error dispersion
## 1 0.001   10 0.2269694 0.04231718
## 2 0.010   10 0.1861305 0.03410513
## 3 0.100   10 0.2027725 0.03114511
```

```
## 4 0.001  100 0.1832926 0.03404819
## 5 0.010  100 0.2317186 0.04654184
## 6 0.100  100 0.2122530 0.03118131
```

Tuning the model

The support vector machine model was tuned for the parameters, gamma and cost function. The range that was used for the parameters for this dataset is 10^{-3} and 10^{-1} for gamma and 10^{1} and 10^{2} for the cost function. 10 folds cross validation was used. Based on the results of the tuning, the optimal values for the two parameters, gamma and cost function were found as 10^{-2} and 10^{1} respectively. Using these parameters the new suport vector machine model was run and the prediction model was developed.

```
m_svm <- svm(Creditability ~ ., data=train, kernel="radial", gamma=10^-2, cost=10^1)

#PREDICTION MODEL
#score test data set
test_score_svm<-predict(m_svm, test1, decision.values = FALSE, probability = FALSE)


###--------------------------PERFORMANCE MESUREMENT---------------------------------
svm.pred=rep("0" ,nrow(test))
svm.pred[test_score_rf>.5]="1"
tab_svm <- table(svm.pred,test[,1])
tab_svm
```

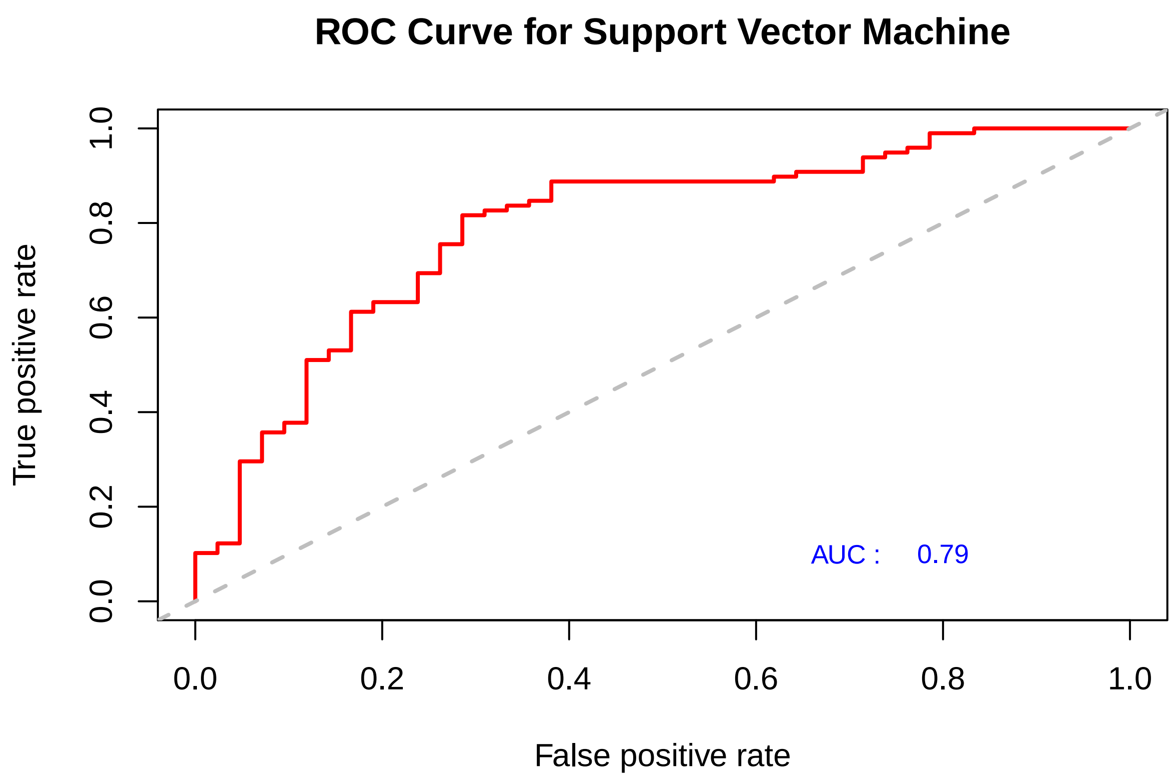
```
##
## svm.pred  0  1
##        0 22 13
##        1 20 85
```

```
acc_svm <-round((tab_svm[1,1] + tab_svm[2,2])/(tab_svm[1,1] + tab_svm[1,2] + tab_svm[2,1] + tab_svm[2,2
sen_svm <-round((tab_svm[2,2])/(tab_svm[2,1] + tab_svm[2,2]), digits=2)
spec_svm <-round((tab_svm[1,1])/(tab_svm[1,1] + tab_svm[1,2]), digits=2)

PM_svm <- data.frame(matrix(nrow = 3, ncol = 2))
PM_svm[1:3,1] <- c("Accuracy", "Sensitivity", "Specificity")
PM_svm[1:3,2] <- c(acc_svm*100, sen_svm*100, spec_svm*100)

colnames(PM_svm) <- c("Metrics", "Values")
kable(PM_svm)
```

Metrics	Values
Accuracy	76
Sensitivity	81
Specificity	63

```
pred.df <- data.frame(svm.pred,test[,1])
pred_svm <- prediction(test_score_svm,test[,1])
perf_svm <- performance(pred_svm,"tpr","fpr")
AUC_temp_svm <- performance(pred_svm,"auc")
```

```
AUC_svm <- as.numeric(AUC_temp_svm@y.values)
AUC_svm <- round(AUC_svm, 2)

plot(perf_svm, main="ROC Curve for Support Vector Machine",col=2,lwd=2)
abline(a=0,b=1,lwd=2,lty=2,col="gray")
text( x = 0.7, y = 0.1, "AUC : ", cex = 0.8, col = "blue" )
text( x = 0.8, y = 0.1, labels = AUC_svm, cex = 0.8, col = "blue" )
```



COMPARISON OF THE THREE MODELS, LOGISTIC REGRESSION, RANDOM FOREST, SUPPORT VECTOR MACHINE

The performance of the three models, logistic regression, random forest and support vector machine for credit scoring were compared using the performance assessment metrices, accuracy, sensitivity, specificity and AUC (area under curve values) of ROC curve.

```
#---------------------------Model Comparison----------------------

PM_comp <- data.frame(matrix(nrow = 4, ncol = 3))
colnames(PM_comp) <- c("Logistic Regression", "Random Forest", "Support Vector Machine")
rownames(PM_comp) <- c("Accuracy %", "Sensitivity %", "Specificity %", "ROC AUC %")
PM_comp[1:4,1] <- c(acc_lr*100, sen_lr*100, spec_lr*100, AUC_logreg*100)
PM_comp[1:4,2] <- c(acc_rf*100, sen_rf*100, spec_rf*100, AUC_rf*100)
PM_comp[1:4,3] <- c(acc_svm*100, sen_svm*100, spec_svm*100, AUC_svm*100)
```

```r
kable(PM_comp)
```

	Logistic Regression	Random Forest	Support Vector Machine
Accuracy %	75	76	76
Sensitivity %	81	81	81
Specificity %	59	63	63
ROC AUC %	75	79	79

The highest accuracy and sensitivity correspond to random forest and support vector machine while specificity is same for the three methods. The area under curve (AUC) for ROC curve is highest for logistic regression and random forest. The comparison of ROC Curve of the three methods is shown below.

```r
plot(perf_logreg, type='l', lwd=2, col="blue", main="Comparison of ROC Curve for the three methods")
lines(perf_rf@x.values[[1]], perf_rf@y.values[[1]], type='l', lwd=2, col="red")
lines(perf_svm@x.values[[1]], perf_svm@y.values[[1]], type='l', lwd=2, col="green")
abline(a=0,b=1,lwd=2,lty=2,col="gray")
legend(0.55,0.195, c("Logistic Regression","Random Forest", "Support Vector Machine"),
      lty=c(1,1,1), lwd=c(2.5,2.5, 2.5),col=c("blue","red", "green"), cex = 0.8)
```