Homework 2

Shreyas Bidwai

October 4, 2017

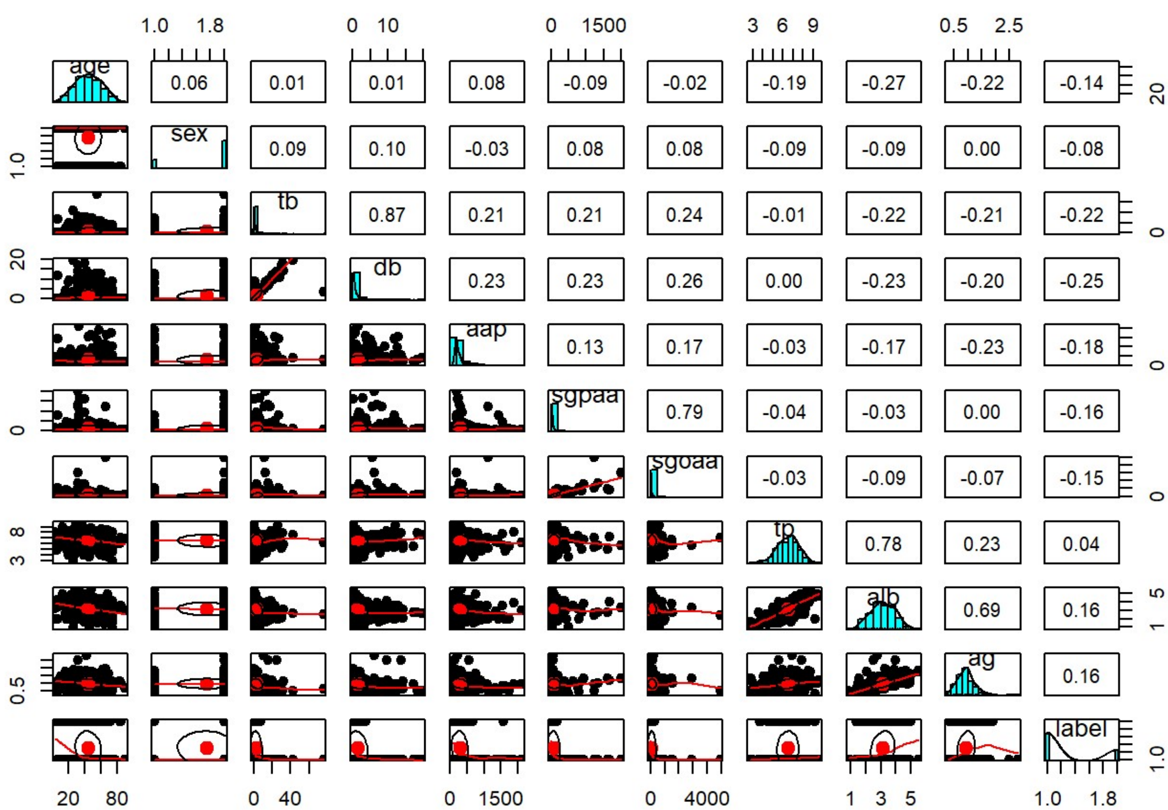
```
library(rpart)
library(caret)
library(rpart.plot)
library(ROCR)

ilpd=read.csv("/ILPD.csv", header = T, sep=",")

set.seed(100)

index<- sample(1:nrow(ilpd), size = 0.6*nrow(ilpd))
train<- ilpd[index, ]
test <- ilpd[-index, ]

library(psych)
pairs.panels(ilpd, pch=19)
```



a)

i)Strongest correlated pair :- db and tb

ii)Weakest correlated pair :- tp and db, sex and ag,sgpaa and ag

iii)Most negatively correlated :- age and alb

iv)Variables that appear to follow a Gaussian distribution :- age, tp, alb, ag

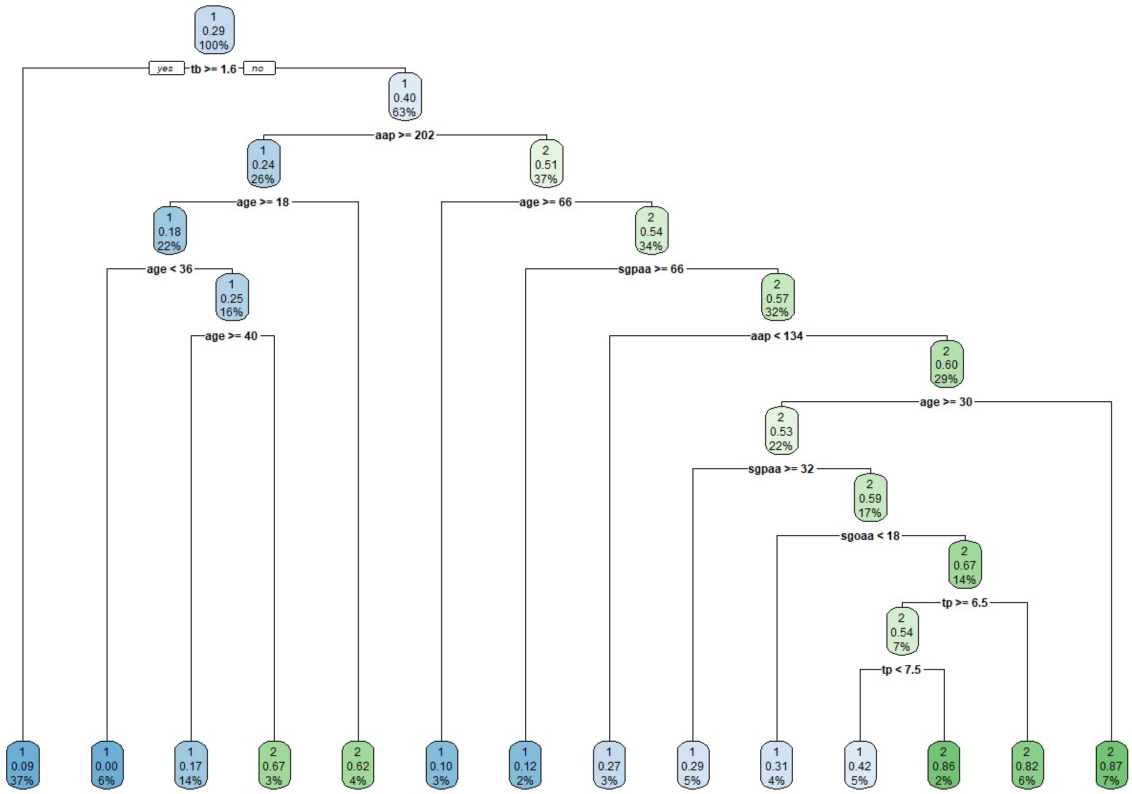
b)

Yes, I think normalising or scaling the attributes will help the classification task. Because, normalising the attributes will result in data which are similar to each other. It will also help in providing a better correlation and there is no point in normalising the data that are similar to each other.

Attributes with varied range of values that should be normalised are:- Age, tp, alb, ag

c)

```
model <- rpart(label~ ., method = "class", data =train)
rpart.plot(model)
```




```
pred <- predict(model, test, type = "class")
confusionMatrix(pred, test[,11], positive = "1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   1   2
##          1 145  51
##          2  22  16
##
##                Accuracy : 0.688
##                  95% CI : (0.6244, 0.7468)
##     No Information Rate : 0.7137
##     P-Value [Acc > NIR] : 0.826721
##
##                   Kappa : 0.123
##  Mcnemar's Test P-Value : 0.001049
##
##             Sensitivity : 0.8683
##             Specificity : 0.2388
##          Pos Pred Value : 0.7398
##          Neg Pred Value : 0.4211
##              Prevalence : 0.7137
##          Detection Rate : 0.6197
##    Detection Prevalence : 0.8376
##       Balanced Accuracy : 0.5535
##
##        'Positive' Class : 1
##
```

Accuracy: 68.8%

TPR(Sensitivity) : 0.8683

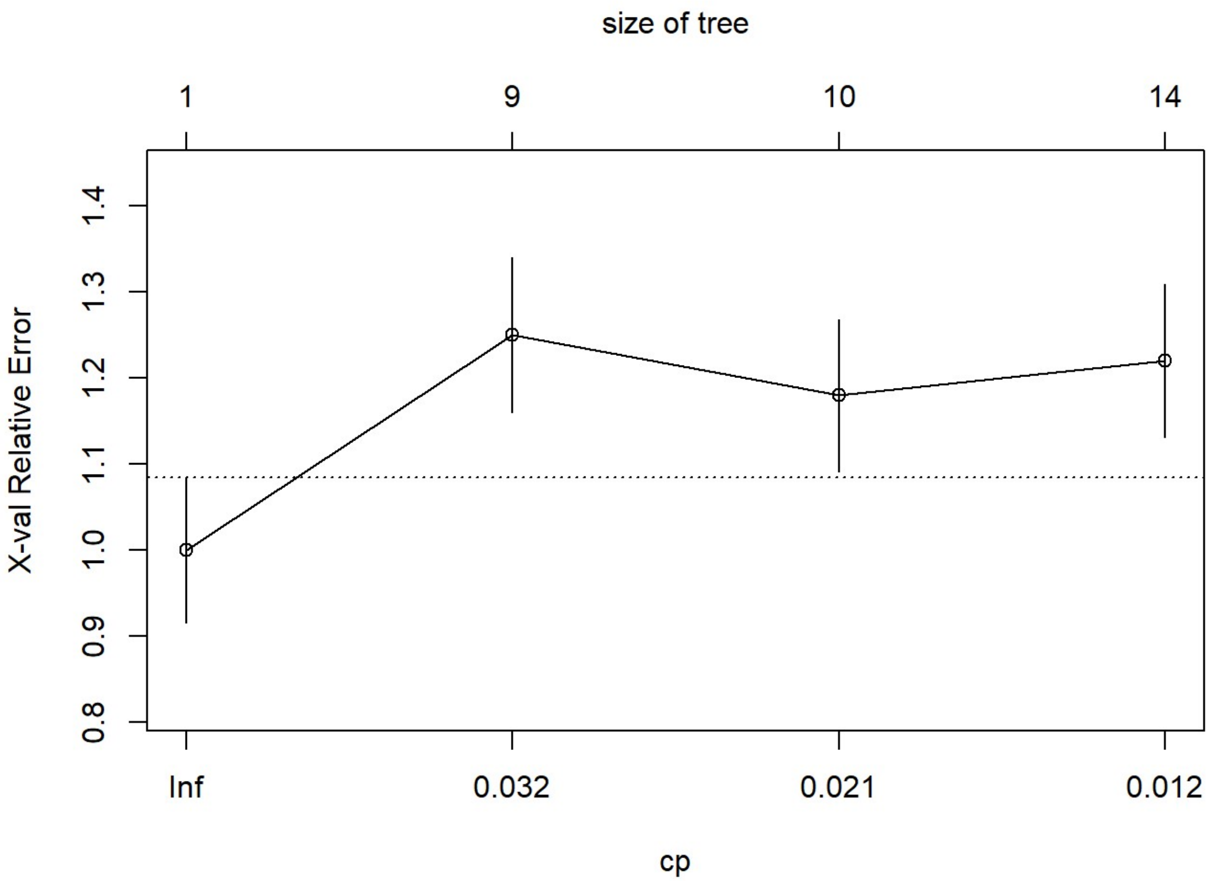
TNR(Specificity) : 0.2388

PPV(Pos Pred Value) : 0.7398

d)

-Prune

```
plotcp(model)
```



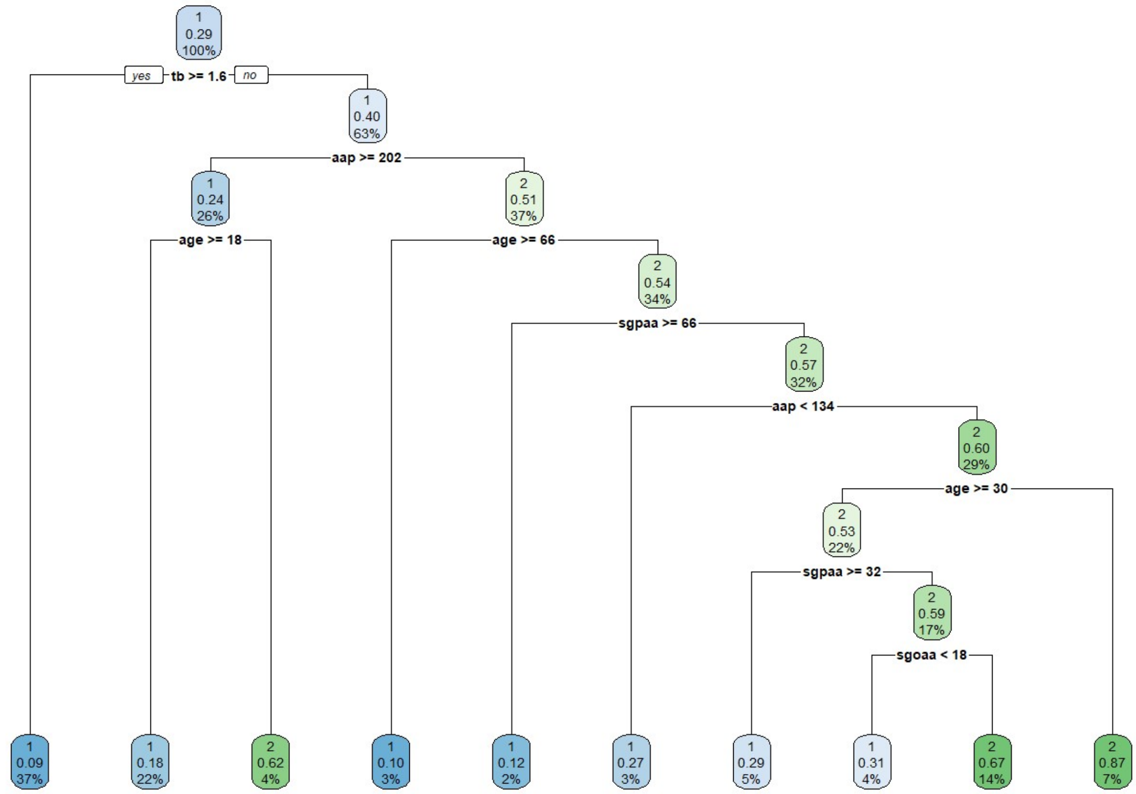

```
printcp(model)
```

```
##
## Classification tree:
## rpart(formula = label ~ ., data = train, method = "class")
##
## Variables actually used in tree construction:
## [1] aap   age   sgoaa sgpaa tb    tp
##
## Root node error: 100/349 = 0.28653
##
## n= 349
##
##         CP nsplit rel error xerror     xstd
## 1 0.033333      0      1.00   1.00 0.084467
## 2 0.030000      8      0.67   1.25 0.089571
## 3 0.015000      9      0.64   1.18 0.088376
## 4 0.010000     13      0.58   1.22 0.089080
```

```r
model.pruned <- prune(model, cp = 0.021)
pred.pruned <- predict(model.pruned, test, type = "class")
confusionMatrix(pred.pruned, test[, 11], positive = "1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   1   2
##          1 142  46
##          2  25  21
##
##                Accuracy : 0.6966
##                  95% CI : (0.6333, 0.7548)
##     No Information Rate : 0.7137
##     P-Value [Acc > NIR] : 0.74428
##
##                   Kappa : 0.1807
##  Mcnemar's Test P-Value : 0.01762
##
##             Sensitivity : 0.8503
##             Specificity : 0.3134
##          Pos Pred Value : 0.7553
##          Neg Pred Value : 0.4565
##              Prevalence : 0.7137
##          Detection Rate : 0.6068
##    Detection Prevalence : 0.8034
##       Balanced Accuracy : 0.5819
##
##        'Positive' Class : 1
##
```

```r
rpart.plot(model.pruned)
```

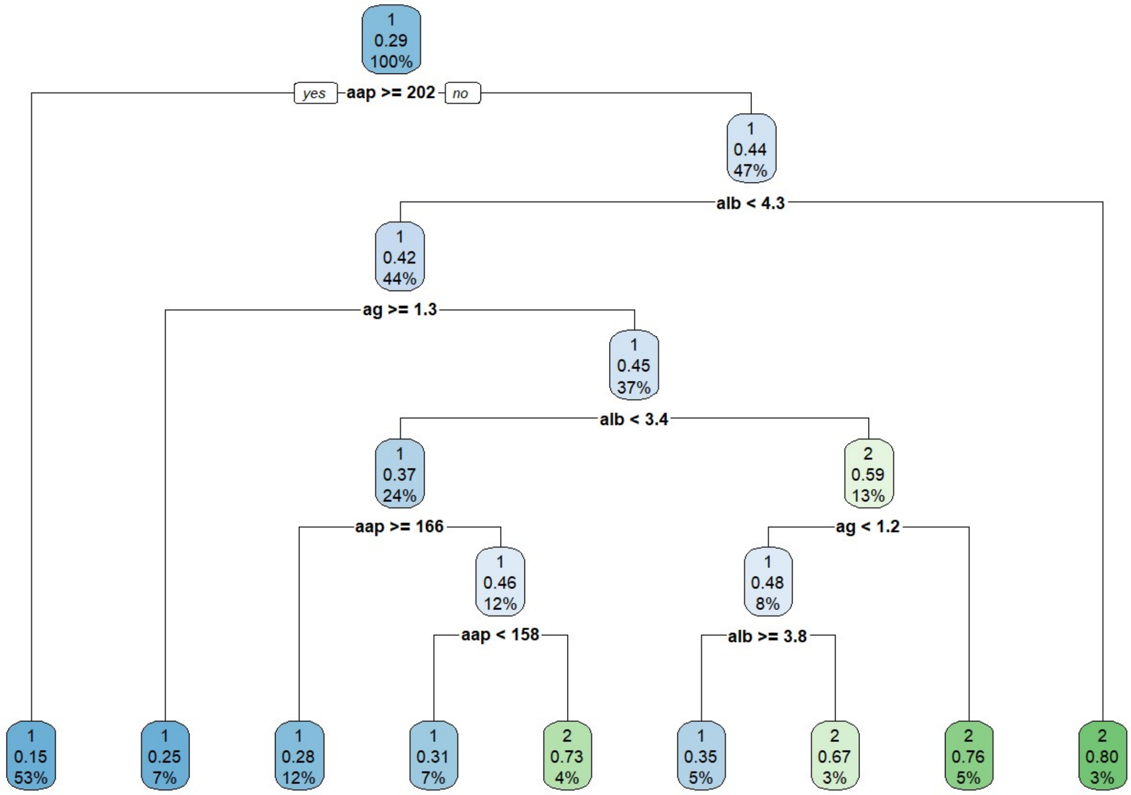


Accuracy of the model is 68.8% and after changing the values of cp I got a better accuracy of 69.66%.

Thus pruned tree has more accuracy because of lesser complexity, as their are less number of nodes to traverse in decision tree.

e)Build a new model

```
new_model<- rpart(label ~ alb+ag+aap, method = "class", data = train)
rpart.plot(new_model)
```




```
new_pred <- predict(new_model,test,type = "class")
confusionMatrix(new_pred,test[,11], positive = "1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   1   2
##          1 152  49
##          2  15  18
##
##                Accuracy : 0.7265
##                  95% CI : (0.6646, 0.7825)
##     No Information Rate : 0.7137
##     P-Value [Acc > NIR] : 0.3623
##
##                   Kappa : 0.2109
##  Mcnemar's Test P-Value : 3.707e-05
##
##             Sensitivity : 0.9102
##             Specificity : 0.2687
##          Pos Pred Value : 0.7562
##          Neg Pred Value : 0.5455
##              Prevalence : 0.7137
##          Detection Rate : 0.6496
##    Detection Prevalence : 0.8590
##       Balanced Accuracy : 0.5894
##
##        'Positive' Class : 1
##
```

```
summary(ilpd)
```

```
##       age             sex            tb                db        
##  Min.   : 4.00   Female:142   Min.   : 0.400   Min.   : 0.100  
##  1st Qu.:33.00   Male  :441   1st Qu.: 0.800   1st Qu.: 0.200  
##  Median :45.00                Median : 1.000   Median : 0.300  
##  Mean   :44.75                Mean   : 3.299   Mean   : 1.486  
##  3rd Qu.:58.00                3rd Qu.: 2.600   3rd Qu.: 1.300  
##  Max.   :90.00                Max.   :75.000   Max.   :19.700  
##       aap             sgpaa             sgoaa              tp       
##  Min.   :  63.0   Min.   :  10.00   Min.   :  10.0   Min.   :2.700  
##  1st Qu.: 175.5   1st Qu.:  23.00   1st Qu.:  25.0   1st Qu.:5.800  
##  Median : 208.0   Median :  35.00   Median :  42.0   Median :6.600  
##  Mean   : 290.6   Mean   :  80.71   Mean   : 109.9   Mean   :6.483  
##  3rd Qu.: 298.0   3rd Qu.:  60.50   3rd Qu.:  87.0   3rd Qu.:7.200  
##  Max.   :2110.0   Max.   :2000.00   Max.   :4929.0   Max.   :9.600  
##       alb              ag             label      
##  Min.   :0.900   Min.   :0.300   Min.   :1.000  
##  1st Qu.:2.600   1st Qu.:0.700   1st Qu.:1.000  
##  Median :3.100   Median :0.940   Median :1.000  
##  Mean   :3.142   Mean   :0.947   Mean   :1.286  
##  3rd Qu.:3.800   3rd Qu.:1.100   3rd Qu.:2.000  
##  Max.   :5.500   Max.   :2.800   Max.   :2.000
```

Accuracy :- 72.65%

TPR(Sensitivity) : 0.9102

TNR(Specificity) : 0.2687

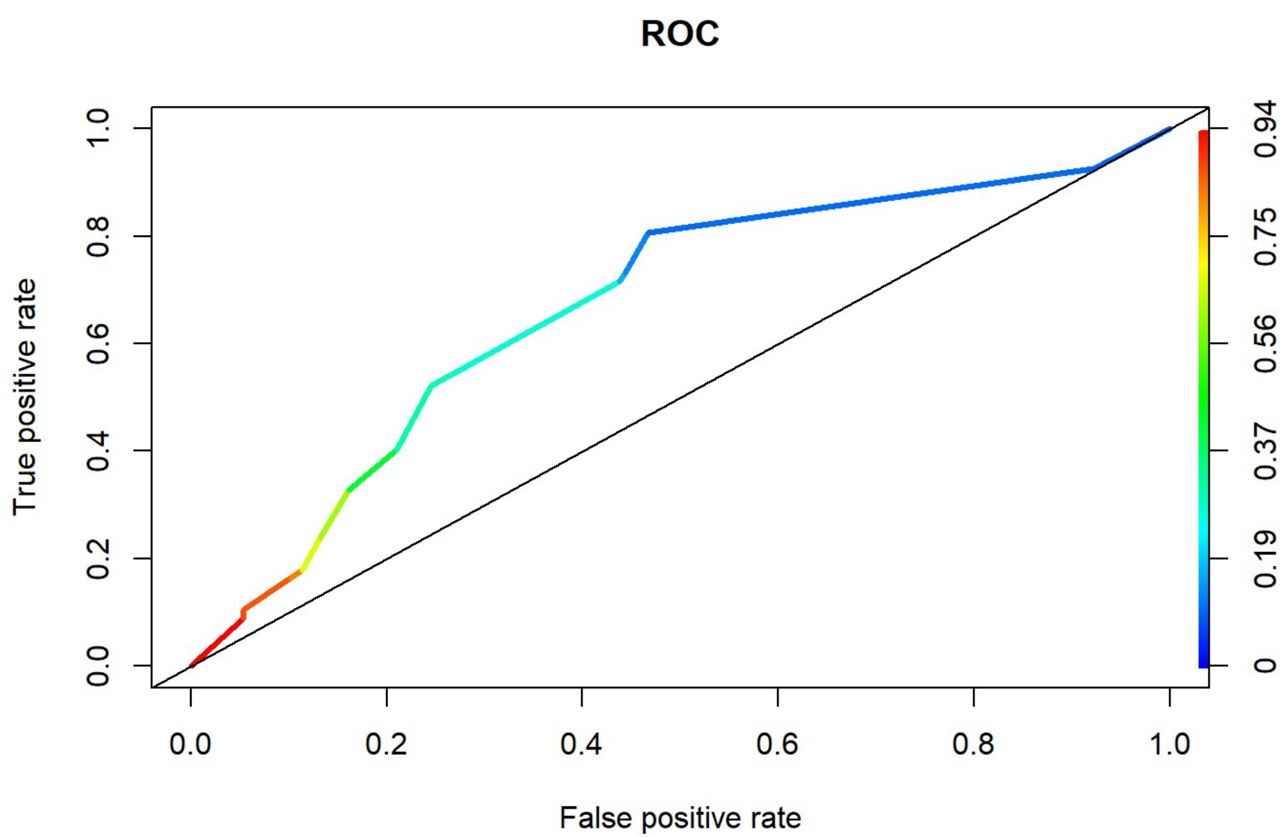
PPV(Pos Pred Value) : 0.7562

f)

(i) a ROC curve using the ROCR package.

ROC for first model

```
pred.roc <- predict(model,newdata=test,type="prob")[,2]
f.pred <- prediction(pred.roc,test$label)
f.perf <- performance(f.pred, "tpr", "fpr")
plot(f.perf, colorize=T, lwd=3, main = "ROC")
abline(0,1)
```

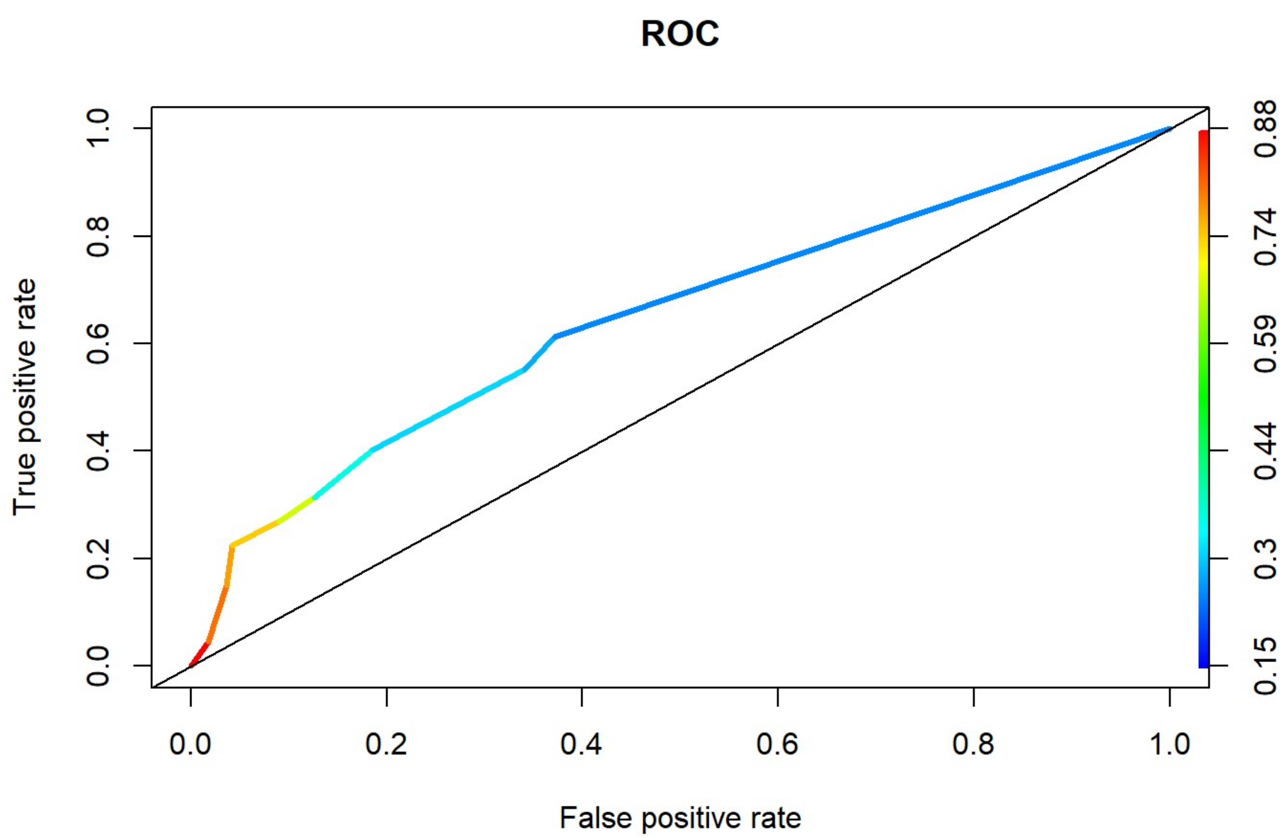



```
auc<-performance(f.pred,measure = "auc")
auc@y.values[[1]]
```

```
## [1] 0.6675753
```

ROC for new model

```
pred.roc1 <- predict(new_model,newdata=test,type="prob")[,2]
f.pred1 <- prediction(pred.roc1,test$label)
f.perf1 <- performance(f.pred1, "tpr", "fpr")
plot(f.perf1, colorize=T, lwd=3, main = "ROC")
abline(0,1)
```




```
auc<-performance(f.pred1,measure = "auc")
auc@y.values[[1]]
```

```
## [1] 0.6455
```

ii) AUC for model: 0.6675

AUC for newmodel: 0.6455

iii) Previous Model is better because the auc is closesr to 1 and greater than as compared to new model.