Project_2

Bhuvaneshwari Nattanmai Kuppusamy

3/17/2021

```
#install.packages("mlbench")
library(mlbench)
library(caret)
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
library(MASS)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v tibble  3.0.5     v dplyr   1.0.3
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0
## v purrr   0.3.4
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x purrr::lift()   masks caret::lift()
## x dplyr::select() masks MASS::select()
```

```
data("BreastCancer")
head(BreastCancer)
```

```
##        Id Cl.thickness Cell.size Cell.shape Marg.adhesion Epith.c.size
## 1 1000025            5         1          1             1            2
## 2 1002945            5         4          4             5            7
## 3 1015425            3         1          1             1            2
## 4 1016277            6         8          8             1            3
## 5 1017023            4         1          1             3            2
## 6 1017122            8        10         10             8            7
##   Bare.nuclei Bl.cromatin Normal.nucleoli Mitoses     Class
## 1           1           3               1       1    benign
## 2          10           3               2       1    benign
## 3           2           3               1       1    benign
## 4           4           3               7       1    benign
## 5           1           3               1       1    benign
## 6          10           9               7       1 malignant
```

```
summary(BreastCancer)
```

```
##       Id             Cl.thickness   Cell.size     Cell.shape  Marg.adhesion
##  Length:699         1      :145   1      :384   1      :353   1      :407
##  Class :character   5      :130   10     : 67   2      : 59   2      : 58
##  Mode  :character   3      :108   3      : 52   10     : 58   3      : 58
##                     4      : 80   2      : 45   3      : 56   10     : 55
##                     10     : 69   4      : 40   4      : 44   4      : 33
##                     2      : 50   5      : 30   5      : 34   8      : 25
##                     (Other):117   (Other): 81   (Other): 95   (Other): 63
##   Epith.c.size  Bare.nuclei   Bl.cromatin  Normal.nucleoli    Mitoses
##  2      :386   1      :402   2      :166   1      :443     1      :579
##  3      : 72   10     :132   3      :165   10     : 61     2      : 35
##  4      : 48   2      : 30   1      :152   3      : 44     3      : 33
##  1      : 47   5      : 30   7      : 73   2      : 36     10     : 14
##  6      : 41   3      : 28   4      : 40   8      : 24     4      : 12
##  5      : 39   (Other): 61   5      : 34   6      : 22     7      :  9
##  (Other): 66   NA's   : 16   (Other): 69   (Other): 69     (Other): 17
##        Class
##  benign   :458
##  malignant:241
##
##
##
##
##
```

```
str(BreastCancer)
```

```
## 'data.frame':    699 obs. of  11 variables:
##  $ Id             : chr  "1000025" "1002945" "1015425" "1016277" ...
##  $ Cl.thickness   : Ord.factor w/ 10 levels "1"<"2"<"3"<"4"<..: 5 5 3 6 4 8 1 2 2 4 ...
##  $ Cell.size      : Ord.factor w/ 10 levels "1"<"2"<"3"<"4"<..: 1 4 1 8 1 10 1 1 1 2 ...
##  $ Cell.shape     : Ord.factor w/ 10 levels "1"<"2"<"3"<"4"<..: 1 4 1 8 1 10 1 2 1 1 ...
##  $ Marg.adhesion  : Ord.factor w/ 10 levels "1"<"2"<"3"<"4"<..: 1 5 1 1 3 8 1 1 1 1 ...
##  $ Epith.c.size   : Ord.factor w/ 10 levels "1"<"2"<"3"<"4"<..: 2 7 2 3 2 7 2 2 2 2 ...
##  $ Bare.nuclei    : Factor w/ 10 levels "1","2","3","4",..: 1 10 2 4 1 10 10 1 1 1 ...
##  $ Bl.cromatin    : Factor w/ 10 levels "1","2","3","4",..: 3 3 3 3 3 9 3 3 1 2 ...
##  $ Normal.nucleoli: Factor w/ 10 levels "1","2","3","4",..: 1 2 1 7 1 7 1 1 1 1 ...
##  $ Mitoses        : Factor w/ 9 levels "1","2","3","4",..: 1 1 1 1 1 1 1 1 5 1 ...
##  $ Class          : Factor w/ 2 levels "benign","malignant": 1 1 1 1 1 2 1 1 1 1 ...
```

```
#Since Bare.nuclei has missing value,let us find the percentage of missing values to find out which met
dim(BreastCancer)
```

```
## [1] 699  11
```

```
number_rows <- nrow(BreastCancer)
number_rows
```

```
## [1] 699
```

```
na_count <-sapply(BreastCancer, function(y) (sum(length(which(is.na(y))))/number_rows)*100)
na_count
```

```
##              Id    Cl.thickness       Cell.size      Cell.shape   Marg.adhesion
##        0.000000        0.000000        0.000000        0.000000        0.000000
##    Epith.c.size     Bare.nuclei     Bl.cromatin Normal.nucleoli         Mitoses
##        0.000000        2.288984        0.000000        0.000000        0.000000
##           Class
##        0.000000
```

```
paste0("Percentage of missing values in Bare.nuclei ",round(na_count[7],2), "%")
```

```
## [1] "Percentage of missing values in Bare.nuclei 2.29%"
```

Data Description:

The BreastCancer data set has 699 observations/records, 10 predictor variables and 1 target variable. Out of the 11 predictor variables,1- Character variable,9- Nominal or ordinal variable and 1- Target class

Also, it is found that there are only 2.29% of missing values in the variable Bare.nuclei.Hence, it is better delete the rows containing missing values

```
#Deleting the rows with NA

BreastCancer.df <- na.omit(BreastCancer)

# The first variable "ID" will not make any sense in modeling phase. so,it is better remove it

BreastCancer.df$Id <- NULL


# Lets check our dataset

head(BreastCancer.df)
```

```
##   Cl.thickness Cell.size Cell.shape Marg.adhesion Epith.c.size Bare.nuclei
## 1            5         1          1             1            2           1
## 2            5         4          4             5            7          10
## 3            3         1          1             1            2           2
## 4            6         8          8             1            3           4
## 5            4         1          1             3            2           1
## 6            8        10         10             8            7          10
##   Bl.cromatin Normal.nucleoli Mitoses     Class
## 1           3               1       1    benign
## 2           3               2       1    benign
## 3           3               1       1    benign
## 4           3               7       1    benign
## 5           3               1       1    benign
## 6           9               7       1 malignant
```

```
ind <- sample(2, nrow(BreastCancer.df), replace = TRUE, prob=c(0.8, 0.2))
```

#Splitting the dataset

```r
#install.packages("caTools")
library(caTools)
set.seed(1234)
split_ratio = sample.split(BreastCancer.df, SplitRatio = 0.7)
train = subset(BreastCancer.df, split_ratio==TRUE)
test = subset(BreastCancer.df, split_ratio==FALSE)
dim(BreastCancer.df)
```

```
## [1] 683  10
```

```r
print(dim(train)); print(dim(test))
```

```
## [1] 479  10
```

```
## [1] 204  10
```

```r
names(test)[10] <- "Result"
test$Result <- as.factor(test$Result)

names(test)
```

```
##  [1] "Cl.thickness"    "Cell.size"       "Cell.shape"      "Marg.adhesion"
##  [5] "Epith.c.size"    "Bare.nuclei"     "Bl.cromatin"     "Normal.nucleoli"
##  [9] "Mitoses"         "Result"
```

```r
names(train)[10] <- "Result"
train$Result <- as.factor(train$Result)

names(train)
```

```
##  [1] "Cl.thickness"    "Cell.size"       "Cell.shape"      "Marg.adhesion"
##  [5] "Epith.c.size"    "Bare.nuclei"     "Bl.cromatin"     "Normal.nucleoli"
##  [9] "Mitoses"         "Result"
```

Create multiple models using different classifiers/algorithms

1. SVM

```r
#install.packages("e1071")
library(e1071)

# svm requires tuning
x.svm.tune <- tune(svm, Result~., data = train,
                   ranges = list(gamma = 2^(-8:1), cost = 2^(0:4)),
                   tunecontrol = tune.control(sampling = "fix"))
# display the tuning results (in text format)
x.svm.tune #note the gamma and cost
```

```
##
## Parameter tuning of 'svm':
##
## - sampling method: fixed training/validation set
##
## - best parameters:
##       gamma cost
##  0.00390625    1
##
## - best performance: 0.01875
```

```
# If the tuning results are on the margin of the parameters (e.g., gamma = 2^-8),
# then widen the parameters.
x.svm <- svm(Result~., data = train, cost=1, gamma=0.00390625   , probability = TRUE) #

x.svm.pred <- predict(x.svm, type="class", newdata=test) #ensemble; only give the class
x.svm.prob <- predict(x.svm, type="prob", newdata=test, probability = TRUE) # has to include probabilit
#t <- attr(x.svm.prob, "probabilities") # only give the probabilities
table(x.svm.pred,test$Result)
```

```
##
## x.svm.pred  benign malignant
##   benign       124         2
##   malignant      5        73
```

```
svm_accuracy <- round(((124 + 73) / nrow(test))*100,2)
paste0("The Accuracy of SVM model is ", svm_accuracy, "%")
```

```
## [1] "The Accuracy of SVM model is 96.57%"
```

2.Naive Bayes

```
#install.packages("klaR")

library(klaR)
x.nb <- naiveBayes(Result ~ ., train, laplace = 0)
x.nb.pred <- predict(x.nb,test,type="class")
x.nb.prob <- predict(x.nb,test,type="raw")
table(x.nb.pred,test$Result)
```

```
##
## x.nb.pred   benign malignant
##   benign       125         0
##   malignant      4        75
```

```
nb_accuracy <- round(((125 + 75) / nrow(test))*100,2)
paste0("The Accuracy of NB model is ", nb_accuracy, "%")
```

```
## [1] "The Accuracy of NB model is 98.04%"
```

3. Neural Network

```r
#install.packages("nnet")
library(nnet)
x.nnet <- nnet(Result ~ ., train, size=2)
```

```
## # weights:  165
## initial  value 304.026925
## iter  10 value 21.142957
## iter  20 value 17.839599
## iter  30 value 17.539058
## iter  40 value 17.450249
## iter  50 value 17.235441
## iter  60 value 16.923582
## iter  70 value 15.624469
## iter  80 value 6.755179
## iter  90 value 6.749253
## iter 100 value 6.747883
## final  value 6.747883
## stopped after 100 iterations
```

```r
x.nnet.pred <- predict(x.nnet,test,type="class")
x.nnet.prob <- predict(x.nnet,test,type="raw")
table(x.nnet.pred,test$Result)
```

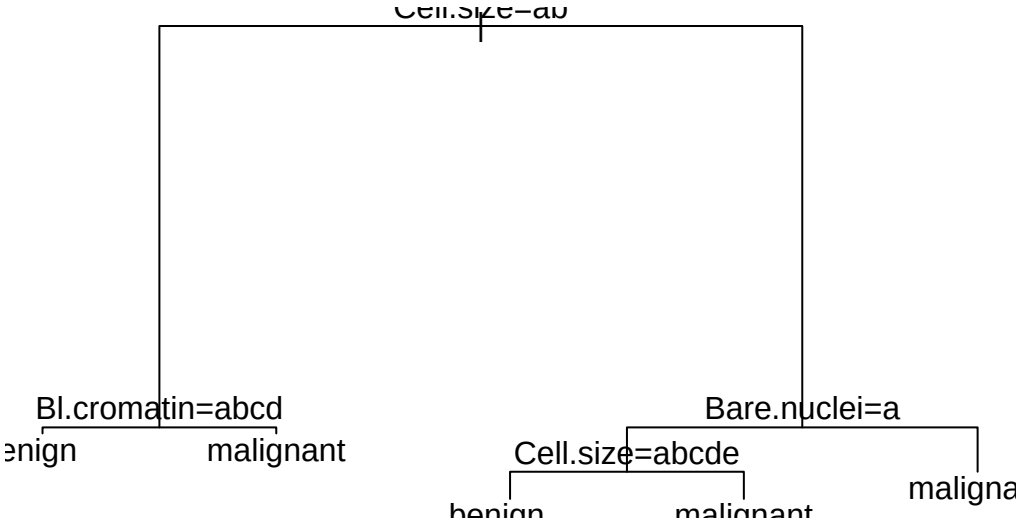
```
##
## x.nnet.pred benign malignant
##   benign       125         5
##   malignant      4        70
```

```r
neuralnet_accuracy <- round(((125 + 69) / nrow(test))*100,2)
paste0("The Accuracy of neuralnetwork model is ", neuralnet_accuracy, "%")
```

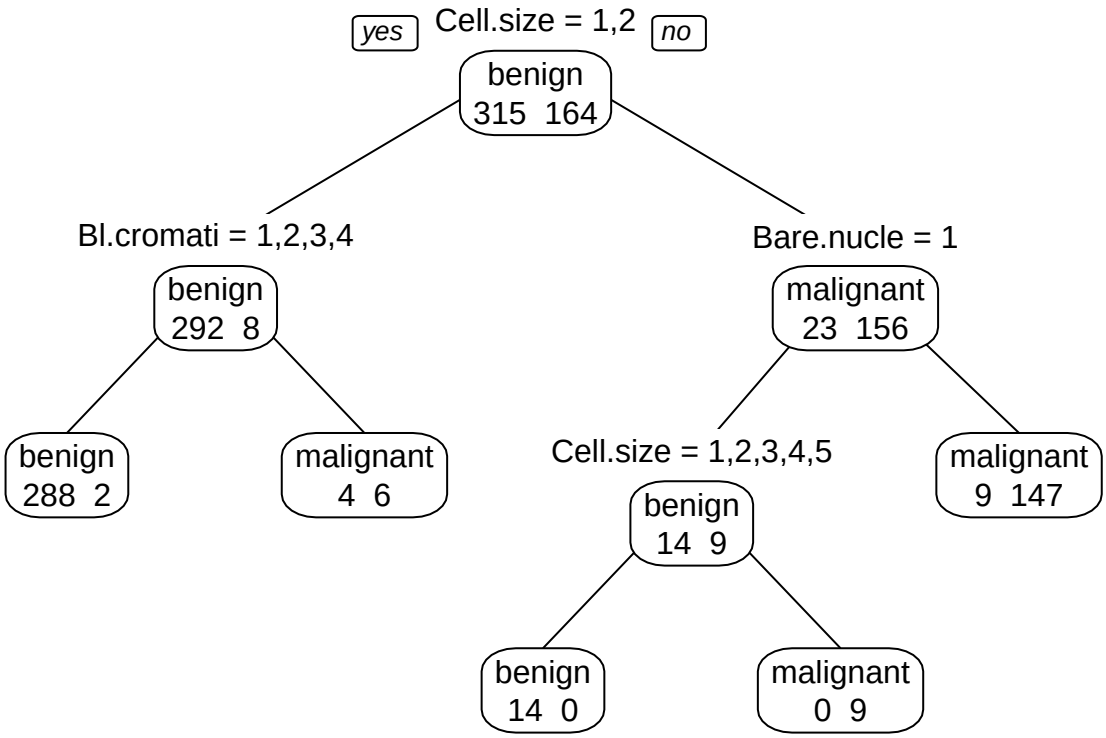
```
## [1] "The Accuracy of neuralnetwork model is 95.1%"
```

4. Decision Trees

```r
#install.packages("MASS")
library(MASS)
library(rpart)
library(rpart.plot)
x.rpart <- rpart(Result ~ ., train)
plot(x.rpart); text(x.rpart)
```




```
prp(x.rpart, type = 1, extra = 1, split.font = 1, varlen = -10)
```




```
#prediction
# predict classes for the evaluation data set
x.rpart.pred <- predict(x.rpart, type="class", newdata=test)  # to ensemble
# score the evaluation data set (extract the probabilities)
x.rpart.prob <- predict(x.rpart, type="prob", newdata=test)
table(x.rpart.pred,test$Result)
```

```
##
## x.rpart.pred benign malignant
##    benign       119         5
##    malignant     10        70
```

```r
dtaccuracy <- round(((119 +70) / nrow(test))*100,2)
paste0("The Accuracy of Decision Trees model is ", dtaccuracy, "%")
```

```
## [1] "The Accuracy of Decision Trees model is 92.65%"
```

5.conditional inference trees

```r
#install.packages("party")
library(party)
```

```
## Loading required package: grid
```

```
## Loading required package: mvtnorm
```

```
## Loading required package: modeltools
```

```
## Loading required package: stats4
```

```
## Loading required package: strucchange
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

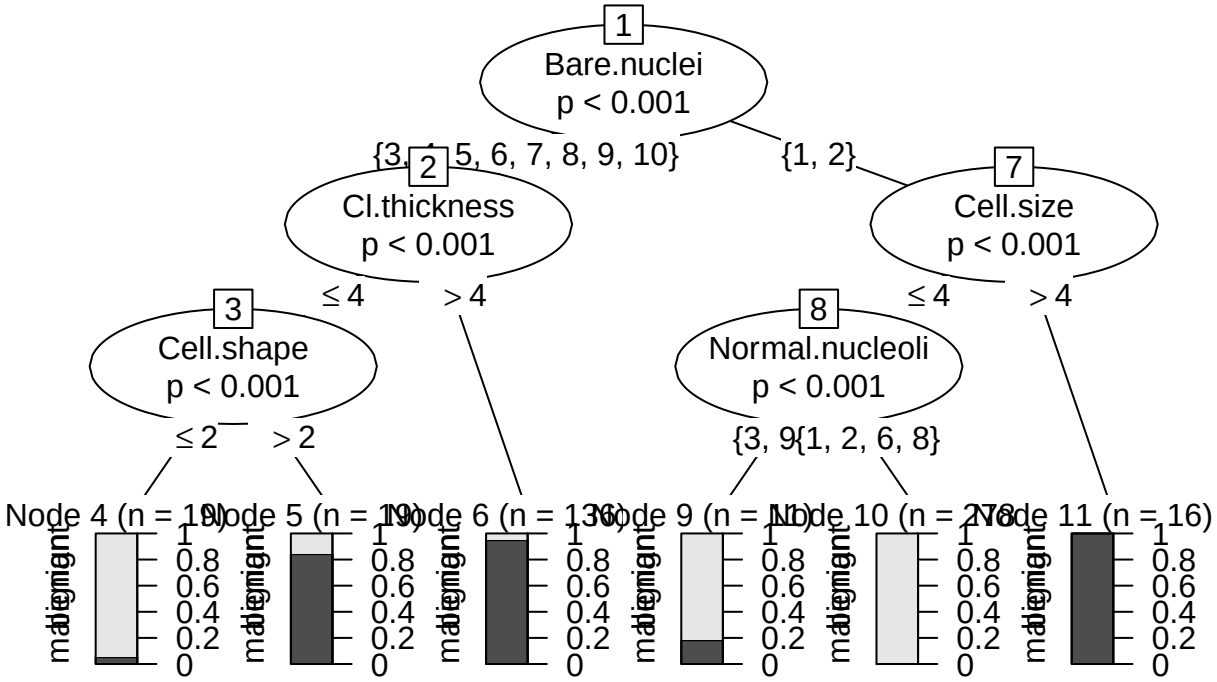
```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
## Loading required package: sandwich
```

```
##
## Attaching package: 'strucchange'
```

```
## The following object is masked from 'package:stringr':
##
##     boundary
```

```r
require(party)
x.ct <- ctree(Result ~ ., data=train)
plot(x.ct, main="Decision tree created using condition inference trees")
```




```
x.ct.pred <- predict(x.ct, newdata=test)
x.ct.prob <-  1- unlist(treeresponse(x.ct, test), use.names=F)[seq(1,nrow(test)*2,2)]
table(x.ct.pred,test$Result)
```

```
##
## x.ct.pred   benign malignant
##   benign       126         4
##   malignant      3        71
```

```
ctaccuracy <- round(((126 +71) / nrow(test))*100,2)
paste0("The Accuracy of condition inference tree model is ", ctaccuracy, "%")
```

```
## [1] "The Accuracy of condition inference tree model is 96.57%"
```

6. Random Forests

```
x.cf <- cforest(Result ~ ., train, control = cforest_unbiased(mtry = 9))

x.cf.pred <- predict(x.cf, newdata=test)
x.cf.prob <-  1- unlist(treeresponse(x.cf, test), use.names=F)[seq(1,nrow(test)*2,2)]

table(x.cf.pred, test$Result)
```

```
##
## x.cf.pred   benign malignant
##   benign       126         3
##   malignant      3        72
```

```r
rfac <- round(((129 +71) / nrow(test))*100,2)
paste0("The Accuracy of Random Forest model is ", rfac, "%")
```

```
## [1] "The Accuracy of Random Forest model is 98.04%"
```

Leave-1-Out Cross Validation (LOOCV)

```r
ans <- numeric(length(BreastCancer.df[,1]))
for (i in 1:length(BreastCancer.df[,1])) {
  rp <- rpart(Class ~ ., BreastCancer.df[-i,])
  rp.predloo <- predict(rp,BreastCancer.df[i,],type="class")
  ans[i] <- rp.predloo
  }

ans <- as.factor(ans)
ans <- factor(ans, levels=c(1,2),
  labels=c('benign','malignant'))

ans <- factor(ans,labels=levels(BreastCancer.df$Class))

cm <- confusionMatrix(ans,BreastCancer.df$Class)
acc <- cm$overall['Accuracy']

paste0("The Accuracy of LOOCV model is ", acc, "%")
```

```
## [1] "The Accuracy of LOOCV model is 0.950219619326501%"
```

bagging (bootstrap aggregating)

```r
# create model using bagging (bootstrap aggregating)
require(ipred)
```

```
## Loading required package: ipred
```

```r
x.ip <- bagging(Result ~ ., data=train)

x.ip.pred <- predict(x.ip, newdata=test)
x.ip.prob <- predict(x.ip, type="prob", newdata=test)
table(x.ip.pred,test$Result)
```

```
##
## x.ip.pred   benign malignant
##   benign       124         6
##   malignant      5        69
```

```r
bagg_accuracy <- round(((124 +68) / nrow(test))*100,2)

paste0("The Accuracy of bagging model is ", bagg_accuracy, "%")
```

```
## [1] "The Accuracy of bagging model is 94.12%"
```

Quadratic Discriminant Analysis

```
library(MASS)
library(dplyr)
train.num <- train %>% dplyr::select(-Result) %>% mutate_if(is.factor,as.character)%>% mutate_if(is.cha
train.num$Result <- train$Result
test.num <- test%>%dplyr::select(-Result) %>% mutate_if(is.factor,as.character)%>% mutate_if(is.charact
test.num$Result <- test$Result

x.qda <- qda(Result~., data = train.num) #qda, formula, right hand is non-factor
x.qda.pred <- predict(x.qda, test.num)$class
x.qda.prob <- predict(x.qda, test.num)$posterior
table(x.qda.pred,test.num$Result)
```

```
##
## x.qda.pred  benign malignant
##   benign       121         2
##   malignant      8        73
```

```
qda_accuracy <- round(((121 +73) / nrow(test))*100,2)

paste0("The Accuracy of QDA model is ", qda_accuracy, "%")
```

```
## [1] "The Accuracy of QDA model is 95.1%"
```

Regularised Discriminant Analysis

```
#not able to use test

library(klaR)
x.rda <- rda(Result~., data = train)
x.rda.pred <- predict(x.rda, test)$class
x.rda.prob <- predict(x.rda, test)$posterior
table(x.rda.pred,test$Result)
```

```
##
## x.rda.pred  benign malignant
##   benign       124         1
##   malignant      5        74
```

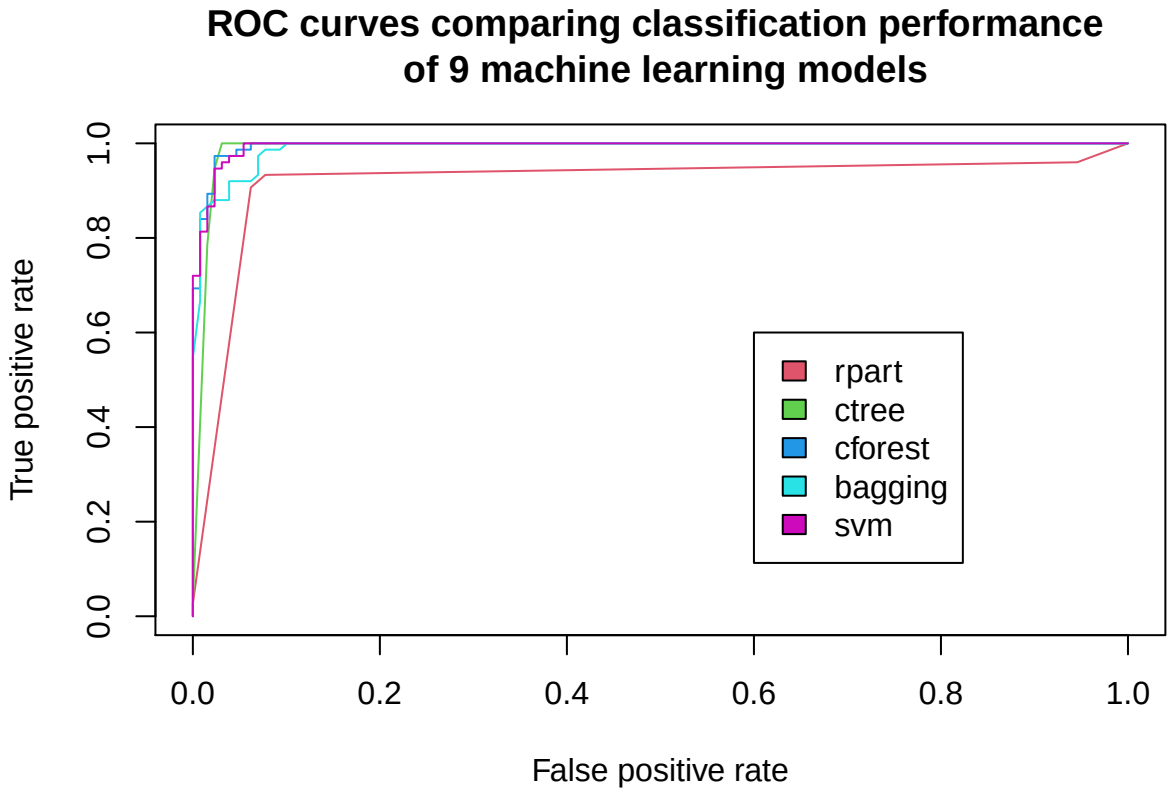
```
rda_accuracy <- round(((124 +74) / nrow(test))*100,2)

paste0("The Accuracy of RDA model is ", rda_accuracy, "%")
```

```
## [1] "The Accuracy of RDA model is 97.06%"
```

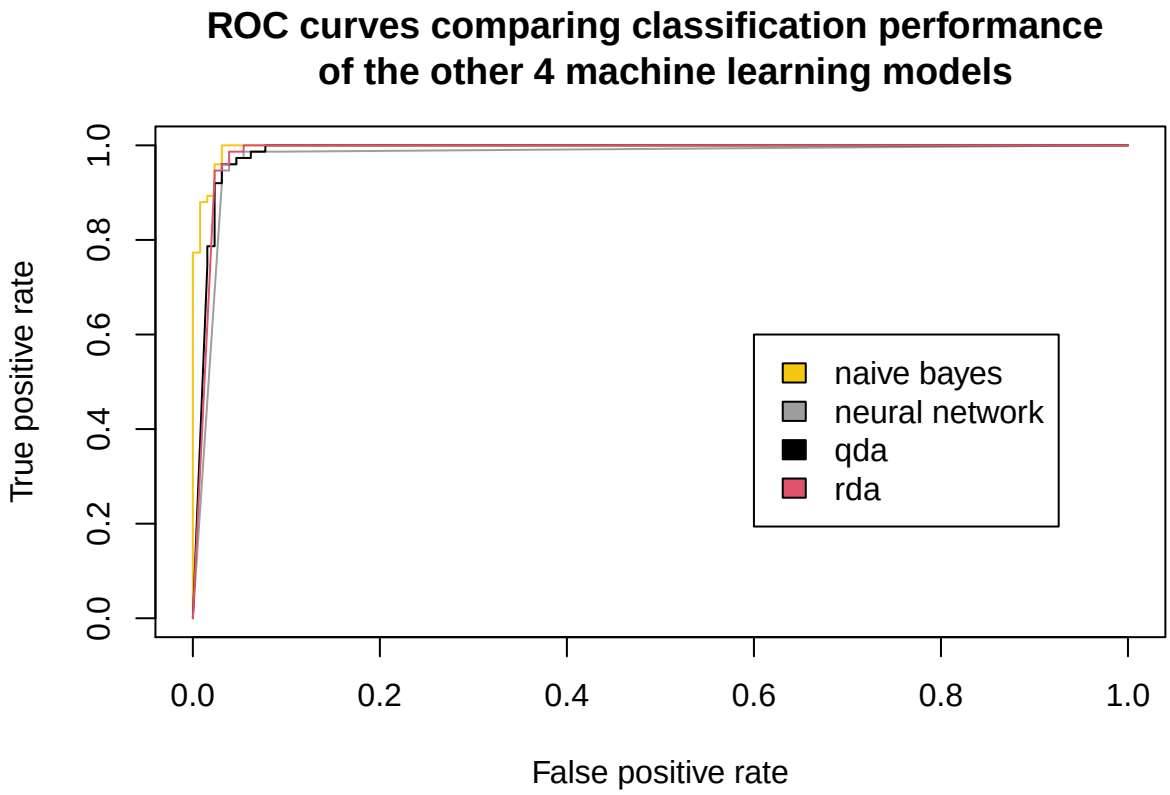
Plot ROC curves to compare the performance of the individual classifiers.

```r
#load the ROCR package which draws the ROC curves
#install.packages("ROCR")
library(ROCR)


# 1.svm
x.svm.prob.rocr <- prediction(attr(x.svm.prob, "probabilities")[,2], test[,'Result'])
x.svm.perf <- performance(x.svm.prob.rocr, "tpr","fpr")

#2.nb
x.nb.prob.rocr <- prediction(x.nb.prob[,2], test[,'Result'])
x.nb.perf <- performance(x.nb.prob.rocr, "tpr","fpr")

#3.nnet
x.nn.prob.rocr <- prediction(x.nnet.prob, test[,'Result'])
x.nn.perf <- performance(x.nn.prob.rocr, "tpr","fpr")

#4. Decision Trees
x.rpart.prob.rocr <- prediction(x.rpart.prob[,2], test[,'Result'])
x.rpart.perf <- performance(x.rpart.prob.rocr, "tpr","fpr")

#5. conditional inference trees
x.ct.prob.rocr <- prediction(x.ct.prob, test[,'Result'])
x.ct.perf <- performance(x.ct.prob.rocr, "tpr","fpr")

#6. Random Forests
x.cf.prob.rocr <- prediction(x.cf.prob, test[,'Result'])
x.cf.perf <- performance(x.cf.prob.rocr, "tpr","fpr")

#7. bagging
x.ip.prob.rocr <- prediction(x.ip.prob[,2], test[,'Result'])
x.ip.perf <- performance(x.ip.prob.rocr, "tpr","fpr")

# 8.qda
x.qda.prob.rocr <- prediction(x.qda.prob[,2], test[,'Result'])
x.qda.perf <- performance(x.qda.prob.rocr, "tpr","fpr")

# 9.rda
x.rda.prob.rocr <- prediction(x.rda.prob[,2], test[,'Result'])
x.rda.perf <- performance(x.rda.prob.rocr, "tpr","fpr")
```

```r
####### plot
# Output the plot to a PNG file for display on web.  To draw to the screen,
# comment this line out.
#png(filename="roc_curve_models1.png", width=700, height=700)

#par(mfrow=c(1,2))
plot(x.rpart.perf, col=2, main="ROC curves comparing classification performance \n of 9 machine learning
legend(0.6, 0.6, c('rpart', 'ctree', 'cforest','bagging','svm'), 2:6)# Draw a legend.
plot(x.ct.perf, col=3, add=TRUE)# add=TRUE draws on the existing chart  #has to be run together.
plot(x.cf.perf, col=4, add=TRUE)
plot(x.ip.perf, col=5, add=TRUE)
plot(x.svm.perf, col=6, add=TRUE)
```




```
# Close and save the PNG file.
#dev.off()

#png(filename="roc_curve_models2.png", width=700, height=700)
plot(x.nb.perf, col=7, main="ROC curves comparing classification performance \n of the other 4 machine l
legend(0.6, 0.6, c('naive bayes', 'neural network', 'qda','rda'), 7:10)
plot(x.nn.perf, col=8, add=TRUE)
plot(x.qda.perf, col=9, add=TRUE)
plot(x.rda.perf, col=10, add=TRUE)
```




```
#dev.off()
```

Let us use "majority rule" ensemble approach by stacking the previous algorithms svm, naive bayes, neural network, decision tree,Leave-1-Out Cross Validation, Regularised Discriminant Analysis and random forest. The overall accuracy of the ensemble model is 98.04%

```
combinedf <- data.frame(x.svm.pred,x.nb.pred,x.nnet.pred,x.rpart.pred,x.cf.pred,Class = test$Result, st


stvm <- svm(Class ~ ., combinedf)
stvm.pred <- predict(stvm, test)

table(stvm.pred,test$Result)
```

```
##
## stvm.pred   benign malignant
##   benign       125         0
##   malignant      4        75
```

```
##
## stvm.pred   benign malignant
##   benign       125         0
##   malignant      4        75
stack_accuracy <- round(((125 +75) / nrow(test))*100,2)

accuracy_df <- rbind("SVM Accuracy" = svm_accuracy, "Naive Bayes Accuracy" = nb_accuracy, "Neural Netwo
accuracy_df
```

```
##                              Accuracy
## SVM Accuracy                96.5700000
## Naive Bayes Accuracy        98.0400000
## Neural Network Accuracy     95.1000000
## Decision Tree Accuracy      92.6500000
## LOOCV Accuracy               0.9502196
## RDA Accuracy                97.0600000
## Random Forest Accuracy      98.0400000
## Majority Ensemble Accuracy 98.0400000
```

```
paste0("Therefore the overall ensemble majority model accuracy is ",stack_accuracy,"%")
```

```
## [1] "Therefore the overall ensemble majority model accuracy is 98.04%"
```