TBANLT560\_ProjectTwo

**Toward a Data Mining Portfolio**

#To start, Load the BreastCancer dataset

#load the mlbench package which has the BreastCancer data set  
require(mlbench)

## Loading required package: mlbench

## Warning: package 'mlbench' was built under R version 4.0.4

# if you don't have any required package, use the install.packages() command  
# load the data set from RStudio  
data(BreastCancer)

#Best practices, time to Clean the Dataset and Parition the Data

#summarize the data so we can see what variables may need to be cleaned  
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   
##   
##   
##   
##   
##

#Drop any NAs to clean the dataset using na.omit. This prevents future failures in the model that cannot process NAs.  
BreastCancer <- na.omit(BreastCancer)  
  
#ID is a categorical variable that is not used to predict relationships in our mdoels. Therefore, we will make the entire column NULL to maintain intergrity but not break the mdoels.  
BreastCancer$Id <- NULL  
  
##Partition the data into 80% training and 20% validation sets. This will allow us to test and validate our prediction methods for accuracy.  
  
set.seed(1)  
train.index <- sample(row.names(BreastCancer), 0.6\*dim(BreastCancer)[1])   
valid.index <- setdiff(row.names(BreastCancer), train.index)   
train.df <- BreastCancer[train.index, ]  
valid.df <- BreastCancer[valid.index, ]

##First model- Support Vector Machine

#install and library the package needed for Support Vector Machine Model  
library(e1071)

## Warning: package 'e1071' was built under R version 4.0.3

#Using the training data sets- build the SVM model. You can then use that SVM model to predict the accuracy when applying it to the validation dataset.   
mysvm <- svm(Class ~ ., train.df)  
mysvm.pred <- predict(mysvm, valid.df)  
  
#visualize the model in a table to see the confusion matrix of accurately predicted values.  
table(mysvm.pred,valid.df$Class)

##   
## mysvm.pred benign malignant  
## benign 178 4  
## malignant 7 85

#Now let’s run our Second Model- NaiveBayes Model

#Install and library the necessary package to run a NaiveBayes Model   
library(klaR)

## Warning: package 'klaR' was built under R version 4.0.4

## Loading required package: MASS

#Using the training data sets- build the NB model. You can then use that NB model to predict the accuracy when applying it to the validation dataset.   
mynb <- NaiveBayes(Class ~ ., train.df)  
mynb.pred <- predict(mynb,valid.df)

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 1

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 2

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 7

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 15

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 18

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 19

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 21

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 26

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 28

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 30

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 36

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 49

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 55

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 60

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 66

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 68

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 72

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 76

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 77

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 79

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 81

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 83

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 85

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 88

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 89

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 90

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 92

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 98

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 103

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 104

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 107

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 109

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 111

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 115

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 117

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 120

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 121

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 124

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 128

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 132

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 134

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 139

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 142

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 146

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 148

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 153

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 161

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 165

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 183

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 189

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 190

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 197

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 198

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 206

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 209

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 218

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 219

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 226

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 235

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 244

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 246

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 250

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 257

#visualize the model in a table to see the confusion matrix of accurately predicted values.  
table(mynb.pred$class,valid.df$Class)

##   
## benign malignant  
## benign 179 1  
## malignant 6 88

#Thrid Model- NeurualNet

#Install and library the necessary package to run a NeuralNet Model   
library(nnet)  
  
#Using the training data sets- build the NN model. You can then use that NN model to predict the accuracy when applying it to the validation dataset.   
mynnet <- nnet(Class ~ ., train.df, size=1)

## # weights: 83  
## initial value 303.379061   
## iter 10 value 31.708720  
## iter 20 value 22.467540  
## iter 30 value 17.936240  
## iter 40 value 17.198613  
## iter 50 value 12.583639  
## iter 60 value 12.570541  
## iter 70 value 12.567291  
## iter 80 value 12.565315  
## iter 90 value 12.563587  
## iter 100 value 12.563066  
## final value 12.563066   
## stopped after 100 iterations

mynnet.pred <- predict(mynnet,valid.df,type="class")  
  
#visualize the model in a table to see the confusion matrix of accurately predicted values.  
table(mynnet.pred,valid.df$Class)

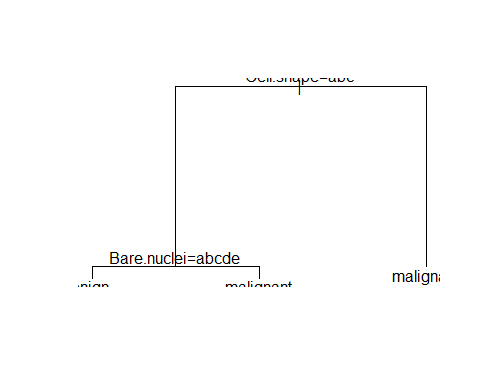
##   
## mynnet.pred benign malignant  
## benign 180 4  
## malignant 5 85

#Fourth Model- Decision Trees

#Install and library the necessary package to run a Decision Tree Model   
library(rpart)

## Warning: package 'rpart' was built under R version 4.0.3

#Using the training data sets- build the DT model. You can then use that DT model to predict the accuracy when applying it to the validation dataset.  
mytree <- rpart(Class ~ ., train.df)  
plot(mytree); text(mytree)



summary(mytree)

## Call:  
## rpart(formula = Class ~ ., data = train.df)  
## n= 409   
##   
## CP nsplit rel error xerror xstd  
## 1 0.83333333 0 1.0000000 1.0000000 0.06497445  
## 2 0.05333333 1 0.1666667 0.1666667 0.03229853  
## 3 0.01000000 2 0.1133333 0.1200000 0.02765488  
##   
## Variable importance  
## Cell.shape Cell.size Bare.nuclei Epith.c.size Bl.cromatin   
## 21 18 16 16 14   
## Normal.nucleoli Cl.thickness   
## 13 1   
##   
## Node number 1: 409 observations, complexity param=0.8333333  
## predicted class=benign expected loss=0.3667482 P(node) =1  
## class counts: 259 150  
## probabilities: 0.633 0.367   
## left son=2 (262 obs) right son=3 (147 obs)  
## Primary splits:  
## Cell.shape splits as LLLRRRRRRR, improve=143.1180, (0 missing)  
## Cell.size splits as LLLRRRRRRR, improve=137.9269, (0 missing)  
## Epith.c.size splits as LLRRRRRRRR, improve=128.0268, (0 missing)  
## Bare.nuclei splits as LLLRRRRRRR, improve=124.3134, (0 missing)  
## Normal.nucleoli splits as LLRRRRRRRR, improve=113.0179, (0 missing)  
## Surrogate splits:  
## Cell.size splits as LLLRRRRRRR, agree=0.949, adj=0.857, (0 split)  
## Epith.c.size splits as LLRRRRRRRR, agree=0.907, adj=0.741, (0 split)  
## Bare.nuclei splits as LLLRRLRRRR, agree=0.880, adj=0.667, (0 split)  
## Bl.cromatin splits as LLLRRRRRRR, agree=0.870, adj=0.639, (0 split)  
## Normal.nucleoli splits as LLRRRRRRRR, agree=0.863, adj=0.619, (0 split)  
##   
## Node number 2: 262 observations, complexity param=0.05333333  
## predicted class=benign expected loss=0.05343511 P(node) =0.6405868  
## class counts: 248 14  
## probabilities: 0.947 0.053   
## left son=4 (252 obs) right son=5 (10 obs)  
## Primary splits:  
## Bare.nuclei splits as LLLLLRRR-R, improve=14.902230, (0 missing)  
## Normal.nucleoli splits as LLRRRR-R-R, improve=11.628060, (0 missing)  
## Cl.thickness splits as LLLLLLRRRR, improve=10.427030, (0 missing)  
## Bl.cromatin splits as LLLLRRR---, improve= 9.506246, (0 missing)  
## Mitoses splits as LRRR--LL-, improve= 7.803346, (0 missing)  
## Surrogate splits:  
## Cl.thickness splits as LLLLLLLLLR, agree=0.977, adj=0.4, (0 split)  
## Marg.adhesion splits as LLLLRRRRRR, agree=0.969, adj=0.2, (0 split)  
## Bl.cromatin splits as LLLLLLR---, agree=0.969, adj=0.2, (0 split)  
## Normal.nucleoli splits as LLLRLR-L-L, agree=0.969, adj=0.2, (0 split)  
## Mitoses splits as LLLR--LL-, agree=0.969, adj=0.2, (0 split)  
##   
## Node number 3: 147 observations  
## predicted class=malignant expected loss=0.07482993 P(node) =0.3594132  
## class counts: 11 136  
## probabilities: 0.075 0.925   
##   
## Node number 4: 252 observations  
## predicted class=benign expected loss=0.01984127 P(node) =0.6161369  
## class counts: 247 5  
## probabilities: 0.980 0.020   
##   
## Node number 5: 10 observations  
## predicted class=malignant expected loss=0.1 P(node) =0.02444988  
## class counts: 1 9  
## probabilities: 0.100 0.900

mytree.pred <- predict(mytree,valid.df,type="class")  
  
#visualize the model in a table to see the confusion matrix of accurately predicted values.  
table(mytree.pred,valid.df$Class)

##   
## mytree.pred benign malignant  
## benign 176 7  
## malignant 9 82

#Fith Model- Quadratic Discrimination Analysis

#Install and library the necessary package to run a Quadratic Discrimination Analysis   
library(MASS)  
  
#convert to integers for this method- Starting with Training data, create a second set with integers  
train.df2 <- train.df  
train.df2$Cl.thickness<-as.integer(train.df2$Cl.thickness)  
train.df2$Class<-ifelse(as.integer(train.df2$Class)==2,1,0)   
train.df2$Mitoses<-as.integer(train.df2$Mitoses)  
train.df2$Cell.size<-as.integer(train.df2$Cell.size)  
train.df2$Cell.shape<-as.integer(train.df2$Cell.shape)  
train.df2$Marg.adhesion<-as.integer(train.df2$Marg.adhesion)  
train.df2$Epith.c.size<-as.integer(train.df2$Epith.c.size)  
train.df2$Bare.nuclei<-as.integer(train.df2$Bare.nuclei)  
train.df2$Bl.cromatin<-as.integer(train.df2$Bl.cromatin)  
train.df2$Normal.nucleoli<-as.integer(train.df2$Normal.nucleoli)   
  
#convert to integers for validation data  
valid.df2 <- valid.df  
valid.df2$Cl.thickness<-as.integer(valid.df2$Cl.thickness)  
valid.df2$Class<-ifelse(as.integer(valid.df2$Class)==2,1,0)  
valid.df2$Mitoses<-as.integer(valid.df2$Mitoses)  
valid.df2$Cell.size<-as.integer(valid.df2$Cell.size)  
valid.df2$Cell.shape<-as.integer(valid.df2$Cell.shape)  
valid.df2$Marg.adhesion<-as.integer(valid.df2$Marg.adhesion)  
valid.df2$Epith.c.size<-as.integer(valid.df2$Epith.c.size)  
valid.df2$Bare.nuclei<-as.integer(valid.df2$Bare.nuclei)  
valid.df2$Bl.cromatin<-as.integer(valid.df2$Bl.cromatin)  
valid.df2$Normal.nucleoli<-as.integer(valid.df2$Normal.nucleoli)   
  
#Remove the ID column that is categorical for the new training and validation datasets since it is categorical and it does not run in this model type.  
train.df2 <- subset(train.df2, select = -c(1))  
valid.df2 <- subset(valid.df2, select = -c(1))  
  
#Using the training data sets- build the QDA model. You can then use that QDA model to predict the accuracy when applying it to the validation dataset.  
myqda <- qda(Class ~ ., train.df2)  
myqda.pred <- predict(myqda, valid.df2)  
  
#visualize the model in a table to see the confusion matrix of accurately predicted values.  
table(myqda.pred$class,valid.df2$Class)

##   
## 0 1  
## 0 173 3  
## 1 12 86

#Sixth Model- Regularised Discriminant Analysis

#Install and library the necessary package to run a Regularised Discriminant Analysis (RDA)  
library(klaR)  
  
#Using the training data sets- build the RDA model. You can then use that RDA model to predict the accuracy when applying it to the validation dataset.  
myrda <- rda(Class ~ ., train.df)  
myrda.pred <- predict(myrda, valid.df)  
  
#visualize the model in a table to see the confusion matrix of accurately predicted values.  
table(myrda.pred$class,valid.df$Class)

##   
## benign malignant  
## benign 177 1  
## malignant 8 88

#Seventh Model Consideration- A loop to generate a confusion matrix for analysis.

BreastCancer <- subset(BreastCancer, select = -c(1))  
  
ans <- numeric(length(BreastCancer[,1]))  
for (i in 1:length(BreastCancer[,1])) {  
 mytree <- rpart(Class ~ ., BreastCancer[-i,])  
 mytree.pred <- predict(mytree,BreastCancer[i,],type="class")  
 ans[i] <- mytree.pred}  
  
ans <- factor(ans,labels=levels(BreastCancer$Class))  
table(ans,BreastCancer$Class)

##   
## ans benign malignant  
## benign 432 20  
## malignant 12 219

#Eight and Final Model- Random Forest

#Install and library the necessary package to run a Random Forest (RF)  
library(randomForest)

## Warning: package 'randomForest' was built under R version 4.0.3

## randomForest 4.6-14

## Type rfNews() to see new features/changes/bug fixes.

#Using the training data sets- build the RF model. You can then use that RF model to predict the accuracy when applying it to the validation dataset.  
myrf <- randomForest(Class ~ .,train.df)  
myrf.pred <- predict(myrf, valid.df)  
  
#visualize the model in a table to see the confusion matrix of accurately predicted values.  
table(myrf.pred, valid.df$Class)

##   
## myrf.pred benign malignant  
## benign 180 2  
## malignant 5 87

#Because we based our models and confusion matrix analysis on the training and validation data sets, we need to apply the projections to the full BreastCancer Dataset for final analysis.

#Update all predicitons to full dataset- Breast Cancer  
#Reload the full dataset and clean it since changes were made to it throughout the code.  
data("BreastCancer")  
  
#Same as the beginning- Drop any NAs to clean the dataset using na.omit. This prevents future failures in the model that cannot process NAs.  
BreastCancer <- na.omit(BreastCancer)  
  
#ID is a categorical variable that is not used to predict relationships in our mdoels. Therefore, we will make the entire column NULL to maintain intergrity but not break the mdoels.  
BreastCancer$Id <- NULL  
  
myrf.predFull <- predict(myrf, BreastCancer)  
myrda.predFull <- predict(myrda, BreastCancer)  
mytree.predFull <- predict(mytree,BreastCancer,type="class")  
mynnet.predFull <- predict(mynnet,BreastCancer,type="class")  
mysvm.predFull <- predict(mysvm, BreastCancer)  
mynb.predFull <- predict(mynb,BreastCancer)

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 2

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 4

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 6

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 13

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 15

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 16

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 21

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 25

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 32

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 38

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 39

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 40

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 42

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 48

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 49

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 50

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 51

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 52

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 55

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 56

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 58

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 62

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 64

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 67

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 70

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 73

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 83

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 84

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 85

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 86

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 97

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 98

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 99

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 100

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 102

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 103

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 104

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 106

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 108

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 109

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 110

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 115

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 116

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 123

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 125

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 127

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 140

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 143

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 145

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 146

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 148

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 155

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 156

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 162

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 170

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 172

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 181

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 182

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 191

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 195

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 200

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 201

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 206

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 210

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 213

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 218

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 222

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 225

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 226

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 227

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 230

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 231

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 232

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 233

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 241

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 245

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 252

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 254

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 257

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 261

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 263

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 266

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 271

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 274

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 276

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 277

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 280

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 281

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 284

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 286

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 288

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 291

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 293

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 297

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 301

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 305

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 307

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 308

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 313

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 317

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 321

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 331

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 335

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 336

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 339

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 340

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 343

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 344

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 345

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 346

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 348

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 368

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 372

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 387

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 398

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 401

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 402

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 404

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 413

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 420

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 439

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 441

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 443

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 451

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 452

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 453

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 468

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 473

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 474

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 475

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 480

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 492

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 505

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 516

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 535

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 539

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 554

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 557

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 560

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 574

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 576

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 577

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 589

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 590

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 606

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 611

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 621

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 633

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 642

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 643

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 653

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 655

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 681

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 682

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 683

#Finally, we combine all the prediction models using the Majority Rules Ensamble Approach

combine.classes<-data.frame(myrf.predFull, myrda.predFull$class,#myqda.pred,  
mytree.predFull,mynnet.predFull,mysvm.predFull, mynb.predFull$class)  
  
combine.classes$myrf.predFull<-ifelse(combine.classes$myrf.pred=="benign", 0, 1)  
combine.classes[,2]<-ifelse(combine.classes[,2]=="benign", 0, 1)  
combine.classes[,3]<-ifelse(combine.classes[,3]=="benign", 0, 1)  
combine.classes[,4]<-ifelse(combine.classes[,4]=="benign", 0, 1)  
combine.classes[,5]<-ifelse(combine.classes[,5]=="benign", 0, 1)  
combine.classes[,6]<-ifelse(combine.classes[,6]=="benign", 0, 1)  
majority.vote=rowSums(combine.classes)  
head(majority.vote)

## 1 2 3 4 5 6   
## 0 3 0 6 0 6

combine.classes[,7]<-rowSums(combine.classes)  
combine.classes[,8]<-ifelse(combine.classes[,7]>=3, "malignant", "benign")  
  
  
table <- table(combine.classes[,8], BreastCancer$Class)  
table

##   
## benign malignant  
## benign 433 2  
## malignant 11 237

accuracy <- sum(diag(table))/sum(table)  
accuracy\*100

## [1] 98.09663