Project-2

Subha

3/5/2021

**Data Description:**

The BreastCancer dataset 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 1- Target class

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

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

## Loading required package: lattice

## Warning: package 'lattice' was built under R version 4.0.2

## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 4.0.2

library(MASS)

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

library(mlbench)

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

library(tidyverse)

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

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

## v tibble 3.0.1 v dplyr 1.0.3  
## v tidyr 1.1.2 v stringr 1.4.0  
## v readr 1.3.1 v forcats 0.5.0  
## v purrr 0.3.4

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

## Warning: package 'readr' was built under R version 4.0.2

## Warning: package 'purrr' was built under R version 4.0.2

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

## Warning: package 'stringr' was built under R version 4.0.2

## Warning: package 'forcats' was built under R version 4.0.2

## -- 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 method to implement to substitute missing values.  
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%"

**It can be found that there are only 2.29% of missing values in the variable Bare.nuclei. Either we can 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 modelling phase. It is better to remove it  
  
BreastCancer.df$Id <- NULL  
  
  
  
# LEt us 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

#Splitting the dataset

#install.packages("caTools")  
library(caTools)

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

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

print(dim(train)); print(dim(test))

## [1] 479 10

## [1] 204 10

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"

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***

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

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

# 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.  
# I manually copied the cost and gamma from console messages above to parameters below.  
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 probability = TRUE while type="prob" is not needed  
#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)

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

mynb <- naiveBayes(Result ~ ., train, laplace = 0)  
mynb.pred <- predict(mynb,test,type="class")  
mynb.prob <- predict(mynb,test,type="raw")  
table(mynb.pred,test$Result)

##   
## mynb.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%"

1. ***Neural Network***

#install.packages("nnet")  
library(nnet)

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

mynnet <- 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 15.625601  
## iter 80 value 6.756225  
## iter 90 value 6.749315  
## iter 100 value 6.747910  
## final value 6.747910   
## stopped after 100 iterations

mynnet.pred <- predict(mynnet,test,type="class")  
mynnet.prob <- predict(mynnet,test,type="raw")  
table(mynnet.pred,test$Result)

##   
## mynnet.pred benign malignant  
## benign 125 5  
## malignant 4 70

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

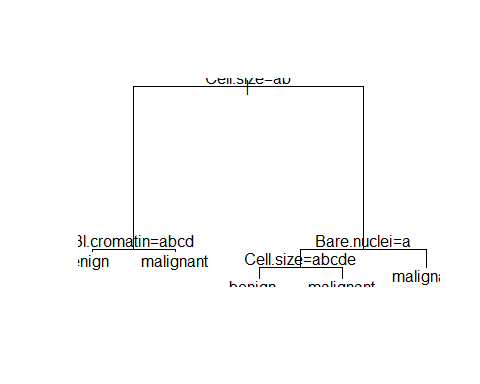
## [1] "The Accuracy of neuralnetwork model is 95.1%"

1. ***Decision Trees***

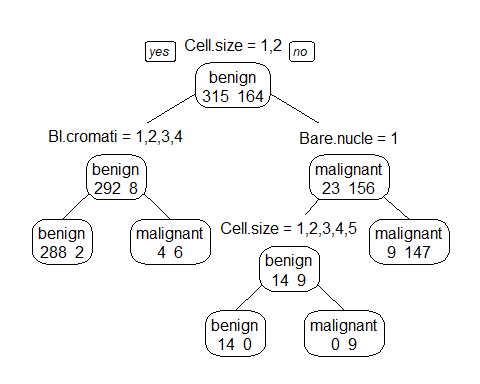
#install.packages("MASS")  
library(MASS)  
library(rpart)  
library(rpart.plot)

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

mytree <- rpart(Result ~ ., train)  
plot(mytree); text(mytree)



prp(mytree, type = 1, extra = 1, split.font = 1, varlen = -10)



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

##   
## pred.pred benign malignant  
## benign 119 5  
## malignant 10 70

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***

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

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

## Loading required package: grid

## Loading required package: mvtnorm

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

## Loading required package: modeltools

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

## Loading required package: stats4

## Loading required package: strucchange

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

## Loading required package: zoo

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

##   
## Attaching package: 'zoo'

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

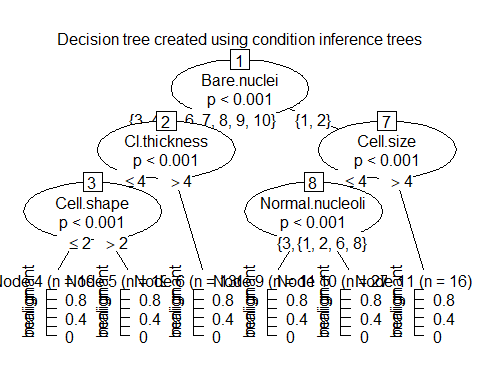
## Loading required package: sandwich

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

##   
## Attaching package: 'strucchange'

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

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



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

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

ctaccuracy <- round(((126 +71) / nrow(test))\*100,2)  
paste0("The Accuracy of Decision Trees model is ", ctaccuracy, "%")

## [1] "The Accuracy of Decision Trees model is 96.57%"

1. ***Random Forests***

#install.packages("randomForest")  
#install.packages("party")  
library(randomForest)

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

## randomForest 4.6-14

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

##   
## Attaching package: 'randomForest'

## The following object is masked from 'package:dplyr':  
##   
## combine

## The following object is masked from 'package:ggplot2':  
##   
## margin

library(party)  
#Applying conditional inference trees as base learners for random forests  
myrf <- randomForest(Result ~ ., train, control = cforest\_unbiased(mtry = 9))  
  
rf.pred <- predict(myrf, newdata=test)  
  
table(rf.pred, test$Result)

##   
## rf.pred benign malignant  
## benign 124 4  
## malignant 5 71

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%"

***7.Leave-1-Out Cross Validation (LOOCV)***

library(caret)  
  
ans <- numeric(length(BreastCancer.df[,1]))  
for (i in 1:length(BreastCancer.df[,1])) {  
 mytree <- rpart(Class ~ ., BreastCancer.df[-i,])  
 mytree.predloo <- predict(mytree,BreastCancer.df[i,],type="class")  
 ans[i] <- mytree.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']\*100  
  
accuracy\_LOOCV <- round(acc,2)  
  
paste0("The Accuracy of LOOCV model is ", accuracy\_LOOCV, "%")

## [1] "The Accuracy of LOOCV model is 95.02%"

1. ***bagging (bootstrap aggregating)***

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

## Loading required package: ipred

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

ip <- bagging(Result ~ ., data=train)   
  
ip.pred <- predict(ip, newdata=test)  
ip.prob <- predict(ip, type="prob", newdata=test)  
table(ip.pred,test$Result)

##   
## ip.pred benign malignant  
## benign 122 6  
## malignant 7 69

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%"

***9.Quadratic Discriminant Analysis***

library(MASS)  
library(dplyr)  
train.num <- train %>% dplyr::select(-Result) %>% mutate\_if(is.factor,as.character)%>% mutate\_if(is.character,as.numeric)  
train.num$Result <- train$Result  
test.num <- test%>%dplyr::select(-Result) %>% mutate\_if(is.factor,as.character)%>% mutate\_if(is.character,as.numeric)  
test.num$Result <- test$Result  
  
qda <- qda(Result~., data = train.num) #qda, formula, right hand is non-factor  
qda.pred <- predict(qda, test.num)$class  
qda.prob <- predict(qda, test.num)$posterior   
table(qda.pred,test.num$Result)

##   
## 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%"

***10.Regularised Discriminant Analysis***

#not able to use test  
  
library(klaR)  
rda <- rda(Result~., data = train)  
rda.pred <- predict(rda, test)$class  
rda.prob <- predict(rda, test)$posterior  
table(rda.pred,test$Result)

##   
## rda.pred benign malignant  
## benign 125 2  
## malignant 4 73

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%"

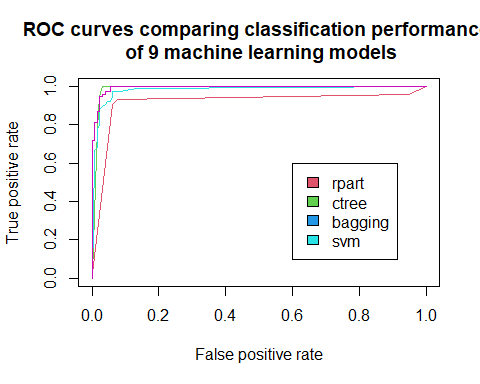
### Plot ROC curves to compare the performance of the individual classifiers.

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

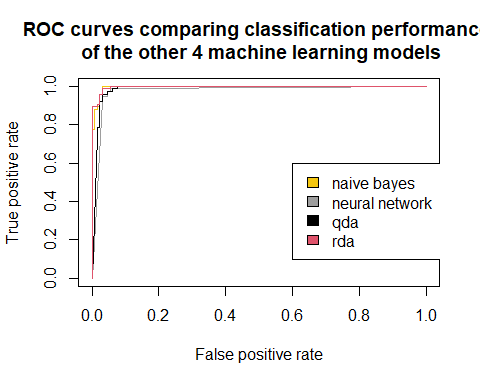
## Warning: package 'ROCR' was built under R version 4.0.4

# 1.svm  
svm.prob.rocr <- prediction(attr(x.svm.prob, "probabilities")[,2], test[,'Result'])  
x.svm.perf <- performance(svm.prob.rocr, "tpr","fpr")  
  
#2.nb  
x.nb.prob.rocr <- prediction(mynb.prob[,2], test[,'Result'])  
x.nb.perf <- performance(x.nb.prob.rocr, "tpr","fpr")  
  
#3.nnet  
x.nn.prob.rocr <- prediction(mynnet.prob, test[,'Result'])  
x.nn.perf <- performance(x.nn.prob.rocr, "tpr","fpr")  
  
#4. Decision Trees  
x.rp.prob.rocr <- prediction(pred.prob[,2], test[,'Result'])  
x.rp.perf <- performance(x.rp.prob.rocr, "tpr","fpr")  
  
#5. conditional inference trees  
x.ct.prob.rocr <- prediction(ct.prob, test[,'Result'])  
x.ct.perf <- performance(x.ct.prob.rocr, "tpr","fpr")  
  
#6. Bagging  
x.ip.prob.rocr <- prediction(ip.prob[,2], test[,'Result'])  
x.ip.perf <- performance(x.ip.prob.rocr, "tpr","fpr")  
  
#7.qda  
x.qda.prob.rocr <- prediction(qda.prob[,2], test[,'Result'])  
x.qda.perf <- performance(x.qda.prob.rocr, "tpr","fpr")  
  
#8.rda  
x.rda.prob.rocr <- prediction(rda.prob[,2], test[,'Result'])  
x.rda.perf <- performance(x.rda.prob.rocr, "tpr","fpr")

####### 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.rp.perf, col=2, main="ROC curves comparing classification performance \n of 9 machine learning models") #   
legend(0.6, 0.6, c('rpart', 'ctree','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.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 learning models")  
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, Regularized Discriminant Analysis and random forest. The overall accuracy of the ensemble model is **98.04%**

stackdf <- data.frame(cbind(pred.pred, ct.pred, rf.pred,ip.pred, x.svm.pred, mynb.pred,mynnet.pred,qda.pred,rda.pred))  
  
names(stackdf) <-c('Decision.Tree','Conditional.Inference.Tree','Random.Forest','Bootstrap','svm','naive.bayes','neutral.network','qda','rda')  
levels(stackdf$neutral.network) =c('1','2')  
  
finaldf <-stackdf%>% sapply(FUN = function(x)(ifelse(x=='1',0,1)))  
finaldf<- addmargins(finaldf, margin = 2) # table/arragy, margin =2 aggregate by col   
finaldf <- data.frame(finaldf)  
finaldf$predition <- ifelse(finaldf$Sum >=5, 'malignant','benign')  
  
  
#confusion matrix   
library(caret)  
finalcm <-confusionMatrix(as.factor(finaldf$predition), test$Result, positive = 'malignant')  
finalcm

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction benign malignant  
## benign 125 0  
## malignant 4 75  
##   
## Accuracy : 0.9804   
## 95% CI : (0.9506, 0.9946)  
## No Information Rate : 0.6324   
## P-Value [Acc > NIR] : <2e-16   
##   
## Kappa : 0.9583   
##   
## Mcnemar's Test P-Value : 0.1336   
##   
## Sensitivity : 1.0000   
## Specificity : 0.9690   
## Pos Pred Value : 0.9494   
## Neg Pred Value : 1.0000   
## Prevalence : 0.3676   
## Detection Rate : 0.3676   
## Detection Prevalence : 0.3873   
## Balanced Accuracy : 0.9845   
##   
## 'Positive' Class : malignant   
##

acc\_ensemble <- finalcm$overall['Accuracy']\*100  
  
Ensemble.acc <- round(acc\_ensemble,2)  
  
  
paste0("Therefore the overall ensemble majority model accuracy is ",Ensemble.acc,"%")

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