AndrewPhillips\_Project\_2\_Final

Andrew Phillips

3/7/2021

library(mlbench)

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

library(caret)

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

## Loading required package: lattice

## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 4.0.3

library(e1071)

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

library(klaR)

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

## Loading required package: MASS

library(nnet)

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

library(rpart)  
library(MASS)  
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.

##   
## Attaching package: 'randomForest'

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

library(caTools)

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

**Since “Bare.nuclei” contains a few missing cells, they will be replaced with 1. The id column will has been removed as it does not contribute to the analysis:**

data("BreastCancer")  
  
  
for (i in 1:length(BreastCancer)[1]) {  
d <- ifelse(sum(is.na(BreastCancer[i]))>5, names(BreastCancer[i]),F)  
 #print(b)   
print(d)  
}

## [1] FALSE  
## [1] FALSE  
## [1] FALSE  
## [1] FALSE  
## [1] FALSE  
## [1] FALSE  
## [1] "Bare.nuclei"  
## [1] FALSE  
## [1] FALSE  
## [1] FALSE  
## [1] FALSE

str(BreastCancer$Bare.nuclei)

## Factor w/ 10 levels "1","2","3","4",..: 1 10 2 4 1 10 10 1 1 1 ...

#replace missing values with 1:  
BreastCancer$Bare.nuclei[is.na(BreastCancer$Bare.nuclei)] <- 1  
sum(is.na(BreastCancer$Bare.nuclei))

## [1] 0

BreastCancer <- BreastCancer[,-c(1)]  
  
set.seed(100)  
  
spl = sample.split(BreastCancer, SplitRatio = 0.7)  
train = subset(BreastCancer, spl==TRUE)  
test = subset(BreastCancer, spl==FALSE)  
dim(BreastCancer)

## [1] 699 10

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

## [1] 489 10

## [1] 210 10

**The overall SVM accuracy is 95.71%:**

mysvm <- svm(Class ~ ., train)  
mysvm.pred <- predict(mysvm, test)  
  
  
svmcv <- confusionMatrix(factor(mysvm.pred),test$Class)  
svmcv

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction benign malignant  
## benign 128 4  
## malignant 5 73  
##   
## Accuracy : 0.9571   
## 95% CI : (0.9202, 0.9802)  
## No Information Rate : 0.6333   
## P-Value [Acc > NIR] : <2e-16   
##   
## Kappa : 0.908   
##   
## Mcnemar's Test P-Value : 1   
##   
## Sensitivity : 0.9624   
## Specificity : 0.9481   
## Pos Pred Value : 0.9697   
## Neg Pred Value : 0.9359   
## Prevalence : 0.6333   
## Detection Rate : 0.6095   
## Detection Prevalence : 0.6286   
## Balanced Accuracy : 0.9552   
##   
## 'Positive' Class : benign   
##

svmcv\_acc <- svmcv$overall['Accuracy'] \* 100  
  
anssvmcv <- round(svmcv\_acc,2)  
cat(anssvmcv,"% Accuracy")

## 95.71 % Accuracy

**The overall NB accuracy is 96.67%:**

mynb <- naiveBayes(Class ~ ., train, laplace = 0)  
mynb.pred <- predict(mynb,test)  
  
nbcv <- confusionMatrix(factor(mynb.pred),test$Class)  
nbcv

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction benign malignant  
## benign 128 2  
## malignant 5 75  
##   
## Accuracy : 0.9667   
## 95% CI : (0.9325, 0.9865)  
## No Information Rate : 0.6333   
## P-Value [Acc > NIR] : <2e-16   
##   
## Kappa : 0.9288   
##   
## Mcnemar's Test P-Value : 0.4497   
##   
## Sensitivity : 0.9624   
## Specificity : 0.9740   
## Pos Pred Value : 0.9846   
## Neg Pred Value : 0.9375   
## Prevalence : 0.6333   
## Detection Rate : 0.6095   
## Detection Prevalence : 0.6190   
## Balanced Accuracy : 0.9682   
##   
## 'Positive' Class : benign   
##

nbcv\_acc <- nbcv$overall['Accuracy'] \* 100  
  
ansnbcv <- round(nbcv\_acc,2)  
cat(ansnbcv,"% Accuracy")

## 96.67 % Accuracy

**The overall neural network accuracy is 90.95%:**

mynnet <- nnet(Class ~ ., train, size=2)

## # weights: 165  
## initial value 368.817582   
## iter 10 value 6.129777  
## iter 20 value 0.084270  
## iter 30 value 0.001140  
## final value 0.000097   
## converged

mynnet.pred <- predict(mynnet,test,type="class")  
  
nncv <- confusionMatrix(factor(mynnet.pred),test$Class)  
nncv

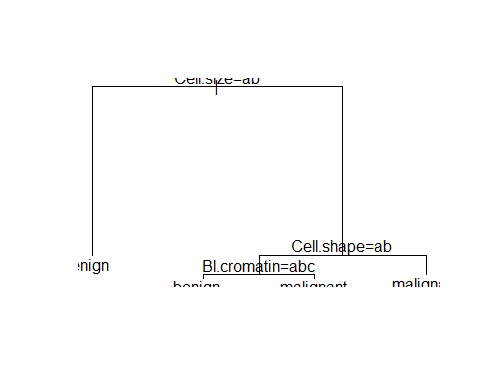
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction benign malignant  
## benign 127 13  
## malignant 6 64  
##   
## Accuracy : 0.9095   
## 95% CI : (0.8623, 0.9446)  
## No Information Rate : 0.6333   
## P-Value [Acc > NIR] : <2e-16   
##   
## Kappa : 0.8014   
##   
## Mcnemar's Test P-Value : 0.1687   
##   
## Sensitivity : 0.9549   
## Specificity : 0.8312   
## Pos Pred Value : 0.9071   
## Neg Pred Value : 0.9143   
## Prevalence : 0.6333   
## Detection Rate : 0.6048   
## Detection Prevalence : 0.6667   
## Balanced Accuracy : 0.8930   
##   
## 'Positive' Class : benign   
##

nncv\_acc <- nncv$overall['Accuracy'] \* 100  
  
ansnncv <- round(nncv\_acc,2)  
cat(ansnncv,"% Accuracy")

## 90.95 % Accuracy

**The overall decision tree accuracy is 93.33%:**

#Decision trees  
  
mytree <- rpart(Class ~ ., train)  
plot(mytree); text(mytree) # in "iris\_tree.ps"



mytree.pred <- predict(mytree,test,type="class")  
  
dscv <- confusionMatrix(mytree.pred,test$Class)  
dscv

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction benign malignant  
## benign 123 4  
## malignant 10 73  
##   
## Accuracy : 0.9333   
## 95% CI : (0.8907, 0.9631)  
## No Information Rate : 0.6333   
## P-Value [Acc > NIR] : <2e-16   
##   
## Kappa : 0.8588   
##   
## Mcnemar's Test P-Value : 0.1814   
##   
## Sensitivity : 0.9248   
## Specificity : 0.9481   
## Pos Pred Value : 0.9685   
## Neg Pred Value : 0.8795   
## Prevalence : 0.6333   
## Detection Rate : 0.5857   
## Detection Prevalence : 0.6048   
## Balanced Accuracy : 0.9364   
##   
## 'Positive' Class : benign   
##

dscv\_acc <- dscv$overall['Accuracy'] \* 100  
  
ansdscv <- round(dscv\_acc,2)  
cat(ansdscv,"% Accuracy")

## 93.33 % Accuracy

**The following is the LOOCV. Due to differing lengths, it would not work within the ensemble method. The overall LOOCV accuracy is 93.81%:**

# Leave-1-Out Cross Validation (LOOCV)  
  
ans <- numeric(length(test[,1]))  
for (i in 1:length(test[,1])) {  
 mytree <- rpart(Class ~ ., train[-i,])  
 mytree.predloocv <- predict(mytree,test[i,],type="class")  
 ans[i] <- mytree.predloocv  
 }  
str(train$Class)

## Factor w/ 2 levels "benign","malignant": 1 1 1 1 2 1 1 1 1 1 ...

ans <- factor(ans,labels=levels(test$Class))  
  
loocvcm <- confusionMatrix(ans, test$Class)  
loocvcm

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction benign malignant  
## benign 123 3  
## malignant 10 74  
##   
## Accuracy : 0.9381   
## 95% CI : (0.8965, 0.9666)  
## No Information Rate : 0.6333   
## P-Value [Acc > NIR] : < 2e-16   
##   
## Kappa : 0.8692   
##   
## Mcnemar's Test P-Value : 0.09609   
##   
## Sensitivity : 0.9248   
## Specificity : 0.9610   
## Pos Pred Value : 0.9762   
## Neg Pred Value : 0.8810   
## Prevalence : 0.6333   
## Detection Rate : 0.5857   
## Detection Prevalence : 0.6000   
## Balanced Accuracy : 0.9429   
##   
## 'Positive' Class : benign   
##

# The same as above in this case  
loocvcm\_acc <- loocvcm$overall['Accuracy'] \* 100  
  
ansac <- round(loocvcm\_acc,2)  
cat(ansac,"% Accuracy")

## 93.81 % Accuracy

**The overall QDA accuracy is 93.81%:**

#Quadratic Discriminant Analysis  
  
trainqda <- lapply(train,as.numeric)  
testqda <- lapply(test,as.numeric)  
trainqda$Class <- factor(trainqda$Class, labels = c("benign", "malignant"))  
testqda$Class <- factor(testqda$Class, labels=c("benign","malignant"))  
  
myqda <- qda(Class ~ ., trainqda)  
str(BreastCancer)

## 'data.frame': 699 obs. of 10 variables:  
## $ 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 ...

myqda.pred <- predict(myqda, testqda)  
table(myqda.pred$class,testqda$Class)

##   
## benign malignant  
## benign 123 3  
## malignant 10 74

qdacv <- confusionMatrix(myqda.pred$class, testqda$Class)  
qdacv

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction benign malignant  
## benign 123 3  
## malignant 10 74  
##   
## Accuracy : 0.9381   
## 95% CI : (0.8965, 0.9666)  
## No Information Rate : 0.6333   
## P-Value [Acc > NIR] : < 2e-16   
##   
## Kappa : 0.8692   
##   
## Mcnemar's Test P-Value : 0.09609   
##   
## Sensitivity : 0.9248   
## Specificity : 0.9610   
## Pos Pred Value : 0.9762   
## Neg Pred Value : 0.8810   
## Prevalence : 0.6333   
## Detection Rate : 0.5857   
## Detection Prevalence : 0.6000   
## Balanced Accuracy : 0.9429   
##   
## 'Positive' Class : benign   
##

qdacv\_acc <- qdacv$overall['Accuracy'] \* 100  
  
ansqdacv <- round(qdacv\_acc,2)  
cat(ansqdacv,"% Accuracy")

## 93.81 % Accuracy

**The overall RDA accuracy is 96.67%:**

#Regularised Discriminant Analysis  
  
myrda <- rda(Class ~ ., train)  
myrda.pred <- predict(myrda, test)  
  
rdacv <- confusionMatrix(factor(myrda.pred$class), test$Class)  
rdacv

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction benign malignant  
## benign 126 1  
## malignant 7 76  
##   
## Accuracy : 0.9619   
## 95% CI : (0.9263, 0.9834)  
## No Information Rate : 0.6333   
## P-Value [Acc > NIR] : <2e-16   
##   
## Kappa : 0.9193   
##   
## Mcnemar's Test P-Value : 0.0771   
##   
## Sensitivity : 0.9474   
## Specificity : 0.9870   
## Pos Pred Value : 0.9921   
## Neg Pred Value : 0.9157   
## Prevalence : 0.6333   
## Detection Rate : 0.6000   
## Detection Prevalence : 0.6048   
## Balanced Accuracy : 0.9672   
##   
## 'Positive' Class : benign   
##

rdacv\_acc <- rdacv$overall['Accuracy'] \* 100  
  
ansrdacv <- round(rdacv\_acc,2)  
cat(ansrdacv,"% Accuracy")

## 96.19 % Accuracy

**The overall random forest accuracy is 95.24%:**

#Random Forests  
  
myrf <- randomForest(Class ~ ., train)  
myrf.pred <- predict(myrf, test)  
  
rfcv <- confusionMatrix(myrf.pred, test$Class)  
rfcv

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction benign malignant  
## benign 126 3  
## malignant 7 74  
##   
## Accuracy : 0.9524   
## 95% CI : (0.9142, 0.9769)  
## No Information Rate : 0.6333   
## P-Value [Acc > NIR] : <2e-16   
##   
## Kappa : 0.8986   
##   
## Mcnemar's Test P-Value : 0.3428   
##   
## Sensitivity : 0.9474   
## Specificity : 0.9610   
## Pos Pred Value : 0.9767   
## Neg Pred Value : 0.9136   
## Prevalence : 0.6333   
## Detection Rate : 0.6000   
## Detection Prevalence : 0.6143   
## Balanced Accuracy : 0.9542   
##   
## 'Positive' Class : benign   
##

rfcv\_acc <- rfcv$overall['Accuracy'] \* 100  
  
ansrfcv <- round(rfcv\_acc,2)  
cat(ansrfcv,"% Accuracy")

## 95.24 % Accuracy

**After stacking the algorithms in a “majority rule” ensemble fashion utilizing the previous algorithms svm, naive bayes, neural network, QDA, RDA, decision tree, and random forest, the overall accuracy of the ensemble model is 96.67%:**

myrda.pred\_s <- myrda.pred$class  
myqda.pred\_s <- myqda.pred$class  
stackdf <- data.frame(mysvm.pred,mynb.pred,mynnet.pred,myqda.pred\_s,mytree.pred,myrda.pred\_s, myrf.pred, Class = test$Class, stringsAsFactors = F)  
  
stvm <- svm(Class ~ ., stackdf)  
  
stvm.pred <- predict(stvm, test)  
  
stcv <- confusionMatrix(stvm.pred, test$Class)  
stcv

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction benign malignant  
## benign 128 2  
## malignant 5 75  
##   
## Accuracy : 0.9667   
## 95% CI : (0.9325, 0.9865)  
## No Information Rate : 0.6333   
## P-Value [Acc > NIR] : <2e-16   
##   
## Kappa : 0.9288   
##   
## Mcnemar's Test P-Value : 0.4497   
##   
## Sensitivity : 0.9624   
## Specificity : 0.9740   
## Pos Pred Value : 0.9846   
## Neg Pred Value : 0.9375   
## Prevalence : 0.6333   
## Detection Rate : 0.6095   
## Detection Prevalence : 0.6190   
## Balanced Accuracy : 0.9682   
##   
## 'Positive' Class : benign   
##

stcv\_acc <- stcv$overall['Accuracy'] \* 100  
  
ansstcv <- round(stcv\_acc,2)  
  
stdf <- rbind("SVM Accuracy" = anssvmcv, "Naive Bayes Accuracy" = ansnbcv, "Neural Network Accuracy" = ansnncv, "Decision Tree Accuracy" = ansdscv, "LOOCV Accuracy" = loocvcm\_acc, "QDA Accuracy" = ansqdacv, "RDA Accuracy" = ansrdacv, "Random Forest Accuracy" = ansrfcv , "Ensemble Majority Accuracy" = ansstcv)  
stdf

## Accuracy  
## SVM Accuracy 95.71000  
## Naive Bayes Accuracy 96.67000  
## Neural Network Accuracy 90.95000  
## Decision Tree Accuracy 93.33000  
## LOOCV Accuracy 93.80952  
## QDA Accuracy 93.81000  
## RDA Accuracy 96.19000  
## Random Forest Accuracy 95.24000  
## Ensemble Majority Accuracy 96.67000

cat("The overall ensemble Majority model accuracy is",ansstcv,"%")

## The overall ensemble Majority model accuracy is 96.67 %