Project Two

2023-03-19

library(caTools)   
library(caret)

## Loading required package: ggplot2

## Loading required package: lattice

library(mice)

##   
## Attaching package: 'mice'

## The following object is masked from 'package:stats':  
##   
## filter

## The following objects are masked from 'package:base':  
##   
## cbind, rbind

library(e1071)  
library(rpart)   
library(randomForest)

## randomForest 4.7-1.1

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

##   
## Attaching package: 'randomForest'

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

# Load the mlbench package  
library(mlbench)  
  
# Load breast cancer dataset  
data("BreastCancer")  
  
# Exploring DataSet  
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 ...

# Finding the levels of target class  
levels(BreastCancer$Class)

## [1] "benign" "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   
##   
##   
##   
##   
##

# Data Cleaning  
# Removing NA values and ID(1st column) from dataset using library mice  
dataset\_impute <- mice(BreastCancer[,2:10], print = FALSE)   
  
# Adding Target class to the imputed dataset without NA  
BreastCancer <- cbind(BreastCancer[,11, drop = FALSE], mice::complete(dataset\_impute, 1))   
  
summary(BreastCancer)

## Class Cl.thickness Cell.size Cell.shape Marg.adhesion  
## benign :458 1 :145 1 :384 1 :353 1 :407   
## malignant:241 5 :130 10 : 67 2 : 59 2 : 58   
## 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 :410 2 :166 1 :443 1 :579   
## 3 : 72 10 :133 3 :165 10 : 61 2 : 35   
## 4 : 48 2 : 31 1 :152 3 : 44 3 : 33   
## 1 : 47 5 : 30 7 : 73 2 : 36 10 : 14   
## 6 : 41 3 : 29 4 : 40 8 : 24 4 : 12   
## 5 : 39 4 : 21 5 : 34 6 : 22 7 : 9   
## (Other): 66 (Other): 45 (Other): 69 (Other): 69 (Other): 17

# Splitting Dataset into training, test and to predict  
set.seed(150)   
split=sample.split(BreastCancer, SplitRatio = 0.7)  
training\_set=subset(BreastCancer,split==TRUE)  
test\_set=subset(BreastCancer,split==FALSE)  
dim(training\_set)

## [1] 490 10

dim(test\_set)

## [1] 209 10

topredict\_set<-test\_set[2:10]  
dim(topredict\_set)

## [1] 209 9

# Model 1: Naive Bayes Classifier  
model\_naive<- naiveBayes(Class ~ ., data = training\_set)  
preds\_naive <- predict(model\_naive, newdata = topredict\_set)  
(conf\_matrix\_naive <- table(preds\_naive, test\_set$Class))

##   
## preds\_naive benign malignant  
## benign 129 2  
## malignant 6 72

confusionMatrix(conf\_matrix\_naive)

## Confusion Matrix and Statistics  
##   
##   
## preds\_naive benign malignant  
## benign 129 2  
## malignant 6 72  
##   
## Accuracy : 0.9617   
## 95% CI : (0.926, 0.9833)  
## No Information Rate : 0.6459   
## P-Value [Acc > NIR] : <2e-16   
##   
## Kappa : 0.9173   
##   
## Mcnemar's Test P-Value : 0.2888   
##   
## Sensitivity : 0.9556   
## Specificity : 0.9730   
## Pos Pred Value : 0.9847   
## Neg Pred Value : 0.9231   
## Prevalence : 0.6459   
## Detection Rate : 0.6172   
## Detection Prevalence : 0.6268   
## Balanced Accuracy : 0.9643   
##   
## 'Positive' Class : benign   
##

# Model 2: Random Forest Classifier  
model\_rf <- randomForest(Class ~ ., data = training\_set, importance=TRUE, ntree = 5)  
  
preds\_rf <- predict(model\_rf, topredict\_set)   
  
(conf\_matrix\_forest <- table(preds\_rf, test\_set$Class))

##   
## preds\_rf benign malignant  
## benign 128 3  
## malignant 7 71

confusionMatrix(conf\_matrix\_forest)

## Confusion Matrix and Statistics  
##   
##   
## preds\_rf benign malignant  
## benign 128 3  
## malignant 7 71  
##   
## Accuracy : 0.9522   
## 95% CI : (0.9138, 0.9768)  
## No Information Rate : 0.6459   
## P-Value [Acc > NIR] : <2e-16   
##   
## Kappa : 0.8967   
##   
## Mcnemar's Test P-Value : 0.3428   
##   
## Sensitivity : 0.9481   
## Specificity : 0.9595   
## Pos Pred Value : 0.9771   
## Neg Pred Value : 0.9103   
## Prevalence : 0.6459   
## Detection Rate : 0.6124   
## Detection Prevalence : 0.6268   
## Balanced Accuracy : 0.9538   
##   
## 'Positive' Class : benign   
##

# Model 3: DecisionTree Classifier  
model\_dtree<- rpart(Class ~ ., data=training\_set)  
preds\_dtree <- predict(model\_dtree,newdata=topredict\_set, type = "class")  
  
# plot(preds\_dtree, main="Decision tree created using rpart")  
(conf\_matrix\_dtree <- table(preds\_dtree, test\_set$Class))

##   
## preds\_dtree benign malignant  
## benign 127 5  
## malignant 8 69

confusionMatrix(conf\_matrix\_dtree)

## Confusion Matrix and Statistics  
##   
##   
## preds\_dtree benign malignant  
## benign 127 5  
## malignant 8 69  
##   
## Accuracy : 0.9378   
## 95% CI : (0.896, 0.9665)  
## No Information Rate : 0.6459   
## P-Value [Acc > NIR] : <2e-16   
##   
## Kappa : 0.8652   
##   
## Mcnemar's Test P-Value : 0.5791   
##   
## Sensitivity : 0.9407   
## Specificity : 0.9324   
## Pos Pred Value : 0.9621   
## Neg Pred Value : 0.8961   
## Prevalence : 0.6459   
## Detection Rate : 0.6077   
## Detection Prevalence : 0.6316   
## Balanced Accuracy : 0.9366   
##   
## 'Positive' Class : benign   
##

# Model 4: SVM Classifier  
model\_svm <- svm(Class ~ ., data = training\_set, kernel = "linear")  
preds\_svm <- predict(model\_svm, newdata = topredict\_set)  
  
(conf\_matrix\_svm <- table(preds\_svm, test\_set$Class))

##   
## preds\_svm benign malignant  
## benign 130 5  
## malignant 5 69

confusionMatrix(conf\_matrix\_svm)

## Confusion Matrix and Statistics  
##   
##   
## preds\_svm benign malignant  
## benign 130 5  
## malignant 5 69  
##   
## Accuracy : 0.9522   
## 95% CI : (0.9138, 0.9768)  
## No Information Rate : 0.6459   
## P-Value [Acc > NIR] : <2e-16   
##   
## Kappa : 0.8954   
##   
## Mcnemar's Test P-Value : 1   
##   
## Sensitivity : 0.9630   
## Specificity : 0.9324   
## Pos Pred Value : 0.9630   
## Neg Pred Value : 0.9324   
## Prevalence : 0.6459   
## Detection Rate : 0.6220   
## Detection Prevalence : 0.6459   
## Balanced Accuracy : 0.9477   
##   
## 'Positive' Class : benign   
##

# Combine the classifiers in an ensemble  
# Make predictions on the test set using each model  
preds\_naive <- predict(model\_naive, newdata = test\_set[,2:10])  
preds\_rf <- predict(model\_rf, newdata = test\_set[,2:10])  
preds\_dtree <- predict(model\_dtree, newdata = test\_set[,2:10], type = "class")  
preds\_svm <- predict(model\_svm, newdata = test\_set[,2:10])  
  
# Combine the predictions using majority voting  
ensemble\_preds <- ifelse(rowSums(cbind(preds\_naive == "M",   
 preds\_rf == "M",   
 preds\_dtree == "M",   
 preds\_svm == "M")) >= 2, "M", "B")  
  
# Convert both variables into factors with the same levels  
ensemble\_preds <- factor(ensemble\_preds, levels = c("B", "M"))  
test\_set$Class <- factor(test\_set$Class, levels = c("B", "M"))  
  
# Evaluate the performance of the ensemble classifier  
conf\_matrix\_ensemble <- table(ensemble\_preds, test\_set$Class)  
confusionMatrix(ensemble\_preds, test\_set$Class)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction B M  
## B 0 0  
## M 0 0  
##   
## Accuracy : NaN   
## 95% CI : (NA, NA)  
## No Information Rate : NA   
## P-Value [Acc > NIR] : NA   
##   
## Kappa : NaN   
##   
## Mcnemar's Test P-Value : NA   
##   
## Sensitivity : NA   
## Specificity : NA   
## Pos Pred Value : NA   
## Neg Pred Value : NA   
## Prevalence : NaN   
## Detection Rate : NaN   
## Detection Prevalence : NaN   
## Balanced Accuracy : NA   
##   
## 'Positive' Class : B   
##