Project 2 560

Cameron Safai

2023-03-14

require(mlbench)

## Loading required package: mlbench

library(janitor)

##   
## Attaching package: 'janitor'

## The following objects are masked from 'package:stats':  
##   
## chisq.test, fisher.test

library(neuralnet)  
library(e1071)  
library(caret)

## Loading required package: ggplot2

## Loading required package: lattice

library(dplyr)

##   
## Attaching package: 'dplyr'

## The following object is masked from 'package:neuralnet':  
##   
## compute

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

# load the data set  
data(BreastCancer)  
ls(BreastCancer)

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

#remove NA's  
BreastCancer <- na.omit(BreastCancer)  
  
# remove the unique identifier  
BreastCancer <- BreastCancer[-1]  
  
head(BreastCancer)

## 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

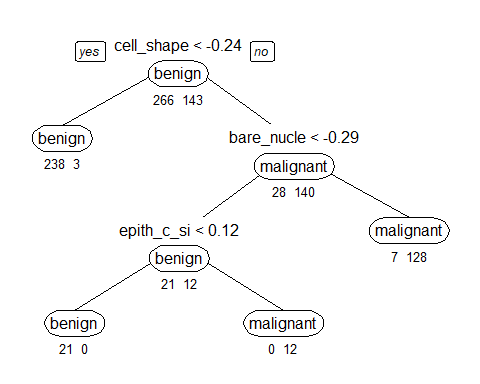
str(BreastCancer)

## 'data.frame': 683 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 ...

BreastCancer2 <- data.frame(sapply(BreastCancer[1:9], function(x) as.numeric(as.character(x))))  
BreastCancer2 <- scale(BreastCancer2[,1:9],center=TRUE,scale=TRUE)  
BreastCancer2 <- na.omit(BreastCancer2)  
  
  
#Clean names  
BreastCancer <- data.frame(BreastCancer$Class, BreastCancer2)  
BreastCancer <- clean\_names(BreastCancer)

#Split train and validation data sets  
set.seed(246)  
splitPercent <- floor(0.6 \* nrow(BreastCancer))  
trainSplit100 <- sample(seq\_len(nrow(BreastCancer)), size = splitPercent)  
  
train.df <- BreastCancer[trainSplit100, ]  
valid.df <- BreastCancer[-trainSplit100, ]

library(rpart)  
library(rpart.plot)  
library(e1071)  
library(caret)  
  
#Decision Tree  
tr <- rpart(breast\_cancer\_class ~ ., data = train.df, method = "class")  
prp(tr, type = 1, extra = 1, under = TRUE, split.font = 1, varlen = -10)



#Confusion Matrix for Classification Tree Traning Set  
pred.train.Tree <- predict(tr,train.df,type = "class")  
 confusionMatrix(pred.train.Tree, as.factor(train.df$breast\_cancer\_class))

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction benign malignant  
## benign 259 3  
## malignant 7 140  
##   
## Accuracy : 0.9756   
## 95% CI : (0.9555, 0.9882)  
## No Information Rate : 0.6504   
## P-Value [Acc > NIR] : <2e-16   
##   
## Kappa : 0.9466   
##   
## Mcnemar's Test P-Value : 0.3428   
##   
## Sensitivity : 0.9737   
## Specificity : 0.9790   
## Pos Pred Value : 0.9885   
## Neg Pred Value : 0.9524   
## Prevalence : 0.6504   
## Detection Rate : 0.6333   
## Detection Prevalence : 0.6406   
## Balanced Accuracy : 0.9764   
##   
## 'Positive' Class : benign   
##

#Confusion Matrix for Classification Tree Validation Set  
pred.train.Tree <- predict(tr,valid.df,type = "class")  
 confusionMatrix(pred.train.Tree, as.factor(valid.df$breast\_cancer\_class))

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction benign malignant  
## benign 172 7  
## malignant 6 89  
##   
## Accuracy : 0.9526   
## 95% CI : (0.9202, 0.9745)  
## No Information Rate : 0.6496   
## P-Value [Acc > NIR] : <2e-16   
##   
## Kappa : 0.8955   
##   
## Mcnemar's Test P-Value : 1   
##   
## Sensitivity : 0.9663   
## Specificity : 0.9271   
## Pos Pred Value : 0.9609   
## Neg Pred Value : 0.9368   
## Prevalence : 0.6496   
## Detection Rate : 0.6277   
## Detection Prevalence : 0.6533   
## Balanced Accuracy : 0.9467   
##   
## 'Positive' Class : benign   
##

#Create prediction data frame for ensemble   
TRClass<- data.frame(ifelse(pred.train.Tree=="malignant", "1", "0"))  
  
names(TRClass) <- 'tr'

#Naive Bayes   
nbn <- naiveBayes(breast\_cancer\_class ~ ., data = train.df)  
  
#Predictions and Confusion Matrix for training set  
pred.class <- predict(nbn, newdata = train.df )  
 confusionMatrix(pred.class, train.df$breast\_cancer\_class)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction benign malignant  
## benign 251 3  
## malignant 15 140  
##   
## Accuracy : 0.956   
## 95% CI : (0.9313, 0.9737)  
## No Information Rate : 0.6504   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.9051   
##   
## Mcnemar's Test P-Value : 0.009522   
##   
## Sensitivity : 0.9436   
## Specificity : 0.9790   
## Pos Pred Value : 0.9882   
## Neg Pred Value : 0.9032   
## Prevalence : 0.6504   
## Detection Rate : 0.6137   
## Detection Prevalence : 0.6210   
## Balanced Accuracy : 0.9613   
##   
## 'Positive' Class : benign   
##

#Predictions and Confusion Matrix for validation set  
pred.class <- predict(nbn, newdata = valid.df )  
confusionMatrix(pred.class, valid.df$breast\_cancer\_class)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction benign malignant  
## benign 171 2  
## malignant 7 94  
##   
## Accuracy : 0.9672   
## 95% CI : (0.9386, 0.9849)  
## No Information Rate : 0.6496   
## P-Value [Acc > NIR] : <2e-16   
##   
## Kappa : 0.9287   
##   
## Mcnemar's Test P-Value : 0.1824   
##   
## Sensitivity : 0.9607   
## Specificity : 0.9792   
## Pos Pred Value : 0.9884   
## Neg Pred Value : 0.9307   
## Prevalence : 0.6496   
## Detection Rate : 0.6241   
## Detection Prevalence : 0.6314   
## Balanced Accuracy : 0.9699   
##   
## 'Positive' Class : benign   
##

#Create prediction data frame for ensemble   
NBClass <- data.frame(ifelse(pred.class=="malignant", "1", "0"))  
  
names(NBClass) <- 'nb'

#Logistic Regression   
model <- glm(breast\_cancer\_class ~ ., data = train.df, family = binomial)  
  
#Training Set Predictions and Confusion Matrix  
pred <- predict(model, newdata = train.df, type = "response")  
pred\_class <- ifelse(pred > 0.5, "malignant","benign")  
confusionMatrix(table(pred\_class, train.df$breast\_cancer\_class))

## Confusion Matrix and Statistics  
##   
##   
## pred\_class benign malignant  
## benign 260 6  
## malignant 6 137  
##   
## Accuracy : 0.9707   
## 95% CI : (0.9493, 0.9847)  
## No Information Rate : 0.6504   
## P-Value [Acc > NIR] : <2e-16   
##   
## Kappa : 0.9355   
##   
## Mcnemar's Test P-Value : 1   
##   
## Sensitivity : 0.9774   
## Specificity : 0.9580   
## Pos Pred Value : 0.9774   
## Neg Pred Value : 0.9580   
## Prevalence : 0.6504   
## Detection Rate : 0.6357   
## Detection Prevalence : 0.6504   
## Balanced Accuracy : 0.9677   
##   
## 'Positive' Class : benign   
##

#Validation Set Predictions and Confusion Matrix  
pred <- predict(model, newdata = valid.df, type = "response")  
pred\_class <- ifelse(pred > 0.5, "malignant","benign")  
confusionMatrix(table(pred\_class, valid.df$breast\_cancer\_class))

## Confusion Matrix and Statistics  
##   
##   
## pred\_class benign malignant  
## benign 173 4  
## malignant 5 92  
##   
## Accuracy : 0.9672   
## 95% CI : (0.9386, 0.9849)  
## No Information Rate : 0.6496   
## P-Value [Acc > NIR] : <2e-16   
##   
## Kappa : 0.928   
##   
## Mcnemar's Test P-Value : 1   
##   
## Sensitivity : 0.9719   
## Specificity : 0.9583   
## Pos Pred Value : 0.9774   
## Neg Pred Value : 0.9485   
## Prevalence : 0.6496   
## Detection Rate : 0.6314   
## Detection Prevalence : 0.6460   
## Balanced Accuracy : 0.9651   
##   
## 'Positive' Class : benign   
##

#Create prediction data frame for ensemble   
LGClass<- data.frame(ifelse(pred\_class=="malignant", "1", "0"))  
  
names(LGClass) <- 'lg'

#Neural Network  
library(nnet)  
#Create Model  
nn <- neuralnet(breast\_cancer\_class ~ ., data = train.df, hidden=5, linear.output = F, learningrate = 0.01)  
  
#Training Set Prediction and Confusion Matrix  
pred <- neuralnet::compute(nn, train.df)  
predicted.class=apply(pred$net.result,1,which.max)-1  
confusionMatrix(as.factor(ifelse(predicted.class=="1","malignant", "benign" )), as.factor(train.df$breast\_cancer\_class))

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction benign malignant  
## benign 264 0  
## malignant 2 143  
##   
## Accuracy : 0.9951   
## 95% CI : (0.9824, 0.9994)  
## No Information Rate : 0.6504   
## P-Value [Acc > NIR] : <2e-16   
##   
## Kappa : 0.9893   
##   
## Mcnemar's Test P-Value : 0.4795   
##   
## Sensitivity : 0.9925   
## Specificity : 1.0000   
## Pos Pred Value : 1.0000   
## Neg Pred Value : 0.9862   
## Prevalence : 0.6504   
## Detection Rate : 0.6455   
## Detection Prevalence : 0.6455   
## Balanced Accuracy : 0.9962   
##   
## 'Positive' Class : benign   
##

#Validation Set Prediction and Confusion Matrix  
pred <- neuralnet::compute(nn, valid.df)  
predicted.class=apply(pred$net.result,1,which.max)-1  
confusionMatrix(as.factor(ifelse(predicted.class=="1", "malignant", "benign")), as.factor(valid.df$breast\_cancer\_class))

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction benign malignant  
## benign 175 4  
## malignant 3 92  
##   
## Accuracy : 0.9745   
## 95% CI : (0.9481, 0.9897)  
## No Information Rate : 0.6496   
## P-Value [Acc > NIR] : <2e-16   
##   
## Kappa : 0.9437   
##   
## Mcnemar's Test P-Value : 1   
##   
## Sensitivity : 0.9831   
## Specificity : 0.9583   
## Pos Pred Value : 0.9777   
## Neg Pred Value : 0.9684   
## Prevalence : 0.6496   
## Detection Rate : 0.6387   
## Detection Prevalence : 0.6533   
## Balanced Accuracy : 0.9707   
##   
## 'Positive' Class : benign   
##

mynnet.pred <- predict(nn,train.df,type="class")  
  
#Create prediction data frame for ensemble   
nnClass <- data.frame(predicted.class)  
names(nnClass) <- 'nn'

#Ensemble Construction   
  
#combine all predictive results which are based off of the validation data sets for all 4 models  
combPRED <- mutate\_all(data.frame(nnClass, TRClass, LGClass, NBClass), as.integer)  
  
combPRED$PREDICTION <- (combPRED$nn + combPRED$tr + combPRED$lg + combPRED$nb)  
  
#vote for classification   
combPRED$Class <- (ifelse(combPRED$PREDICTION > 2,"malignant","benign"))  
  
table(combPRED$Class)

##   
## benign malignant   
## 179 95

#confusion matrix for performance statistics   
confusionMatrix(as.factor(combPRED$Class), valid.df$breast\_cancer\_class)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction benign malignant  
## benign 175 4  
## malignant 3 92  
##   
## Accuracy : 0.9745   
## 95% CI : (0.9481, 0.9897)  
## No Information Rate : 0.6496   
## P-Value [Acc > NIR] : <2e-16   
##   
## Kappa : 0.9437   
##   
## Mcnemar's Test P-Value : 1   
##   
## Sensitivity : 0.9831   
## Specificity : 0.9583   
## Pos Pred Value : 0.9777   
## Neg Pred Value : 0.9684   
## Prevalence : 0.6496   
## Detection Rate : 0.6387   
## Detection Prevalence : 0.6533   
## Balanced Accuracy : 0.9707   
##   
## 'Positive' Class : benign   
##