ProjectTwo

Gauri Patil

3/15/2021

Load library

library(rpart)  
library(rpart.plot)

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

library(mlbench)

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

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 4.0.3

library(caret)

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

## Loading required package: lattice

## Loading required package: ggplot2

## Warning: package 'ggplot2' 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

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

library(e1071)

## Warning: package 'e1071' was built under R version 4.0.4

library(nnet)

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

library(klaR)  
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(varhandle)

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

Load Data

data("BreastCancer")  
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   
##   
##   
##   
##   
##

Clean Data

#remove rows with missing values  
BreastCancer <- na.omit(BreastCancer)   
# remove the unique identifier  
BreastCancer$Id <- NULL

Partition Data

#Create partition  
index <- sample(2, nrow(BreastCancer), replace = TRUE, prob=c(0.8, 0.2))  
training <- BreastCancer[index == 1,]  
testing <- BreastCancer[index == 2,]

Recursive Partitioning and Regression Tree

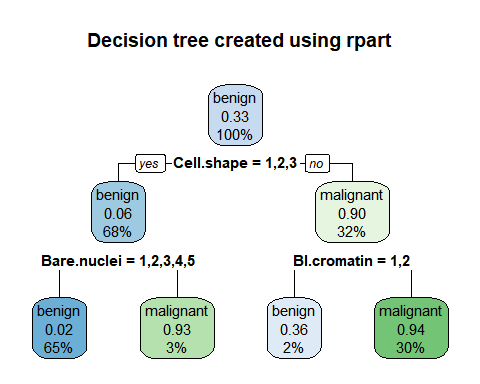
# create model using recursive partitioning on the training data set  
x.rp <- rpart(Class ~ ., data=training, method = "class")  
  
summary(x.rp)

## Call:  
## rpart(formula = Class ~ ., data = training, method = "class")  
## n= 554   
##   
## CP nsplit rel error xerror xstd  
## 1 0.78260870 0 1.0000000 1.0000000 0.06024723  
## 2 0.07065217 1 0.2173913 0.2717391 0.03665452  
## 3 0.01630435 2 0.1467391 0.1576087 0.02849089  
## 4 0.01000000 3 0.1304348 0.1521739 0.02802200  
##   
## Variable importance  
## Cell.shape Cell.size Bare.nuclei Bl.cromatin Normal.nucleoli   
## 21 18 16 14 14   
## Epith.c.size Cl.thickness Marg.adhesion Mitoses   
## 14 1 1 1   
##   
## Node number 1: 554 observations, complexity param=0.7826087  
## predicted class=benign expected loss=0.33213 P(node) =1  
## class counts: 370 184  
## probabilities: 0.668 0.332   
## left son=2 (376 obs) right son=3 (178 obs)  
## Primary splits:  
## Cell.shape splits as LLLRRRRRRR, improve=171.8372, (0 missing)  
## Cell.size splits as LLLRRRRRRR, improve=171.6111, (0 missing)  
## Bare.nuclei splits as LLRRRRRRRR, improve=162.5323, (0 missing)  
## Bl.cromatin splits as LLLRRRRRRR, improve=154.8021, (0 missing)  
## Epith.c.size splits as LLRRRRRRRR, improve=146.7508, (0 missing)  
## Surrogate splits:  
## Cell.size splits as LLLRRRRRRR, agree=0.946, adj=0.831, (0 split)  
## Epith.c.size splits as LLLRRRRRRR, agree=0.886, adj=0.646, (0 split)  
## Bare.nuclei splits as LLLRRLRRRR, agree=0.883, adj=0.635, (0 split)  
## Normal.nucleoli splits as LLRRRRRRRR, agree=0.879, adj=0.624, (0 split)  
## Bl.cromatin splits as LLLRRRRRRR, agree=0.877, adj=0.618, (0 split)  
##   
## Node number 2: 376 observations, complexity param=0.07065217  
## predicted class=benign expected loss=0.06117021 P(node) =0.6787004  
## class counts: 353 23  
## probabilities: 0.939 0.061   
## left son=4 (361 obs) right son=5 (15 obs)  
## Primary splits:  
## Bare.nuclei splits as LLLLLRRR-R, improve=23.76826, (0 missing)  
## Bl.cromatin splits as LLLRRRRR--, improve=18.79988, (0 missing)  
## Cl.thickness splits as LLLLLLLLRR, improve=18.10967, (0 missing)  
## Normal.nucleoli splits as LLRRRRRR-R, improve=14.21960, (0 missing)  
## Mitoses splits as LRRRL-L-R, improve=13.11478, (0 missing)  
## Surrogate splits:  
## Cl.thickness splits as LLLLLLLLRR, agree=0.976, adj=0.400, (0 split)  
## Marg.adhesion splits as LLLLRRRRRR, agree=0.973, adj=0.333, (0 split)  
## Normal.nucleoli splits as LLLRLRRL-L, agree=0.968, adj=0.200, (0 split)  
## Mitoses splits as LLLRL-L-L, agree=0.968, adj=0.200, (0 split)  
## Cell.size splits as LLLLLRRRRR, agree=0.963, adj=0.067, (0 split)  
##   
## Node number 3: 178 observations, complexity param=0.01630435  
## predicted class=malignant expected loss=0.09550562 P(node) =0.3212996  
## class counts: 17 161  
## probabilities: 0.096 0.904   
## left son=6 (11 obs) right son=7 (167 obs)  
## Primary splits:  
## Bl.cromatin splits as LLRRRRRRRR, improve=6.859505, (0 missing)  
## Cell.size splits as LLLLRRRRRR, improve=6.189347, (0 missing)  
## Bare.nuclei splits as LRRRRRRRRR, improve=4.876420, (0 missing)  
## Cell.shape splits as LLLLRRRRRR, improve=4.582694, (0 missing)  
## Cl.thickness splits as LLLLLRRRRR, improve=3.438685, (0 missing)  
## Surrogate splits:  
## Cell.size splits as LRRRRRRRRR, agree=0.955, adj=0.273, (0 split)  
##   
## Node number 4: 361 observations  
## predicted class=benign expected loss=0.02493075 P(node) =0.6516245  
## class counts: 352 9  
## probabilities: 0.975 0.025   
##   
## Node number 5: 15 observations  
## predicted class=malignant expected loss=0.06666667 P(node) =0.02707581  
## class counts: 1 14  
## probabilities: 0.067 0.933   
##   
## Node number 6: 11 observations  
## predicted class=benign expected loss=0.3636364 P(node) =0.0198556  
## class counts: 7 4  
## probabilities: 0.636 0.364   
##   
## Node number 7: 167 observations  
## predicted class=malignant expected loss=0.05988024 P(node) =0.301444  
## class counts: 10 157  
## probabilities: 0.060 0.940

# predict classes for the evaluation data set  
x.rp.pred <- predict(x.rp, type="class", newdata=testing)  
  
# score the evaluation data set (extract the probabilities)  
x.rp.prob <- predict(x.rp, type="prob", newdata=testing)  
  
#confusion matrix  
confusionMatrix(x.rp.pred,testing$Class)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction benign malignant  
## benign 71 6  
## malignant 3 49  
##   
## Accuracy : 0.9302   
## 95% CI : (0.8717, 0.9676)  
## No Information Rate : 0.5736   
## P-Value [Acc > NIR] : <2e-16   
##   
## Kappa : 0.8564   
##   
## Mcnemar's Test P-Value : 0.505   
##   
## Sensitivity : 0.9595   
## Specificity : 0.8909   
## Pos Pred Value : 0.9221   
## Neg Pred Value : 0.9423   
## Prevalence : 0.5736   
## Detection Rate : 0.5504   
## Detection Prevalence : 0.5969   
## Balanced Accuracy : 0.9252   
##   
## 'Positive' Class : benign   
##

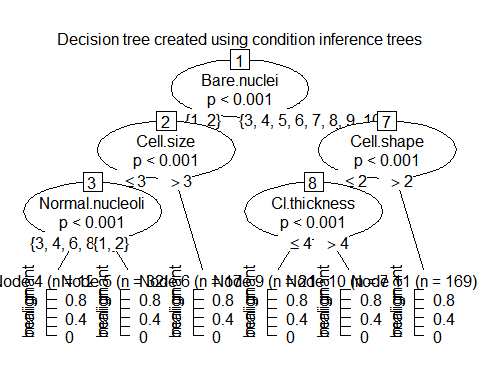
# decision tree  
rpart.plot(x.rp, main="Decision tree created using rpart")

 Conditional Inference Tree

# create model using conditional inference trees  
x.ct <- ctree(Class ~ ., data=training)  
# predict classes for the evaluation data set  
x.ct.pred <- predict(x.ct, newdata=testing)  
# score the evaluation data set (extract the probabilities)  
x.ct.prob <- 1- unlist(treeresponse(x.ct, testing), use.names=F)[seq(1,nrow(testing)\*2,2)]  
#confusion matrix  
confusionMatrix(x.ct.pred,testing$Class)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction benign malignant  
## benign 71 0  
## malignant 3 55  
##   
## Accuracy : 0.9767   
## 95% CI : (0.9335, 0.9952)  
## No Information Rate : 0.5736   
## P-Value [Acc > NIR] : <2e-16   
##   
## Kappa : 0.9528   
##   
## Mcnemar's Test P-Value : 0.2482   
##   
## Sensitivity : 0.9595   
## Specificity : 1.0000   
## Pos Pred Value : 1.0000   
## Neg Pred Value : 0.9483   
## Prevalence : 0.5736   
## Detection Rate : 0.5504   
## Detection Prevalence : 0.5504   
## Balanced Accuracy : 0.9797   
##   
## 'Positive' Class : benign   
##

# decision tree  
plot(x.ct, main="Decision tree created using condition inference trees")



Cforest forest

# create model using random forest and bagging ensemble using conditional inference trees  
x.cf <- cforest(Class ~ ., data=training, control = cforest\_unbiased(mtry = ncol(BreastCancer)-2))  
# predict classes for the evaluation data set  
x.cf.pred <- predict(x.cf, newdata=testing)  
# score the evaluation data set (extract the probabilities)  
x.cf.prob <- 1- unlist(treeresponse(x.cf, testing), use.names=F)[seq(1,nrow(testing)\*2,2)]  
#confusion matrix  
confusionMatrix(x.cf.pred,testing$Class)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction benign malignant  
## benign 73 0  
## malignant 1 55  
##   
## Accuracy : 0.9922   
## 95% CI : (0.9576, 0.9998)  
## No Information Rate : 0.5736   
## P-Value [Acc > NIR] : <2e-16   
##   
## Kappa : 0.9842   
##   
## Mcnemar's Test P-Value : 1   
##   
## Sensitivity : 0.9865   
## Specificity : 1.0000   
## Pos Pred Value : 1.0000   
## Neg Pred Value : 0.9821   
## Prevalence : 0.5736   
## Detection Rate : 0.5659   
## Detection Prevalence : 0.5659   
## Balanced Accuracy : 0.9932   
##   
## 'Positive' Class : benign   
##

Naive Bayes Model

# create model using Naive Bayes   
x.nb <- NaiveBayes(Class ~ ., data=training)  
  
# predict classes for the evaluation data set  
x.nb.pred <- predict(x.nb, newdata=testing)  
  
#confusion matrix  
confusionMatrix(x.nb.pred$class,testing$Class)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction benign malignant  
## benign 73 0  
## malignant 1 55  
##   
## Accuracy : 0.9922   
## 95% CI : (0.9576, 0.9998)  
## No Information Rate : 0.5736   
## P-Value [Acc > NIR] : <2e-16   
##   
## Kappa : 0.9842   
##   
## Mcnemar's Test P-Value : 1   
##   
## Sensitivity : 0.9865   
## Specificity : 1.0000   
## Pos Pred Value : 1.0000   
## Neg Pred Value : 0.9821   
## Prevalence : 0.5736   
## Detection Rate : 0.5659   
## Detection Prevalence : 0.5659   
## Balanced Accuracy : 0.9932   
##   
## 'Positive' Class : benign   
##

Support Vector Machine Classifier

# create model using svm  
x.svm <- svm(Class ~ ., data=training)  
  
# predict classes for the evaluation data set  
x.svm.pred <- predict(x.svm, newdata=testing)  
  
#confusion matrix  
confusionMatrix(x.svm.pred,testing$Class)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction benign malignant  
## benign 73 1  
## malignant 1 54  
##   
## Accuracy : 0.9845   
## 95% CI : (0.9451, 0.9981)  
## No Information Rate : 0.5736   
## P-Value [Acc > NIR] : <2e-16   
##   
## Kappa : 0.9683   
##   
## Mcnemar's Test P-Value : 1   
##   
## Sensitivity : 0.9865   
## Specificity : 0.9818   
## Pos Pred Value : 0.9865   
## Neg Pred Value : 0.9818   
## Prevalence : 0.5736   
## Detection Rate : 0.5659   
## Detection Prevalence : 0.5736   
## Balanced Accuracy : 0.9842   
##   
## 'Positive' Class : benign   
##

Neural Net Model

# create model using neural net  
x.nnet <- nnet(Class ~ ., training, size=1)

## # weights: 83  
## initial value 357.897181   
## iter 10 value 77.142557  
## iter 20 value 50.386866  
## iter 30 value 44.042379  
## iter 40 value 40.002934  
## iter 50 value 39.013255  
## iter 60 value 38.980280  
## iter 70 value 29.983648  
## iter 80 value 27.542725  
## iter 90 value 26.260539  
## iter 100 value 26.247058  
## final value 26.247058   
## stopped after 100 iterations

# predict classes for the evaluation data set  
x.nnet.pred <- predict(x.nnet,testing,type="class")  
#confusion matrix  
confusionMatrix(as.factor(x.nnet.pred),testing$Class)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction benign malignant  
## benign 72 3  
## malignant 2 52  
##   
## Accuracy : 0.9612   
## 95% CI : (0.9119, 0.9873)  
## No Information Rate : 0.5736   
## P-Value [Acc > NIR] : <2e-16   
##   
## Kappa : 0.9206   
##   
## Mcnemar's Test P-Value : 1   
##   
## Sensitivity : 0.9730   
## Specificity : 0.9455   
## Pos Pred Value : 0.9600   
## Neg Pred Value : 0.9630   
## Prevalence : 0.5736   
## Detection Rate : 0.5581   
## Detection Prevalence : 0.5814   
## Balanced Accuracy : 0.9592   
##   
## 'Positive' Class : benign   
##

Cross Validation R tree

# Leave-1-Out Cross Validation (LOOCV)  
x.cv.pred <- numeric(length(testing[,10]))  
for (i in 1:length(training[,10])) {  
 x1.tree <- rpart(Class ~ ., training[-i,])  
}  
  
for (i in 1:length(testing[,10])) {  
 x1.pred <- predict(x1.tree,testing[i,],type="class")  
 x.cv.pred[i] <- x1.pred  
}  
  
  
x.cv.pred <- factor(x.cv.pred,labels=levels(testing$Class))  
confusionMatrix(x.cv.pred,testing$Class)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction benign malignant  
## benign 71 6  
## malignant 3 49  
##   
## Accuracy : 0.9302   
## 95% CI : (0.8717, 0.9676)  
## No Information Rate : 0.5736   
## P-Value [Acc > NIR] : <2e-16   
##   
## Kappa : 0.8564   
##   
## Mcnemar's Test P-Value : 0.505   
##   
## Sensitivity : 0.9595   
## Specificity : 0.8909   
## Pos Pred Value : 0.9221   
## Neg Pred Value : 0.9423   
## Prevalence : 0.5736   
## Detection Rate : 0.5504   
## Detection Prevalence : 0.5969   
## Balanced Accuracy : 0.9252   
##   
## 'Positive' Class : benign   
##

QDA

training1<-training  
for (i in 1:ncol(training1)){  
 training1[,i] <- as.integer(training1[,i])  
}  
  
testing1<-testing  
for (i in 1:ncol(testing1)){  
 testing1[,i] <- as.integer(testing1[,i])  
}  
  
#convert class to binary  
training1$Class<-ifelse(as.integer(training1$Class)==2,1,0)  
testing1$Class<-ifelse(as.integer(testing1$Class)==2,1,0)  
#Run QDA model  
x.qda <- qda(Class ~ ., data=training1)  
#predict using training dataset   
x.qda.pred <- predict(x.qda, testing1)  
#confusion matrix  
confusionMatrix(x.qda.pred$class,as.factor(testing1$Class))

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 69 1  
## 1 5 54  
##   
## Accuracy : 0.9535   
## 95% CI : (0.9015, 0.9827)  
## No Information Rate : 0.5736   
## P-Value [Acc > NIR] : <2e-16   
##   
## Kappa : 0.9058   
##   
## Mcnemar's Test P-Value : 0.2207   
##   
## Sensitivity : 0.9324   
## Specificity : 0.9818   
## Pos Pred Value : 0.9857   
## Neg Pred Value : 0.9153   
## Prevalence : 0.5736   
## Detection Rate : 0.5349   
## Detection Prevalence : 0.5426   
## Balanced Accuracy : 0.9571   
##   
## 'Positive' Class : 0   
##

Regularised Discriminant Analysis

#Build model  
x.rda <- rda(Class ~ ., training)  
#Predict using validation data  
x.rda.pred <- predict(x.rda, testing)  
#Confusion matrix  
confusionMatrix(x.rda.pred$class,testing$Class)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction benign malignant  
## benign 71 0  
## malignant 3 55  
##   
## Accuracy : 0.9767   
## 95% CI : (0.9335, 0.9952)  
## No Information Rate : 0.5736   
## P-Value [Acc > NIR] : <2e-16   
##   
## Kappa : 0.9528   
##   
## Mcnemar's Test P-Value : 0.2482   
##   
## Sensitivity : 0.9595   
## Specificity : 1.0000   
## Pos Pred Value : 1.0000   
## Neg Pred Value : 0.9483   
## Prevalence : 0.5736   
## Detection Rate : 0.5504   
## Detection Prevalence : 0.5504   
## Balanced Accuracy : 0.9797   
##   
## 'Positive' Class : benign   
##

Random Forests

#Build model  
x.rf <- randomForest(Class ~ ., training)  
#Predict using validation data  
x.rf.pred <- predict(x.rf, testing)  
#confusion matrix  
confusionMatrix(x.rf.pred, testing$Class)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction benign malignant  
## benign 73 0  
## malignant 1 55  
##   
## Accuracy : 0.9922   
## 95% CI : (0.9576, 0.9998)  
## No Information Rate : 0.5736   
## P-Value [Acc > NIR] : <2e-16   
##   
## Kappa : 0.9842   
##   
## Mcnemar's Test P-Value : 1   
##   
## Sensitivity : 0.9865   
## Specificity : 1.0000   
## Pos Pred Value : 1.0000   
## Neg Pred Value : 0.9821   
## Prevalence : 0.5736   
## Detection Rate : 0.5659   
## Detection Prevalence : 0.5659   
## Balanced Accuracy : 0.9932   
##   
## 'Positive' Class : benign   
##

Ensemble output using models

#Add all prediction in a table  
predall<-data.frame(x.rp.pred,x.ct.pred,x.cf.pred,x.nb.pred$class,x.svm.pred,x.nnet.pred,x.cv.pred,x.rda.pred$class,x.rf.pred)  
#convert to binary  
for (i in 1:ncol(predall)){  
 predall[,i] <- ifelse(predall[,i]=="benign",0,1)  
}  
#unfactor to calcuate rowsums  
predall$x.qda.pred<-unfactor(x.qda.pred$class)  
#calculate rowsums  
predall$sum<-rowSums(predall)  
#Majority Rule   
predall$majority<-ifelse(predall$sum>=6, "malignant", "benign")  
#Confusion matrix   
confusionMatrix(as.factor(predall$majority), testing$Class)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction benign malignant  
## benign 73 1  
## malignant 1 54  
##   
## Accuracy : 0.9845   
## 95% CI : (0.9451, 0.9981)  
## No Information Rate : 0.5736   
## P-Value [Acc > NIR] : <2e-16   
##   
## Kappa : 0.9683   
##   
## Mcnemar's Test P-Value : 1   
##   
## Sensitivity : 0.9865   
## Specificity : 0.9818   
## Pos Pred Value : 0.9865   
## Neg Pred Value : 0.9818   
## Prevalence : 0.5736   
## Detection Rate : 0.5659   
## Detection Prevalence : 0.5736   
## Balanced Accuracy : 0.9842   
##   
## 'Positive' Class : benign   
##