project 2

CanPeng

3/18/2021

require(mlbench)

## Loading required package: 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 ggplot2 3.3.2 v purrr 0.3.4  
## v tibble 3.0.3 v dplyr 1.0.2  
## v tidyr 1.1.2 v stringr 1.4.0  
## v readr 1.3.1 v forcats 0.5.0

## -- Conflicts ------------------------------------------------------------------------------------------------------- tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

### Data preparation

#Load Data

# load the data set  
data(BreastCancer)  
# some algorithms don't like missing values, so remove rows with missing values  
BreastCancer <- na.omit(BreastCancer)   
# remove the unique identifier, which is useless and would confuse the machine learning algorithms  
BreastCancer$Id <- NULL   
# check the class of each varibles  
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 ...  
## - attr(\*, "na.action")= 'omit' Named int [1:16] 24 41 140 146 159 165 236 250 276 293 ...  
## ..- attr(\*, "names")= chr [1:16] "24" "41" "140" "146" ...

#partition the data

# partition the data set for 80% training and 20% evaluation  
set.seed(2)  
ind <- sample(2, nrow(BreastCancer), replace = TRUE, prob=c(0.8, 0.2))  
train <- BreastCancer[ind==1,]  
valid <- BreastCancer[ind==2,]

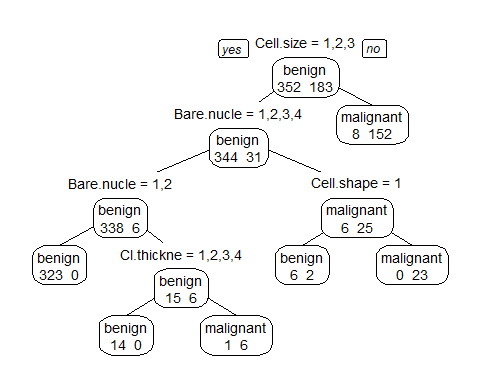
### Create multiple models using different classifiers/algorithms

#1 Decision trees

library(rpart)  
library(rpart.plot)

## Warning: package 'rpart.plot' was built under R version 4.0.4

x.rp <- rpart(Class ~ ., data=train)  
#plot(x.rp, main="Decision tree created using rpart")   
prp(x.rp, type = 1, extra = 1, split.font = 1, varlen = -10)



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

##   
## x.rp.pred benign malignant  
## benign 86 2  
## malignant 6 54

# Leave-1-Out Cross Validation (LOOCV)

ans <- numeric(nrow(BreastCancer))  
for (i in 1:nrow(BreastCancer)) {  
 mytree <- rpart(Class ~ ., BreastCancer[-i,])  
 mytree.pred <- predict(mytree,BreastCancer[i,],type="class")  
 ans[i] <- mytree.pred  
}  
ans <- factor(ans,labels=levels(BreastCancer$Class))  
table(ans,BreastCancer$Class)

##   
## ans benign malignant  
## benign 430 20  
## malignant 14 219

#2 condition inference trees # create model using conditional inference trees

require(party)

## Loading required package: 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

##   
## Attaching package: 'zoo'

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

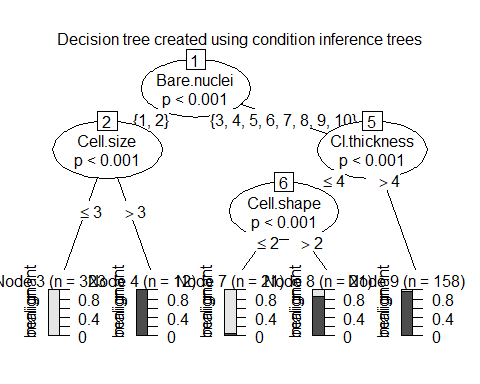
## Loading required package: sandwich

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

##   
## Attaching package: 'strucchange'

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

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



x.ct.pred <- predict(x.ct, newdata=valid) #ensemble  
x.ct.prob <- 1- unlist(treeresponse(x.ct, valid), use.names=F)[seq(1,nrow(valid)\*2,2)]  
table(x.ct.pred,valid$Class)

##   
## x.ct.pred benign malignant  
## benign 86 2  
## malignant 6 54

#3 Random Forests

x.cf <- cforest(Class ~ ., data=train, control = cforest\_unbiased(mtry = 9))   
  
x.cf.pred <- predict(x.cf, newdata=valid)  
x.cf.prob <- 1- unlist(treeresponse(x.cf, valid), use.names=F)[seq(1,nrow(valid)\*2,2)]  
table(x.cf.pred,valid$Class)

##   
## x.cf.pred benign malignant  
## benign 86 2  
## malignant 6 54

#4 bagging # create model using bagging (bootstrap aggregating)

require(ipred)

## Loading required package: ipred

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

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

##   
## x.ip.pred benign malignant  
## benign 87 2  
## malignant 5 54

#5.SVM # create model using svm (support vector machine)

require(e1071)

## Loading required package: e1071

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

# svm requires tuning  
x.svm.tune <- tune(svm, Class~., 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  
## 0.0625 1  
##   
## - best performance: 0.01675978

# 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(Class~., data = train, cost=1, gamma=0.03125, probability = TRUE)   
x.svm.pred <- predict(x.svm, type="class", newdata=valid)  
x.svm.prob <- predict(x.svm, type="prob", newdata=valid, probability = TRUE)  
table(x.svm.pred,valid$Class)

##   
## x.svm.pred benign malignant  
## benign 87 1  
## malignant 5 55

#7. neural network

library(nnet)  
x.nn <- nnet(Class~., data = train,size=1)

## # weights: 83  
## initial value 345.642770   
## iter 10 value 36.837980  
## iter 20 value 24.123706  
## iter 30 value 18.389177  
## iter 40 value 17.965373  
## iter 50 value 17.912767  
## iter 60 value 17.897303  
## iter 70 value 17.893276  
## iter 80 value 17.892504  
## iter 90 value 17.892226  
## iter 100 value 17.892029  
## final value 17.892029   
## stopped after 100 iterations

x.nn.pred <- predict(x.nn,valid,type="class")  
x.nn.prob <- predict(x.nn,valid,type="raw")  
table(x.nn.pred,valid$Class)

##   
## x.nn.pred benign malignant  
## benign 86 2  
## malignant 6 54

#8 QDA #Quadratic Discriminant Analysis

library(MASS)

##   
## Attaching package: 'MASS'

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

library(dplyr)  
train.num <- train %>% dplyr::select(-Class) %>% mutate\_if(is.factor,as.character)%>% mutate\_if(is.character,as.numeric) #change the class  
train.num$Class <- train$Class  
valid.num <- valid%>%dplyr::select(-Class) %>% mutate\_if(is.factor,as.character)%>% mutate\_if(is.character,as.numeric) #change the class  
valid.num$Class <- valid$Class  
str(train.num)

## 'data.frame': 535 obs. of 10 variables:  
## $ Cl.thickness : num 5 5 3 6 1 2 4 1 2 5 ...  
## $ Cell.size : num 1 4 1 8 1 1 2 1 1 3 ...  
## $ Cell.shape : num 1 4 1 8 1 1 1 1 1 3 ...  
## $ Marg.adhesion : num 1 5 1 1 1 1 1 1 1 3 ...  
## $ Epith.c.size : num 2 7 2 3 2 2 2 1 2 2 ...  
## $ Bare.nuclei : num 1 10 2 4 10 1 1 1 1 3 ...  
## $ Bl.cromatin : num 3 3 3 3 3 1 2 3 2 4 ...  
## $ Normal.nucleoli: num 1 2 1 7 1 1 1 1 1 4 ...  
## $ Mitoses : num 1 1 1 1 1 5 1 1 1 1 ...  
## $ Class : Factor w/ 2 levels "benign","malignant": 1 1 1 1 1 1 1 1 1 2 ...

x.qda <- qda(Class~., data = train.num)   
x.qda.pred <- predict(x.qda, valid.num)$class  
table(x.qda.pred,valid.num$Class)

##   
## x.qda.pred benign malignant  
## benign 86 1  
## malignant 6 55

#9 RDA #Regularised Discriminant Analysis

library(klaR)

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

x.rda <- rda(Class~., data = train)  
x.rda.pred <- predict(x.rda, valid)$class  
table(x.rda.pred,valid$Class)

##   
## x.rda.pred benign malignant  
## benign 87 0  
## malignant 5 56

## ROC

## plot ROC curves to compare the performance of the individual classifiers

# Output the plot to a PNG file for display on web. To draw to the screen,

# comment this line out.

#png(filename=“roc\_curve\_5\_models.png”, width=700, height=700)

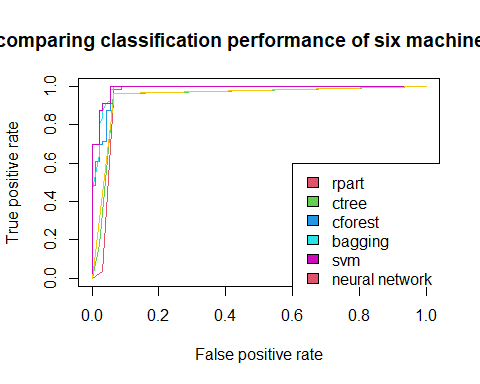
#load the ROCR package which draws the ROC curves  
require(ROCR)

## Loading required package: ROCR

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

# rptree  
# create an ROCR prediction object from rpart() probabilities  
x.rp.prob.rocr <- prediction(x.rp.prob[,2], BreastCancer[ind == 2,'Class'])  
# prepare an ROCR performance object for ROC curve (tpr=true positive rate, fpr=false positive rate)  
x.rp.perf <- performance(x.rp.prob.rocr, "tpr","fpr")  
  
# ctree  
x.ct.prob.rocr <- prediction(x.ct.prob, BreastCancer[ind == 2,'Class'])  
x.ct.perf <- performance(x.ct.prob.rocr, "tpr","fpr")  
  
# cforest  
x.cf.prob.rocr <- prediction(x.cf.prob, BreastCancer[ind == 2,'Class'])  
x.cf.perf <- performance(x.cf.prob.rocr, "tpr","fpr")  
  
# bagging  
x.ip.prob.rocr <- prediction(x.ip.prob[,2], BreastCancer[ind == 2,'Class'])  
x.ip.perf <- performance(x.ip.prob.rocr, "tpr","fpr")  
  
# svm  
x.svm.prob.rocr <- prediction(attr(x.svm.prob, "probabilities")[,2], BreastCancer[ind == 2,'Class'])  
x.svm.perf <- performance(x.svm.prob.rocr, "tpr","fpr")  
  
# nn  
x.nn.prob.rocr <- prediction(x.nn.prob, BreastCancer[ind == 2,'Class'])  
x.nn.perf <- performance(x.nn.prob.rocr, "tpr","fpr")

# plot it  
plot(x.rp.perf, col=2, main="ROC curves comparing classification performance of six machine learning models")   
# Draw a legend  
legend(0.6, 0.6, c('rpart', 'ctree', 'cforest','bagging','svm', 'neural network'),2:6)  
plot(x.ct.perf, col=3, add=TRUE)  
plot(x.cf.perf, col=4, add=TRUE)  
plot(x.ip.perf, col=5, add=TRUE)  
plot(x.svm.perf, col=6, add=TRUE)  
plot(x.nn.perf, col=7, add=TRUE)



### Ensemble: creating an ensemble for combining all classifiers

classifier <- data.frame(cbind(x.rp.pred, x.ct.pred, x.cf.pred, x.ip.pred, x.svm.pred ,x.nn.pred))  
  
names(classifier) <-c('recursive.tree','conditional.inference.tree','random.forest','bootstrap','svm','neutral.network')  
levels(classifier$neutral.network) =c('1','2')  
  
classifier <-classifier%>% sapply(FUN = function(x)(ifelse(x=='1',0,1)))  
classifier<- addmargins(classifier, margin = 2) # table/arragy, margin =2 aggregate by col   
classifier <- data.frame(classifier)  
classifier$predition <- ifelse(classifier$Sum >=5, 'malignant','benign')  
head(classifier)

## recursive.tree conditional.inference.tree random.forest bootstrap svm  
## 1 0 0 0 0 0  
## 2 1 1 1 1 1  
## 3 0 0 0 0 0  
## 4 1 1 1 1 1  
## 5 0 0 0 0 0  
## 6 0 0 0 0 0  
## neutral.network Sum predition  
## 1 1 1 benign  
## 2 1 6 malignant  
## 3 1 1 benign  
## 4 1 6 malignant  
## 5 1 1 benign  
## 6 1 1 benign

library(ggplot2)  
table(classifier$predition, valid$Class)

##   
## benign malignant  
## benign 87 2  
## malignant 5 54

#confusion matrix

library(caret)

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

## Loading required package: lattice

##   
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':  
##   
## lift

confusionMatrix(as.factor(classifier$predition), valid$Class, positive = 'malignant')

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction benign malignant  
## benign 87 2  
## malignant 5 54  
##   
## Accuracy : 0.9527   
## 95% CI : (0.905, 0.9808)  
## No Information Rate : 0.6216   
## P-Value [Acc > NIR] : <2e-16   
##   
## Kappa : 0.9005   
##   
## Mcnemar's Test P-Value : 0.4497   
##   
## Sensitivity : 0.9643   
## Specificity : 0.9457   
## Pos Pred Value : 0.9153   
## Neg Pred Value : 0.9775   
## Prevalence : 0.3784   
## Detection Rate : 0.3649   
## Detection Prevalence : 0.3986   
## Balanced Accuracy : 0.9550   
##   
## 'Positive' Class : malignant   
##