Project\_Two\_560

Peter

3/15/2021

[anhphan1101/peterphan.github.io: Website to deploy Week12 Model](https://github.com/anhphan1101/peterphan.github.io)

require(mlbench) #package which as dataset- Breastcancer  
library(tidyverse) #for datamining  
library(ggcorrplot)   
library(GGally)  
library(randomForest)  
library(e1071)   
library(ROCR)  
library(pROC)  
library(RCurl)  
library(MASS)  
library(caTools) #forsplit dataset into train test  
library(caret)  
library(party)  
library(rpart)  
library(rpart.plot)  
library(ipred)  
library(e1071)

BreastCancer data-set consists of a sample of patients reported to Dr.Wolberg. The objective is to predict whether a new patient has a malignant tumor from a set of predicting variables.It has 699 observations on 11 variables, one being a character variable, 9 being ordered or nominal, and 1 target class.

###A- Data Understanding

# load the data set  
data(BreastCancer)  
# Remove NA- in this case using omit  
Breastcancer <- na.omit(BreastCancer)   
# remove the unique identifier  
Breastcancer$Id<- NULL  
# Convert input values to numeric  
for(i in 1:9) {  
 Breastcancer[,i] <- as.numeric(as.character(Breastcancer[,i]))  
}  
  
summary(Breastcancer)

## Cl.thickness Cell.size Cell.shape Marg.adhesion   
## Min. : 1.000 Min. : 1.000 Min. : 1.000 Min. : 1.00   
## 1st Qu.: 2.000 1st Qu.: 1.000 1st Qu.: 1.000 1st Qu.: 1.00   
## Median : 4.000 Median : 1.000 Median : 1.000 Median : 1.00   
## Mean : 4.442 Mean : 3.151 Mean : 3.215 Mean : 2.83   
## 3rd Qu.: 6.000 3rd Qu.: 5.000 3rd Qu.: 5.000 3rd Qu.: 4.00   
## Max. :10.000 Max. :10.000 Max. :10.000 Max. :10.00   
## Epith.c.size Bare.nuclei Bl.cromatin Normal.nucleoli  
## Min. : 1.000 Min. : 1.000 Min. : 1.000 Min. : 1.00   
## 1st Qu.: 2.000 1st Qu.: 1.000 1st Qu.: 2.000 1st Qu.: 1.00   
## Median : 2.000 Median : 1.000 Median : 3.000 Median : 1.00   
## Mean : 3.234 Mean : 3.545 Mean : 3.445 Mean : 2.87   
## 3rd Qu.: 4.000 3rd Qu.: 6.000 3rd Qu.: 5.000 3rd Qu.: 4.00   
## Max. :10.000 Max. :10.000 Max. :10.000 Max. :10.00   
## Mitoses Class   
## Min. : 1.000 benign :444   
## 1st Qu.: 1.000 malignant:239   
## Median : 1.000   
## Mean : 1.603   
## 3rd Qu.: 1.000   
## Max. :10.000

# Data types  
sapply(Breastcancer, class)

## Cl.thickness Cell.size Cell.shape Marg.adhesion Epith.c.size   
## "numeric" "numeric" "numeric" "numeric" "numeric"   
## Bare.nuclei Bl.cromatin Normal.nucleoli Mitoses Class   
## "numeric" "numeric" "numeric" "numeric" "factor"

###A- Data Understanding #1-Class distribution

cbind(freq=table(Breastcancer$Class), percentage=prop.table(table(Breastcancer$Class))\*100)

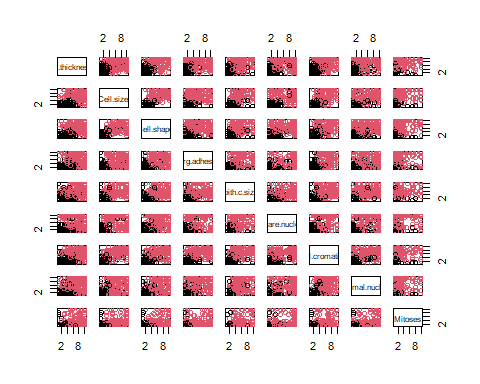
## freq percentage  
## benign 444 65.00732  
## malignant 239 34.99268

#1- Correlation & 9 Attributes’Desity plot

cor(Breastcancer[1:9])

## Cl.thickness Cell.size Cell.shape Marg.adhesion Epith.c.size  
## Cl.thickness 1.0000000 0.6424815 0.6534700 0.4878287 0.5235960  
## Cell.size 0.6424815 1.0000000 0.9072282 0.7069770 0.7535440  
## Cell.shape 0.6534700 0.9072282 1.0000000 0.6859481 0.7224624  
## Marg.adhesion 0.4878287 0.7069770 0.6859481 1.0000000 0.5945478  
## Epith.c.size 0.5235960 0.7535440 0.7224624 0.5945478 1.0000000  
## Bare.nuclei 0.5930914 0.6917088 0.7138775 0.6706483 0.5857161  
## Bl.cromatin 0.5537424 0.7555592 0.7353435 0.6685671 0.6181279  
## Normal.nucleoli 0.5340659 0.7193460 0.7179634 0.6031211 0.6289264  
## Mitoses 0.3509572 0.4607547 0.4412576 0.4188983 0.4805833  
## Bare.nuclei Bl.cromatin Normal.nucleoli Mitoses  
## Cl.thickness 0.5930914 0.5537424 0.5340659 0.3509572  
## Cell.size 0.6917088 0.7555592 0.7193460 0.4607547  
## Cell.shape 0.7138775 0.7353435 0.7179634 0.4412576  
## Marg.adhesion 0.6706483 0.6685671 0.6031211 0.4188983  
## Epith.c.size 0.5857161 0.6181279 0.6289264 0.4805833  
## Bare.nuclei 1.0000000 0.6806149 0.5842802 0.3392104  
## Bl.cromatin 0.6806149 1.0000000 0.6656015 0.3460109  
## Normal.nucleoli 0.5842802 0.6656015 1.0000000 0.4337573  
## Mitoses 0.3392104 0.3460109 0.4337573 1.0000000

#Scatterplot  
par(mfrow=c(5,2))  
jittered <- sapply(Breastcancer[,1:9], jitter)  
pairs(jittered, names(Breastcancer[,1:9]), col=Breastcancer$Class)



## - Splitting the dataset into training, test and predict.

# splitting data into training and test data with ratio 70/30  
set.seed(1234)  
ind <- sample(2, nrow(Breastcancer), replace = TRUE, prob=c(0.7, 0.3))  
train <- Breastcancer[ind==1,]  
valid <- Breastcancer[ind==2,]  
  
# Or   
split=sample.split(Breastcancer, SplitRatio = 0.7)   
train1=subset(Breastcancer,split==TRUE)   
test1=subset(Breastcancer,split==FALSE)   
dim(train1)

## [1] 478 10

### Run multiple models using different classifiers/algorithms

1. naive bayes -classifiers are a family of simple

library(klaR)  
x.nb <- NaiveBayes(Class~., data = train)  
x.nb.pred <- predict(x.nb,valid)$class   
x.nb.prob <- predict(x.nb,valid)$posterior  
table(x.nb.pred,valid$Class)

##   
## x.nb.pred benign malignant  
## benign 136 1  
## malignant 3 71

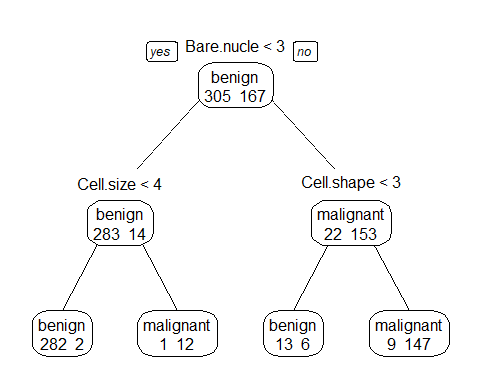
1. random forest : an implementation of the random forest and bagging ensemble algorithms utilizing conditional inference trees as base learners.

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 136 1  
## malignant 3 71

1. decision tree

x.rp <- rpart(Class ~ ., data=train)  
prp(x.rp, type = 1, extra = 1, split.font = 1, varlen = -10)

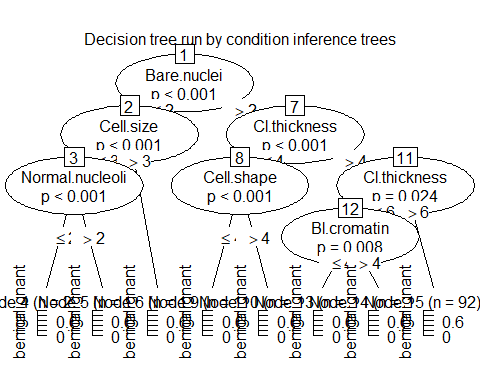


#prediction  
# predict classes for the evaluation data set  
x.rp.pred <- predict(x.rp, type="class", newdata=valid) # to ensemble  
# score the evaluation data set (extract the probabilities)  
x.rp.prob <- predict(x.rp, type="prob", newdata=valid)  
table(x.rp.pred,valid$Class)

##   
## x.rp.pred benign malignant  
## benign 136 1  
## malignant 3 71

1. conditional inference trees

require(party)  
x.ct <- ctree(Class ~ ., data=train)  
plot(x.ct, main="Decision tree run by 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 136 4  
## malignant 3 68

1. Create bagging using boostrap aggregating

set.seed(1234)  
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 136 1  
## malignant 3 71

1. svm - support vector machines # NOT WORK

set.seed(1234)  
x.svm <- svm(Class~., data = train, cost=1, gamma=0.03125, probability = TRUE)   
x.svm.pred <- predict(x.svm, type="class", newdata=valid) #ensemble; only give the class  
x.svm.prob <- predict(x.svm, type="prob", newdata=valid, probability = TRUE) # has to include probability = TRUE   
  
table(x.svm.pred,valid$Class)

##   
## x.svm.pred benign malignant  
## benign 136 2  
## malignant 3 70

1. neural network

library(nnet)  
set.seed(1234)  
x.nn <- nnet(Class~., data = train,size=1)

## # weights: 12  
## initial value 353.595928   
## iter 10 value 152.841714  
## iter 20 value 51.717836  
## iter 30 value 41.681004  
## iter 40 value 38.816114  
## iter 50 value 37.247762  
## iter 60 value 36.525222  
## iter 70 value 36.380864  
## iter 80 value 36.370267  
## iter 90 value 36.338409  
## iter 100 value 35.549485  
## final value 35.549485   
## 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 135 0  
## malignant 4 72

1. Quadratic Discriminant Analysis

set.seed(1234)  
train.num <- train %>% dplyr::select(-Class) %>% mutate\_if(is.factor,as.character)%>% mutate\_if(is.character,as.numeric)  
train.num$Class <- train$Class  
valid.num <- valid%>%dplyr::select(-Class) %>% mutate\_if(is.factor,as.character)%>% mutate\_if(is.character,as.numeric)  
valid.num$Class <- valid$Class  
x.qda <- qda(Class~., data = train.num)   
x.qda.pred <- predict(x.qda, valid.num)$class  
x.qda.prob <- predict(x.qda, valid.num)$posterior   
table(x.qda.pred,valid.num$Class)

##   
## x.qda.pred benign malignant  
## benign 135 1  
## malignant 4 71

1. Regularized Discriminant Analysis

library(klaR)  
set.seed(1234)  
x.rda <- rda(Class~., data = train)  
x.rda.pred <- predict(x.rda, valid)$class  
x.rda.prob <- predict(x.rda, valid)$posterior  
table(x.rda.pred,valid$Class)

##   
## x.rda.pred benign malignant  
## benign 136 2  
## malignant 3 70

## Plot individual file

png(filename="roc\_curve\_5\_models.png", width=700, height=700)

## Merging all the result together

classifier1<- c(0,1,0,1,0)  
classifier2<-c(0,0,0,1,0)  
classifier3<-c(0,0,0,1,0)  
classifier4<-c(0,0,0,1,0)  
classifier5<-c(0,0,0,1,0)  
classifier6<-c(0,0,0,1,0)  
classifier7<-c(0,0,0,1,0)  
classifier8<-c(0,0,0,1,0)  
  
#classifier9<-c(0,0,0,1,0)  
combine.df<-cbind(classifier1, classifier2, classifier3, classifier4, classifier5, classifier6, classifier7, classifier8  
 #,classifier9   
 )

## Assigned the predict value to these classier 1:n

classifier1<-x.rda.pred  
classifier2<-x.qda.pred  
classifier3<-x.nn.pred  
classifier4<-x.svm.pred  
classifier5<-x.ip.pred  
classifier6<-x.ct.pred   
classifier7<-x.cf.pred  
classifier8<-x.nb.pred  
#classifier9<-  
combine.df<-data.frame(classifier1, classifier2, classifier3, classifier4, classifier5, classifier6,classifier7, classifier8  
 #,classifier9  
 )  
head(combine.df)

## classifier1 classifier2 classifier3 classifier4 classifier5 classifier6  
## 5 benign benign benign benign benign benign  
## 14 benign benign benign benign benign benign  
## 16 malignant malignant malignant malignant malignant malignant  
## 27 benign benign benign benign benign benign  
## 29 benign benign benign benign benign benign  
## 30 benign benign benign benign benign benign  
## classifier7 classifier8  
## 5 benign benign  
## 14 benign benign  
## 16 malignant malignant  
## 27 benign benign  
## 29 benign benign  
## 30 benign benign

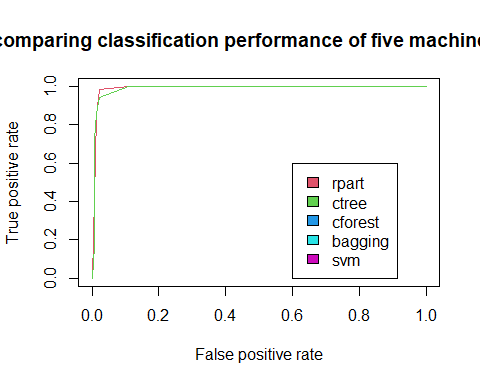
## Converting categorical values to numeric

combine.df[,1]<-ifelse(combine.df[,1]=="benign",0,1)  
combine.df[,2]<-ifelse(combine.df[,2]=="benign",0,1)  
combine.df[,3]<-ifelse(combine.df[,3]=="benign",0,1)  
combine.df[,4]<-ifelse(combine.df[,4]=="benign",0,1)  
combine.df[,5]<-ifelse(combine.df[,5]=="benign",0,1)  
combine.df[,6]<-ifelse(combine.df[,6]=="benign",0,1)  
combine.df[,7]<-ifelse(combine.df[,7]=="benign",0,1)  
combine.df[,8]<-ifelse(combine.df[,8]=="benign",0,1)  
  
# Checking  
head(combine.df)

## classifier1 classifier2 classifier3 classifier4 classifier5 classifier6  
## 5 0 0 0 0 0 0  
## 14 0 0 0 0 0 0  
## 16 1 1 1 1 1 1  
## 27 0 0 0 0 0 0  
## 29 0 0 0 0 0 0  
## 30 0 0 0 0 0 0  
## classifier7 classifier8  
## 5 0 0  
## 14 0 0  
## 16 1 1  
## 27 0 0  
## 29 0 0  
## 30 0 0

#combine.df[,9]<- ifelse(rowSums(combine.df)>=4,"malignant","benign")  
#head(combine.df)  
#table(combine.df[,9], Breastcancer$Class)

# load the ROCR package which draws the ROC curves  
#require(ROCR)  
  
# 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")  
# plot it  
plot(x.rp.perf, col=2, main="ROC curves comparing classification performance of five machine learning models")  
  
# Draw a legend.  
legend(0.6, 0.6, c('rpart', 'ctree', 'cforest','bagging','svm'), 2:6)  
  
# ctree  
x.ct.prob.rocr <- prediction(x.ct.prob, Breastcancer[ind == 2,'Class'])  
x.ct.perf <- performance(x.ct.prob.rocr, "tpr","fpr")  
# add=TRUE draws on the existing chart   
plot(x.ct.perf, col=3, add=TRUE)



#plot(x.svm.perf, col=6, add=TRUE) #svm  
#plot(x.ip.perf, col=5, add=TRUE) #bagging  
#plot(x.cf.perf, col=4, add=TRUE) #cforest

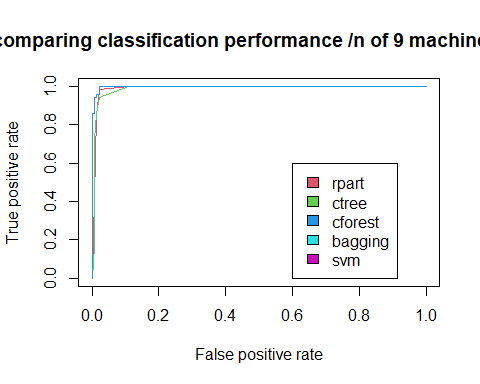
### Plot ROC curves for comparing all the classifiers results

#load the ROCR package which draws the ROC curves  
library(ROCR)  
# 1.rptree  
# create an ROCR prediction object from rpart() probabilities  
x.rp.prob.rocr <- prediction(x.rp.prob[,2], Breastcancer[ind == 2,'Class'])  
x.rp.prob.rocr #Slot "n.pos": check and see if it identifies the positive class; compare with summary(valid$Class)

## A prediction instance  
## with 211 data points

# 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")  
# 2.ctree  
x.ct.prob.rocr <- prediction(x.ct.prob, Breastcancer[ind == 2,'Class'])  
x.ct.perf <- performance(x.ct.prob.rocr, "tpr","fpr")  
# 3.cforest  
x.cf.prob.rocr <- prediction(x.cf.prob, Breastcancer[ind == 2,'Class'])  
x.cf.perf <- performance(x.cf.prob.rocr, "tpr","fpr")  
# 4.bagging  
x.ip.prob.rocr <- prediction(x.ip.prob[,2], Breastcancer[ind == 2,'Class'])  
x.ip.perf <- performance(x.ip.prob.rocr, "tpr","fpr")  
# 5.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")  
# 6.nb ### calculate the prob   
#x.nb.prob.rocr <- prediction(x.nb.prob[,2], Breastcancer[ind == 2,'Class'])  
#x.nb.perf <- performance(x.nb.prob.rocr, "tpr","fpr")  
# 7.nn  
#x.nn.prob.rocr <- prediction(x.nn.prob, Breastcancer[ind == 2,'Class'])  
#x.nn.perf <- performance(x.nn.prob.rocr, "tpr","fpr")  
# 8.qda  
#x.qda.prob.rocr <- prediction(x.qda.prob[,2], Breastcancer[ind == 2,'Class'])  
#x.qda.perf <- performance(x.qda.prob.rocr, "tpr","fpr")  
# 9.rda  
#x.rda.prob.rocr <- prediction(x.rda.prob[,2], Breastancer[ind == 2,'Class'])  
#x.rda.perf <- performance(x.rda.prob.rocr, "tpr","fpr")

####### plot  
# Output the plot to a PNG file for display on web. To draw to the screen,   
# comment this line out.  
#png(filename="roc\_curve\_models1.png", width=700, height=700)  
#par(mfrow=c(1,2))  
plot(x.rp.perf, col=2, main="ROC curves comparing classification performance /n of 9 machine learning models") #   
legend(0.6, 0.6, c('rpart', 'ctree', 'cforest','bagging','svm'), 2:6)# Draw a legend.  
plot(x.ct.perf, col=3, add=TRUE)# add=TRUE draws on the existing chart #has to be run together.  
plot(x.cf.perf, col=4, add=TRUE)



#plot(x.ip.perf, col=5, add=TRUE)  
#plot(x.svm.perf, col=6, add=TRUE)  
# Close and save the PNG file.  
#dev.off()  
png(filename="roc\_curve\_models2.png", width=700, height=700)  
#plot(x.nb.perf, col=7, main="ROC curves comparing classification performance of the other 4 machine learning models")  
#legend(0.6, 0.6, c('naive bayes', 'neural network', 'qda','rda'), 7:10)  
#plot(x.nn.perf, col=8, add=TRUE)  
#plot(x.qda.perf, col=9, add=TRUE)  
#plot(x.rda.perf, col=10, add=TRUE)  
#dev.off()

### Ensemble: combine all the nine classifiers and generate the final prediction based on the majority rule.

classifier <- data.frame(cbind(x.rp.pred, x.ct.pred, x.cf.pred, x.ip.pred, x.svm.pred, x.nb.pred,x.nn.pred,x.qda.pred,x.rda.pred))  
names(classifier) <-c('recursive.tree','conditional.inference.tree','random.forest','bootstrap','svm','naive.bayes','neutral.network','qda','rda')  
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')  
  
#confusion matrix   
#library(caret)  
confusionMatrix(as.factor(classifier$predition), valid$Class, positive = 'malignant')

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction benign malignant  
## benign 136 0  
## malignant 3 72  
##   
## Accuracy : 0.9858   
## 95% CI : (0.959, 0.9971)  
## No Information Rate : 0.6588   
## P-Value [Acc > NIR] : <2e-16   
##   
## Kappa : 0.9687   
##   
## Mcnemar's Test P-Value : 0.2482   
##   
## Sensitivity : 1.0000   
## Specificity : 0.9784   
## Pos Pred Value : 0.9600   
## Neg Pred Value : 1.0000   
## Prevalence : 0.3412   
## Detection Rate : 0.3412   
## Detection Prevalence : 0.3555   
## Balanced Accuracy : 0.9892   
##   
## 'Positive' Class : malignant   
##