Project Two

Hari Aravind

3/17/2021

Contents

Project Description:

This project involves generating several classification models for the same data set and then combining the output from the models in an ensemble fashion.

```
## Metapackage of all tidyverse packages

library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.3     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   1.4.0     v forcats 0.5.0
```

```
## Warning: package 'ggplot2' was built under R version 4.0.4
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
## Inputing data

list.files(path = "../input")
```

```
## character(0)
```

```
## Loading mlbench

require(mlbench)
```

```
## Loading required package: mlbench
```

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

```
## Loading the data set
data(BreastCancer)
```

```r
## Removing missing values from the dataset
BreastCancer <- na.omit(BreastCancer)

## Remove the unique identifier, which is useless and would confuse the machine learning algorithms
BreastCancer$Id <- NULL

## To view the data
head(BreastCancer,5)
```

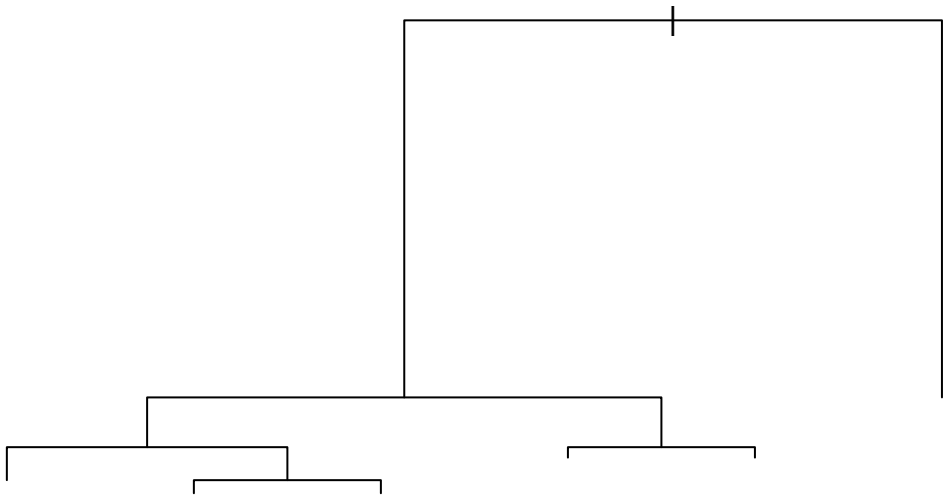
```
##   Cl.thickness Cell.size Cell.shape Marg.adhesion Epith.c.size Bare.nuclei
## 1            5         1          1             1            2           1
## 2            5         4          4             5            7          10
## 3            3         1          1             1            2           2
## 4            6         8          8             1            3           4
## 5            4         1          1             3            2           1
##   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
```

```r
## Partition the data set for 80% training & 20% for evaluation
set.seed(2)

ind <- sample(2, nrow(BreastCancer), replace = TRUE, prob=c(0.8, 0.2))

## Create model using recursive partitioning on the training data set
require(rpart)
```

```
## Loading required package: rpart
```

```r
x.rp <- rpart(Class ~ ., data=BreastCancer[ind == 1,])

## Predict classes for the evaluation data set
x.rp.pred <- predict(x.rp, type="class", newdata=BreastCancer[ind == 2,])

## Score the evaluation data set (extract the probabilities)
x.rp.prob <- predict(x.rp, type="prob", newdata=BreastCancer[ind == 2,])
```

```r
## To view the decision tree
plot(x.rp, main="Decision tree created using rpart")
```




```
## Create model using conditional inference trees
require(party)
```

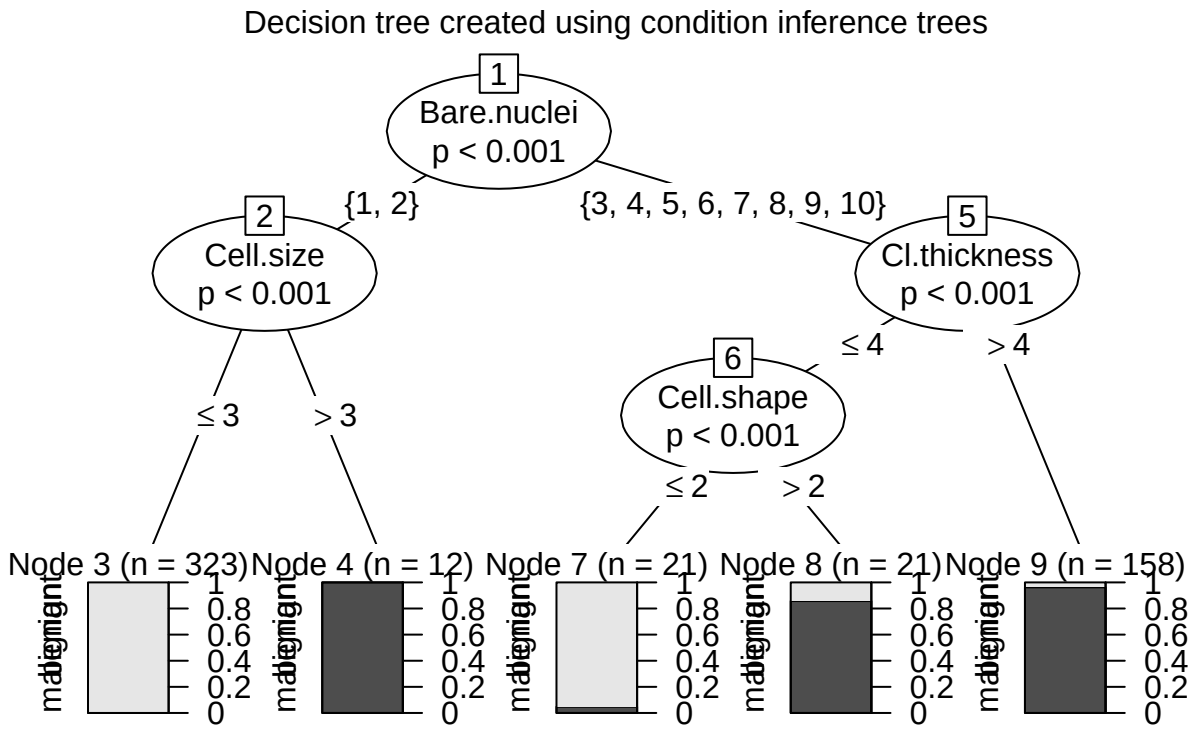
```
## Loading required package: party
## Warning: package 'party' was built under R version 4.0.4
## Loading required package: grid
## Loading required package: mvtnorm
## Loading required package: modeltools
## Loading required package: stats4
## Loading required package: strucchange
## Warning: package '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=BreastCancer[ind == 1,])

x.ct.pred <- predict(x.ct, newdata=BreastCancer[ind == 2,])

x.ct.prob <-  1- unlist(treeresponse(x.ct, BreastCancer[ind == 2,]), use.names=F)[seq(1,nrow(BreastCanc
```

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




```
## Create model using random forest and bagging ensemble using conditional inference trees
x.cf <- cforest(Class ~ ., data=BreastCancer[ind == 1,], control = cforest_unbiased(mtry = ncol(BreastCa

x.cf.pred <- predict(x.cf, newdata=BreastCancer[ind == 2,])

x.cf.prob <-  1- unlist(treeresponse(x.cf, BreastCancer[ind == 2,]), use.names=F)[seq(1,nrow(BreastCanc
```

```
## Create model using bagging (bootstrap aggregating)
require(ipred)
```

```
## Loading required package: ipred
```

```r
x.ip <- bagging(Class ~ ., data=BreastCancer[ind == 1,])

x.ip.prob <- predict(x.ip, type="prob", newdata=BreastCancer[ind == 2,])
```

```r
## Create model using svm (support vector machine)
require(e1071)
```

```
## Loading required package: e1071
```

```r
## SVM requires tuning
x.svm.tune <- tune.svm(Class~., data = BreastCancer[ind == 1,],gamma = 2^(-8:1), cost = 2^(0:4))

## Display the tuning results (in text format)
x.svm.tune
```

```
##
## Parameter tuning of 'svm':
##
## - sampling method: 10-fold cross validation
##
## - best parameters:
##  gamma cost
##   0.25    2
##
## - best performance: 0.02243187
```

```r
## If the tuning results are on the margin of the parameters (e.g., gamma = 2^-8), then widen the param
## I manually copied the cost and gamma from console messages above to parameters below.
x.svm <- svm(Class~., data = BreastCancer[ind == 1,], cost=4, gamma=0.0625, probability = TRUE)

x.svm.prob <- predict(x.svm, type="prob", newdata=BreastCancer[ind == 2,], probability = TRUE)
```

```r
# 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,

png(filename="roc_curve_5_models.png", width=800, height=700)
```

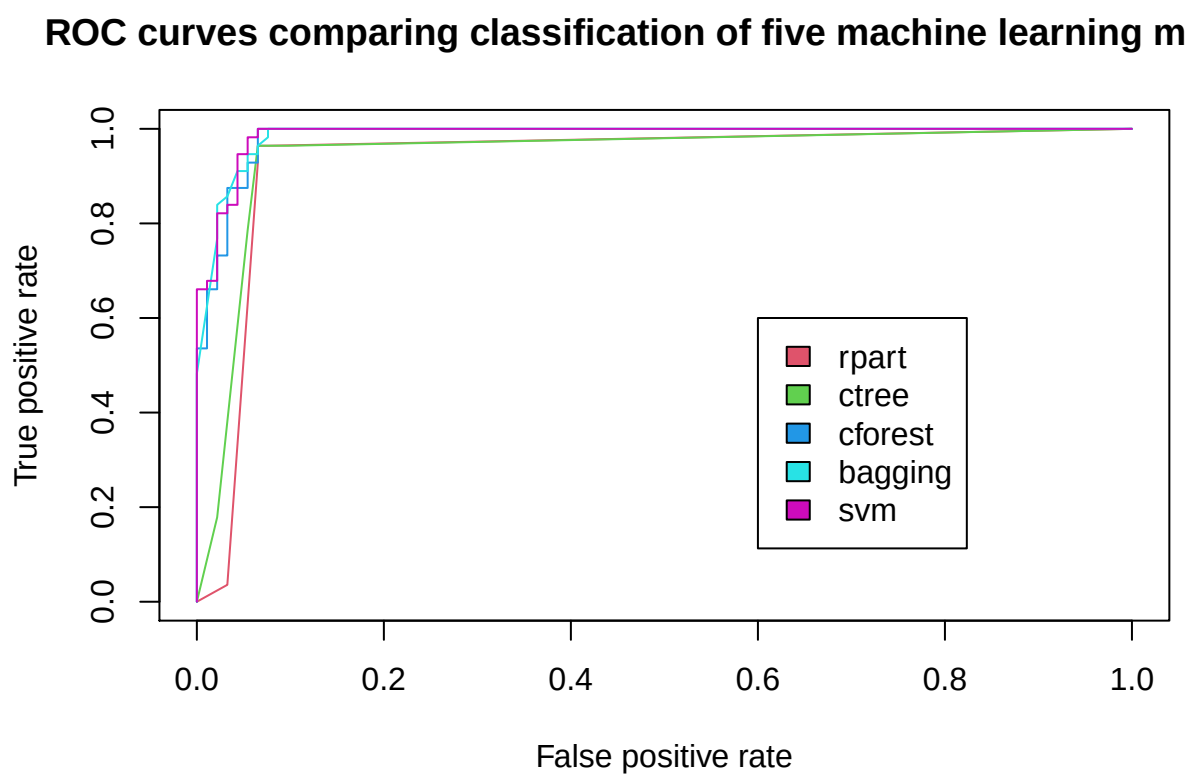
```r
## load the ROCR package which draws the ROC curves
require(ROCR)
```

```
## Loading required package: ROCR
```

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

```r
## 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 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)

## cforest
x.cf.prob.rocr <- prediction(x.cf.prob, BreastCancer[ind == 2,'Class'])
x.cf.perf <- performance(x.cf.prob.rocr, "tpr","fpr")
plot(x.cf.perf, col=4, add=TRUE)

## bagging
x.ip.prob.rocr <- prediction(x.ip.prob[,2], BreastCancer[ind == 2,'Class'])
x.ip.perf <- performance(x.ip.prob.rocr, "tpr","fpr")
plot(x.ip.perf, col=5, add=TRUE)

## 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")
plot(x.svm.perf, col=6, add=TRUE)
```




```
## Close and save the PNG file.
dev.off()
```

```
## png
##   3
```

```
## Creating an ensemble for combining other classifiers by majority vote
classifier.rpart <- c(x.rp.prob)
```

```
classifier.ctree <- c(x.ct.prob)
classifier.cforeset <- c(x.cf.prob)
classifier.bagging <- c(x.ip.prob)
classifier.svm <- c(x.svm.prob)

combine.classifier <- cbind(classifier.rpart,classifier.ctree,classifier.cforeset,classifier.bagging,cla

#head(combine)
head(x.rp.prob)
```

```
##    benign malignant
## 5    1.00      0.00
## 6    0.05      0.95
## 8    1.00      0.00
## 16   0.05      0.95
## 17   1.00      0.00
## 23   1.00      0.00
```

```
head(x.ct.prob)
```

```
## [1] 0.0000000 0.9620253 0.0000000 1.0000000 0.0000000 0.0000000
```

```
combine.classifier[,1]<-ifelse(combine.classifier[,1]=="benign", 0, 1)
combine.classifier[,2]<-ifelse(combine.classifier[,2]=="benign", 0, 1)
combine.classifier[,3]<-ifelse(combine.classifier[,3]=="benign", 0, 1)
combine.classifier[,4]<-ifelse(combine.classifier[,4]=="benign", 0, 1)
combine.classifier[,5]<-ifelse(combine.classifier[,5]=="benign", 0, 1)
majority.vote<-rowSums(combine.classifier)
head(majority.vote)
```

```
## [1] 5 5 5 5 5 5
```