Ensemble Methods

Aaron Thai

2024-03-18

Load dataset

```
#load the mlbench package which has the BreastCancer data set
library(mlbench)

# if you don't have any required package, use the install.packages() command
# load the data set
data(BreastCancer)
```

clean data by removing missing values and extra columns

```
# 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
```

Create training and test sets (indexes)

```
# partition the data set for 80% training and 20% evaluation (adapted from ?randomForest)
set.seed(2)

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

Create Model 1: Decision Tree

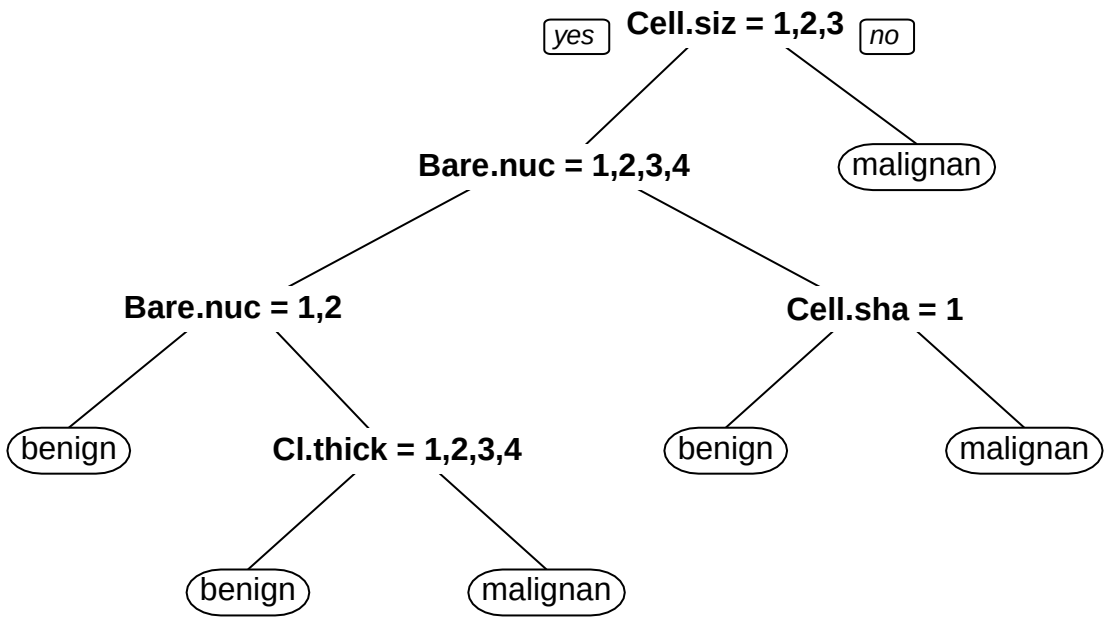
```
# Model 1: Decision Tree
# create model using recursive partitioning on the training data set
require(rpart)
```

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

```
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,])

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

```
## Warning: package 'rpart.plot' was built under R version 4.1.3
```

```
prp(x.rp)
```



Create Model 2: Conditional Inference Trees

```
# Model 2: Conditional Inference Trees
# create model using conditional inference trees
require(party)
```

```
## Loading required package: party
```

```
## Loading required package: grid
```

```
## Loading required package: mvtnorm
```

```
## Loading required package: modeltools
```

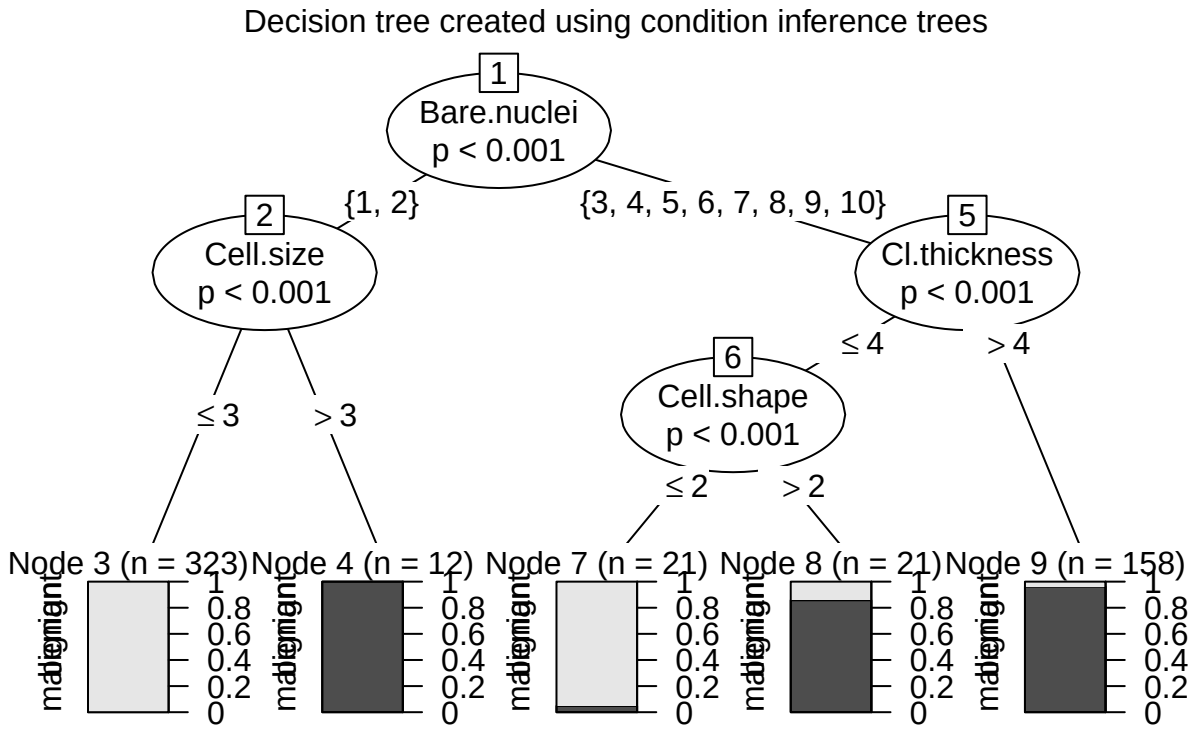
```
## Loading required package: stats4
```

```
## Loading required package: strucchange
```

```
## Warning: package 'strucchange' was built under R version 4.1.3
```

```
## Loading required package: zoo

## Warning: package 'zoo' was built under R version 4.1.3

##
## 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 4.1.3
```

```
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, uncomment this line.
plot(x.ct, main="Decision tree created using condition inference trees")
```



Create Model 3: Random Forest

```
# Model 3: Random Forest
# 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 4: Bagging

```
# Model 4: Bagging
# create model using bagging (bootstrap aggregating)
require(ipred)
```

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

```
## Warning: package 'ipred' was built under R version 4.1.3
```

```
x.ip <- bagging(Class ~ ., data=BreastCancer[ind == 1,])
x.ip.pred <- predict(x.ip, type="class", newdata=BreastCancer[ind == 2,])
x.ip.prob <- predict(x.ip, type="prob", newdata=BreastCancer[ind == 2,])
```

Create Model 5: SVM

```
# Model 5: SVM
# create model using svm (support vector machine)
require(e1071)
```

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

```
## Warning: package 'e1071' was built under R version 4.1.3
```

```
# svm requires tuning
x.svm.tune <- tune(svm, Class~., data = BreastCancer[ind == 1,],
                   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
```

```
##
## Parameter tuning of 'svm':
##
## - sampling method: fixed training/validation set
##
## - best parameters:
##   gamma cost
##  0.0625    1
##
## - best performance: 0.02234637
```

```r
# 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 = BreastCancer[ind == 1,], cost=4, gamma=0.0625, probability = TRUE)
x.svm.pred <- predict(x.svm, type="class", newdata=BreastCancer[ind == 2,], probability = TRUE)
x.svm.prob <- predict(x.svm, type="prob", newdata=BreastCancer[ind == 2,], probability = TRUE)
```

Create Ensemble Model using the 5 Models

```r
# Ensemble Model
# Create ensemble to combine classifiers

# get predicted classes from each model
classifier1 <- x.rp.pred
classifier2 <- x.ct.pred
classifier3 <- x.cf.pred
classifier4 <- x.ip.pred
classifier5 <- x.svm.pred

# combine results into a dataframe
combine.df<-data.frame(cbind(classifier1, classifier2, classifier3, classifier4, classifier5))
combine.df$vote <- rowSums(combine.df) # 1 is benign, 2 is malignant

# predict the result with the ensemble with majority vote
combine.df$class <- ifelse(combine.df$vote > 5, "malignant","benign")
combine.df$class <- factor(combine.df$class)
```

Plot ROC Curve for each Model

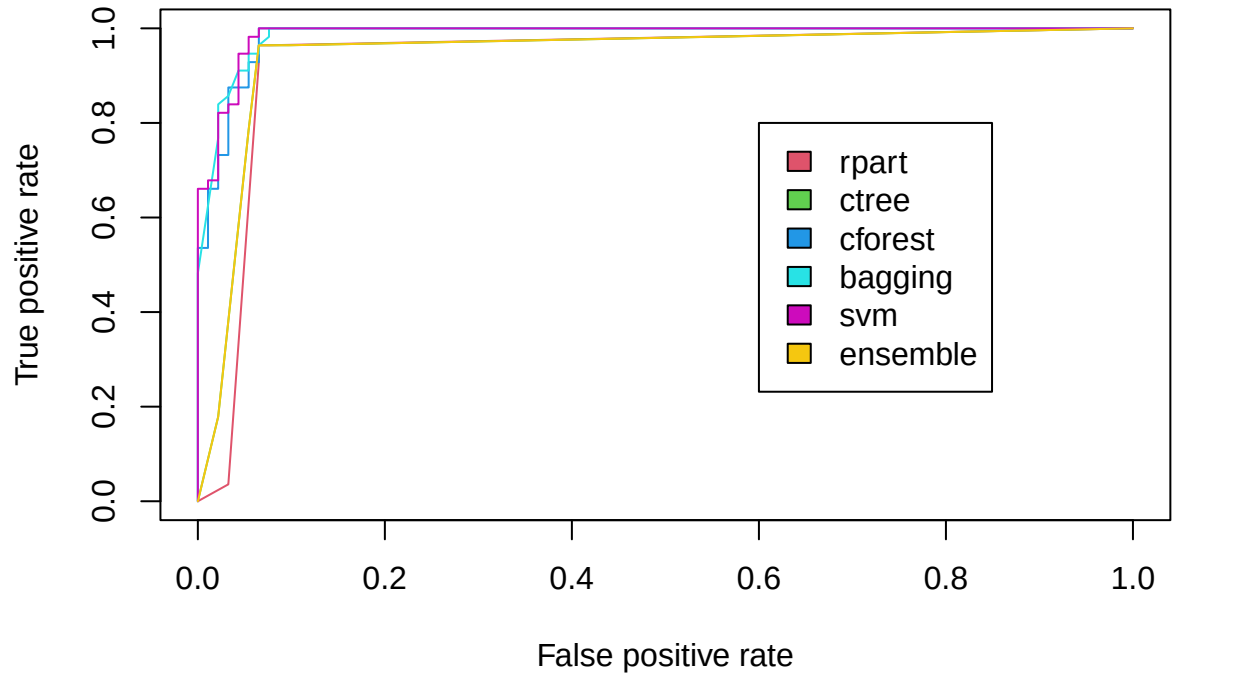
```r
#
## plot ROC curves to compare the performance of the individual classifiers
##

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

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

```
## Warning: package 'ROCR' was built under R version 4.1.3
```

```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 performance of five machine learning m

# Draw a legend.
legend(0.6, 0.8, c('rpart', 'ctree', 'cforest','bagging','svm', 'ensemble'), 2:7)

# 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.new()
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)

# ensemble model
x.en.prob.rocr <- prediction(combine.df$vote, BreastCancer[ind == 2,'Class'])
x.en.perf <- performance(x.ct.prob.rocr, "tpr","fpr")
# add=TRUE draws on the existing chart
plot(x.en.perf, col=7, add=TRUE)
```


: curves comparing classification performance of five machine learning
True positive rate
False positive rate
0.0
0.2
0.4
0.6
0.8
1.0
rpart
ctree
cforest
bagging
svm
ensemble