**Assignment #6: Tree-Based Methods**

**Submit through link: eCampus -> Assignments->Assignment 6 Submission**

**Deadline: November 20 (Tuesday) @17:00 pm**

**The filename should have this format: LastName-FirstName-hw06.doc**

**Problem 1 (10pt)**

In the lab, a classification tree was applied to the Carseats data set after converting Sales into a binary response variable. This question will seek to predict Sales using regression trees and related approaches, treating the response as a quantitative variable (that is, without the conversion).

(a) Split the data set into a training set and a test set.

Answer :

mydata <- Carseats

library(caTools)

set.seed(1)

train <- sample.split(1:400, SplitRatio = 0.7)

test = !train

carseats.train <- subset(x = mydata,subset = train)

carseats.test <- mydata[test,]

(b) Fit a regression tree to the training set. Plot the tree, and interpret the results. Then compute the test MSE.

Answer:

tree.carseat = tree(Sales~ . , carseats.train)

plot(tree.carseat)

text(tree.carseat,pretty=0)

summary(tree.carseat)

# calculating test MSE

tree.pred <-predict(tree.carseat,newdata = carseats.test)

mean((tree.pred-carseats.test)^2)

= **4.01**

(c) Prune the tree obtained in (b). Use cross validation to determine the optimal level of tree complexity. Plot the pruned tree and interpret the results. Compute the test MSE of the pruned tree. Does pruning improve the test error?

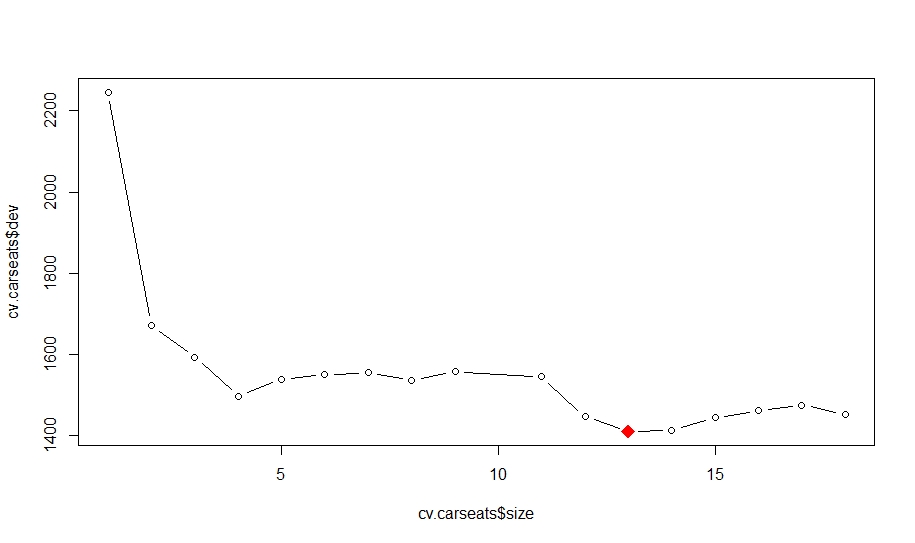
Answer:

cv.carseats<- cv.tree(tree.carseat )

cv.carseats

plot( cv.carseats$size, cv.carseats$dev, type = "b" )

points(13,1410.527, col ="red",pch= 18, cex= 2)



From the plot we can see the best tree size= **13 (it has the lowest test MSE)**

prune.carseats <- prune.tree(tree.carseat,best = 13)

plot(prune.carseats)

text(prune.carseats, pretty=0)

sales.pred <- predict(prune.carseats, newdata = carseats.test)

mean((sales.pred - carseats.test$Sales)^2)

= **4.247**

**No for this dataset cost complexity pruning didn’t seem to do very well in reducing RSS. On the contrary it has increased the MSE a little. But it has indeed made the tree less complex with fewer leaves.**

(d) Use the bagging approach to analyze the data. What test MSE do you obtain? Determine which variables are most important.

Answer:

set.seed(1)

bag.carseats <- randomForest(Sales~ ., data= carseats.train , mtry=10 ,importance=TRUE)

bag.carseats

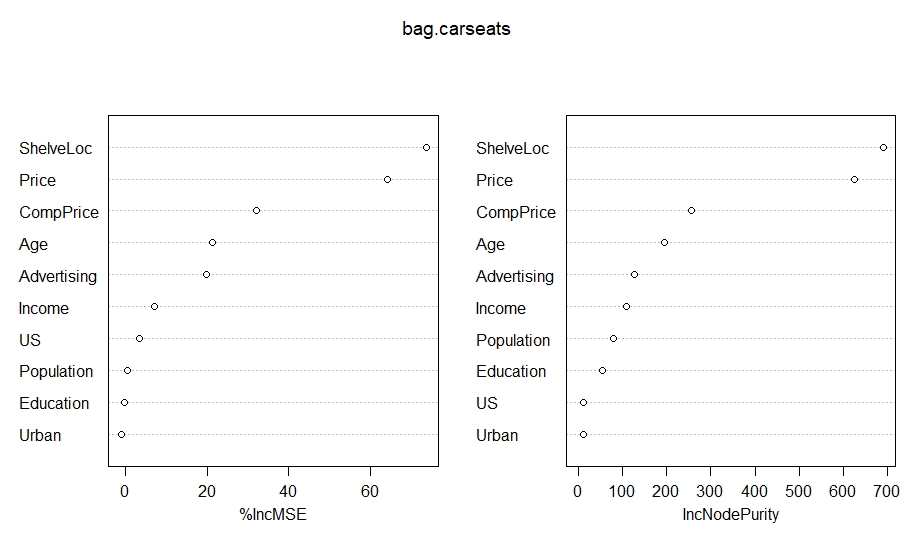
pred.bag <- predict(bag.carseats, newdata = carseats.test)

mean((pred.bag - carseats.test$Sales)^2)

= **2.366**

importance(x =bag.carseats)

varImpPlot(bag.carseats)



If we look at the plot we can see that shelve location, price and comp price are the most important variables and in the respective order too.

(e) Use random forests to analyze the data. What test MSE do you obtain? Determine which variables are most important.

Answer:

set.seed(1)

rf.carseats <- randomForest(Sales~ ., data= carseats.train , mtry=3 ,importance=TRUE)

rf.carseats

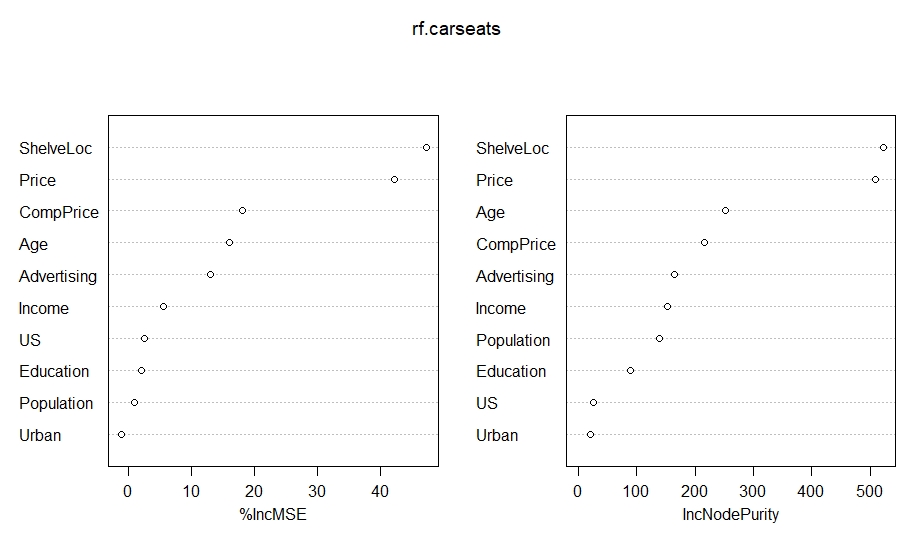
pred.rf <- predict(rf.carseats, newdata = carseats.test)

mean((pred.rf - carseats.test$Sales)^2)

=**2.62**

importance(x =rf.carseats)

varImpPlot(rf.carseats)



**Again shelveLoc, Price and compprice are the most important variables according to both node purity and % increase in MSE**

**Problem 2 (5pt)**

In the lab, we applied random forests to the Boston data using mtry=6 and ntree=100.

(a) Consider a more comprehensive range of values for mtry: 1, 2,…,13. Given each value of mtry, find the test error resulting from random forests on the Boston data (using ntree=100). Create a plot displaying the test error rate vs. the value of mtry. Comment on the results in the plot.

Answer:

library(MASS)

dim(Boston)

set.seed(1)

train <- sample.split(1:506, 0.7)

boston.train <- subset(Boston, train)

boston.test <- subset(Boston, !train)

medv.test <- subset(Boston$medv , !train)

# using randomForest

MSE <- rep(0,13)

mtry<- 1:13

for(i in 1:13) {

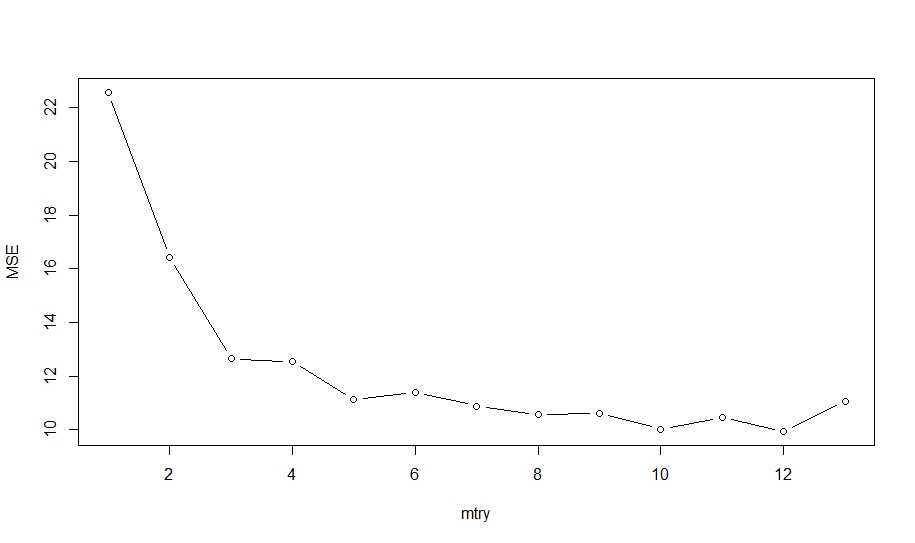
rf.Boston <- randomForest(medv~ ., data= boston.train , mtry=i ,ntree= 100)

pred.rf <- predict(rf.Boston, newdata = boston.test)

MSE[i] <-mean((pred.rf - medv.test)^2)

}

plot(mtry, MSE, type= 'b')



We can see from this curve between square test error and mtry that as we increase the mtry the MSE initially decreases very steeply then starts to increase which was expected as the in random forest the optimum value of mtry to get the least test error is somewhere between sqrt(no of predictors).

(b) Similarly, consider a range of values for ntree (between 5 to 200). Given each value of ntree, find the test error resulting from random forests (using mtry=6). Create a plot displaying the test error vs. the value of ntree. Comment on the results in the plot.

#rf

MSE <- rep(0,196)

ntree<- 5: 200

for(i in 5:200) {

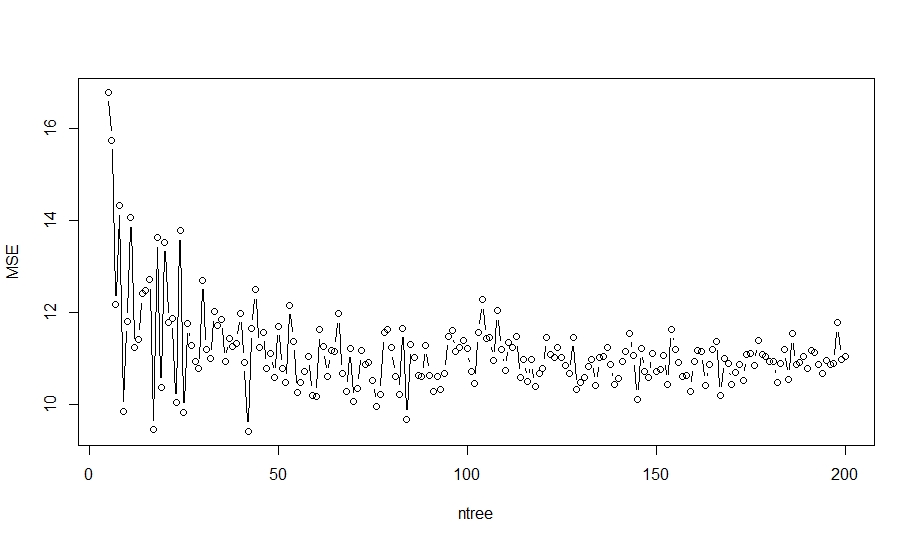
rf.Boston <- randomForest(medv~ ., data= boston.train , mtry=6 ,ntree= i)

pred.rf <- predict(rf.Boston, newdata = boston.test)

MSE[(i-4)] <-mean((pred.rf - medv.test)^2)

}

plot(ntree, MSE, type= 'b')



The plot is very unstable as the MSE value is oscillating very quickly on changing the ntree value.

But the overall trend is that the MSE is decreasing with increasing ntree value, but the general trend also gets flat after a certain value of ntree.