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

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

**Deadline: November 16 (Saturday) @17:00 pm**

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

**For the first two questions it would be helpful to go over R lecture: Lecture 18\_R.pdf.**

**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

> library(tree)

> library(ISLR)

> attach(Carseats)

> set.seed(1)

> train <- sample(1:nrow(Carseats), nrow(Carseats) / 2)

> Carseats.train <- Carseats[train, ]

> Carseats.test <- Carseats[-train, ]

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

> tree.carseats <- tree(Sales ~ ., data = Carseats.train)

> summary(tree.carseats)

Regression tree:

tree(formula = Sales ~ ., data = Carseats.train)

Variables actually used in tree construction:

[1] "ShelveLoc" "Price" "Age" "Advertising" "CompPrice" "US"

Number of terminal nodes: 18

Residual mean deviance: 2.167 = 394.3 / 182

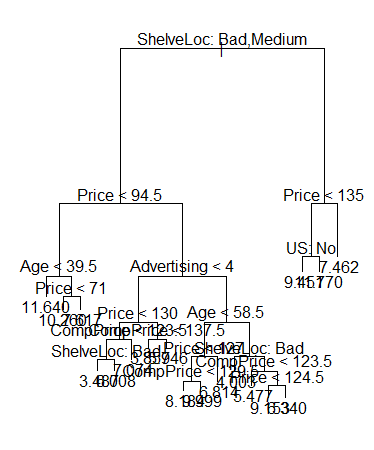
Distribution of residuals:

Min. 1st Qu. Median Mean 3rd Qu. Max.

-3.88200 -0.88200 -0.08712 0.00000 0.89590 4.09900

> plot(tree.carseats)

> text(tree.carseats, pretty = 0)



> yhat <- predict(tree.carseats, newdata = Carseats.test)

> mean((yhat - Carseats.test$Sales)^2)

[1] 5.47643

***The test MSE obtained is 4.922***

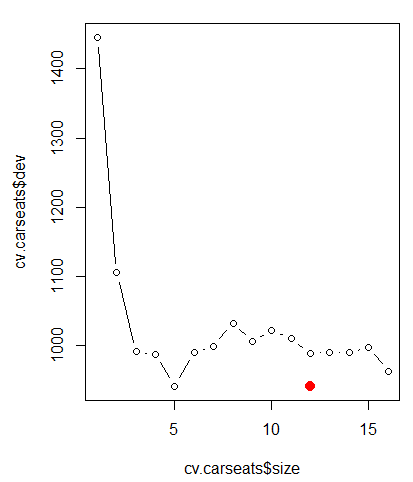
(c) Prune the tree obtained in (b). Use cross validation to determine the optimal level of tree complexity. Plot the pruned tree and interpret the results. Compute the test MSE of the pruned tree. Does pruning improve the test error?

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

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

> tree.min <- which.min(cv.carseats$dev)

> points(tree.min, cv.carseats$dev[tree.min], col = "red", cex = 2, pch = 20)

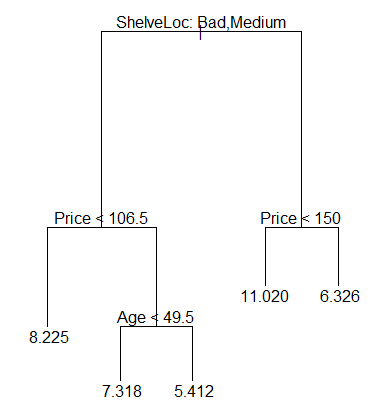


In this case, the tree of size 8 is selected by cross-validation. We now prune the tree to obtain the 8-node tree.

> prune.carseats <- prune.tree(tree.carseats, best = 5)

> plot(prune.carseats)

> text(prune.carseats, pretty = 0)



> yhat <- predict(prune.carseats, newdata = Carseats.test)

> mean((yhat - Carseats.test$Sales)^2)

[1] 6.114674

***We see that pruning increases the test MSE to 6.115***

**In this case, pruning the tree actually makes the MSE worse - it goes from 4.922 to 6.115. This isn't always the case - depending on the test and training sets, sometimes the pruned tree will result in a better MSE than the unpruned tree (when the unpruned tree has been overfit)**

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

> library(randomForest)

> set.seed(1)

> bag.car = randomForest(Sales~.,data=Car.train,mtry = 10, importance = TRUE)

> yhat.bag = predict(bag.car,newdata=Car.test)

> mean((yhat.bag-Car.test$Sales)^2)

**[1] 2.743098**

***The test MSE is 2.743 with bagging. Thus the MSE reduced considerably with bagging the trees.***

> importance(bag.car)

%IncMSE IncNodePurity

CompPrice 28.9098120 207.569850

Income 13.6826772 96.618412

Advertising 11.4958129 86.090140

Population -0.2023831 59.696804

Price 49.3091766 400.405728

ShelveLoc 56.9036512 538.494370

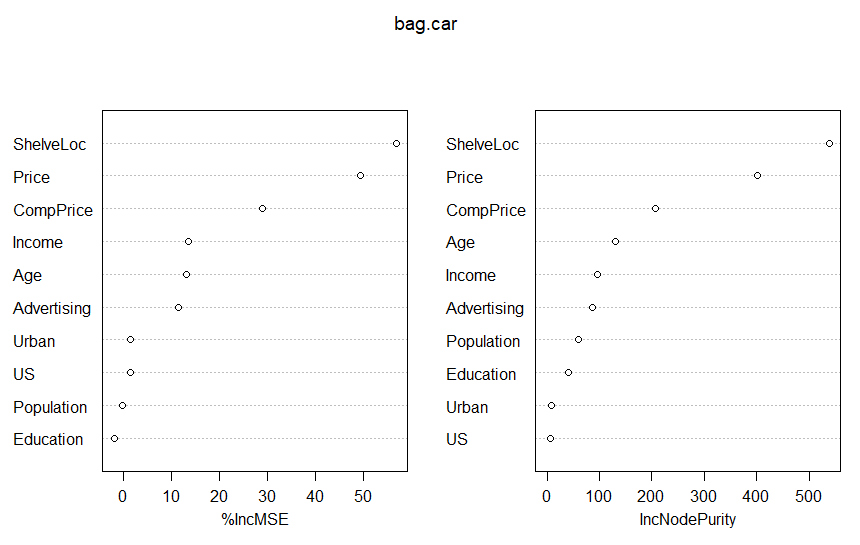
Age 13.1449130 129.564321

Education -1.8877687 40.863575

Urban 1.5744168 8.087432

US 1.5694968 6.801470

> varImpPlot(bag.car)



**As seen above, it looks like the price of the carseat and where it is located on the shelf are the most important predictors of how a carseat will sell. Age, competitor price, and income, and advertising budget also appear to have an effect, but all other variables seem to be less important.**

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

> set.seed(1)

> rf.car = randomForest(Sales~.,data=Car.train,mtry = 3, importance = TRUE)

> yhat.rf = predict(rf.car,newdata=Car.test)

> mean((yhat.rf-Car.test$Sales)^2)

**[1] 2.941463**

***The Test MSE is 2.941 with random forest. This is slightly higher than bagging which is not expected with Random Forest. This may have happened because the number of variables were too few for random forest to give better results and randomization of variables selected resulted in poorer trees because random forest gave more importance to variables that were not ideal.***

> importance(rf.car)

%IncMSE IncNodePurity

CompPrice 16.7877299 190.62748

Income 7.6774238 128.34227

Advertising 10.4351859 120.55621

Population -1.0192573 103.04813

Price 30.5926744 339.43316

ShelveLoc 38.8151768 376.96711

Age 10.1093189 153.62657

Education -2.5751222 71.35645

Urban -0.4498694 18.33295

US 2.9023810 13.92101

> varImpPlot(rf.car)

**Here, again, it looks like price and the location on the shelf are the most important predictors, but age and advertising again look to have some import. Competitors price, age, income, advertising, and population also have a good effect on the model. Population is a feature that did not have much effect in the bagging model but randomization of variables showed us that this feature was ignored in bagging and was given due importance in the random forest model.**

**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.

> set.seed(10)

> train <- sample(1:nrow(Boston), nrow(Boston)/2)

> Boston.train <- Boston[train, ]

> Boston.test <- Boston[-train, ]

> test.MSE <- c()

> for(i in 1:13)

+ {

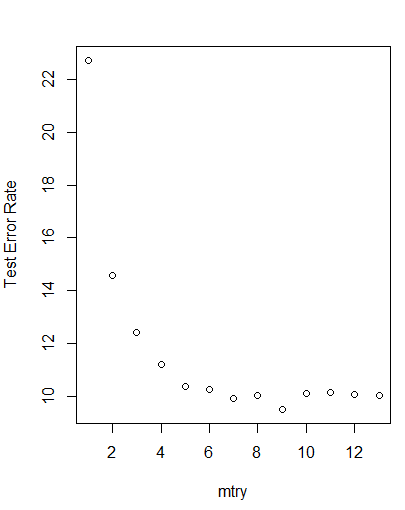
+ rf.boston <- randomForest(medv ~.,data = Boston, subset = train, mtry=i, ntree = 100, importance = TRUE)

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

+ test.MSE[i]<- mean((yhat.rf - Boston.test$medv)^2)

+ }

> plot(1:13, test.MSE, xlab="mtry", ylab = "Test Error Rate")



***With the cross validation, we determine that mtree=9 is the value that reduces the test error best and gives us the most optimum tree with lowest entropy.***

(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.

> test2.MSE <- c()

> for(i in 5:200) {

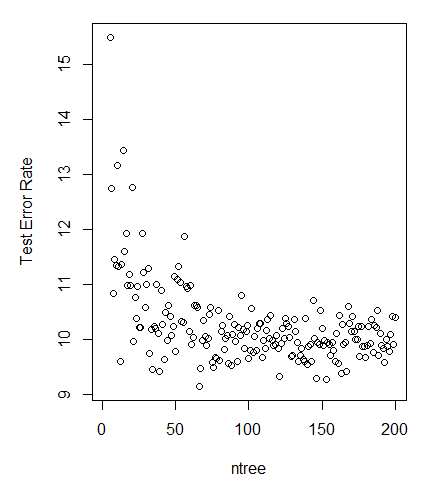
+ rf.boston <- randomForest(medv~., data=Boston, subset=train, mtry=6, ntree = i,importance=TRUE)

+ yhat.rf=predict(rf.boston, newdata=Boston.test)

+ test2.MSE[i] <- mean((yhat.rf-Boston.test$medv)^2)

+ }

> plot(1:200, test2.MSE, xlab= "ntree", ylab="Test Error Rate" )



***With ntree=65, we get the lowest test MSE by cross validating the tree with number of trees in the random forest model.***

**Problem 3 (Bonus question 5pt)** Write the solution to the in-class exercise a)-e) in Lecture 17 slides.

