Assignment 5 4620

Tyler Poelking

12/3/2017

In the lab, we applied random forests to the Boston data using mtry=6 and using ntree=25 and ntree=500. Create a plot displaying the test error resulting from random forests on this data set for a more comprehensive range of values for mtry and ntree. You can model your plot after Figure 8.10. Describe the results obtained.

library(randomForest)

## randomForest 4.6-12

## Type rfNews() to see new features/changes/bug fixes.

library(MASS)  
attach(Boston)  
library(ISLR)  
set.seed(1)  
  
# Make train and test data  
train = sample(dim(Boston)[1], dim(Boston)[1]/1.5)  
  
X\_train = Boston[train, -14]  
Y\_train = Boston[train, 14]  
X\_test = Boston[-train, -14]  
Y\_test = Boston[-train, 14]  
  
p1 = 13  
model1 = randomForest(X\_train, Y\_train, xtest = X\_test, ytest = Y\_test,ntree = 500 ,mtry = p1)  
  
p2 = p1/2  
model2 = randomForest(X\_train, Y\_train, xtest = X\_test, ytest = Y\_test, ntree = 500,mtry = p2)  
  
p3 = sqrt(p1)  
model3 = randomForest(X\_train, Y\_train, xtest = X\_test, ytest = Y\_test, ntree = 500,mtry = p3)  
  
plot(1:500, model1$test$mse, col = "blue", type = "l", xlab = "Number of Trees",   
 ylab = "Test MSE", ylim = c(10, 19))  
lines(1:500, model2$test$mse, col = "green", type = "l")  
lines(1:500, model3$test$mse, col = "red", type = "l")  
legend("topright", c("m=p", "m=p/2", "m=sqrt(p)"), col = c("blue", "green", "red"), lty = 1)

For each of the models, as the number of trees generated increased, the mse values trends downward. This effect was very strong at first and eventually weaned off and stablized around the ntrees = 100 mark. The model corresponding to the red line has the m equal to the square root of the number of predictors experienced the most success and its MSE was consistently smaller than the other model's. The model corresponding to the green line with m equal to the number of predictors/2 did second best. Lastly the model corresponding to the blue line did significantly word than the other two. This model corresponds to the bagging model, where all of the predictors are considered instead of a contrained and random amount.

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

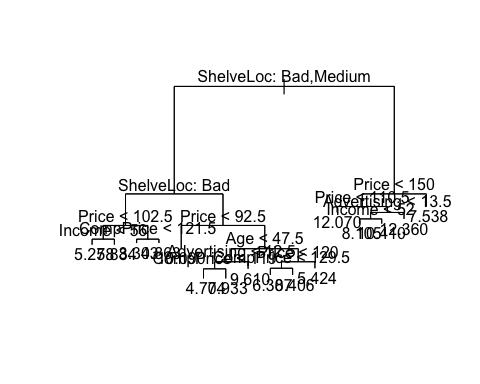
attach(Carseats)  
  
#train = sample(nrow(Carseats), nrow(Carseats)/2)  
#Train2 = Carseats[train, ]  
#Test2 = Carseats[-train, ]  
  
  
train = sample(dim(Carseats)[1], dim(Carseats)[1]/2)  
Train = Carseats[train, ]  
Test = Carseats[-train, ]

Fit a regression tree to the training set. Plot the tree, and interpret the results. What test MSE do you obtain?

library(tree)  
#fit regression tree  
cs.tree = tree(Sales~. , data = Train)  
summary(cs.tree)

##   
## Regression tree:  
## tree(formula = Sales ~ ., data = Train)  
## Variables actually used in tree construction:  
## [1] "ShelveLoc" "Price" "Income" "CompPrice" "Age"   
## [6] "Advertising"  
## Number of terminal nodes: 16   
## Residual mean deviance: 1.904 = 350.4 / 184   
## Distribution of residuals:  
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## -3.1760 -0.8390 0.1216 0.0000 0.9770 3.7670

#Plot  
plot(cs.tree)  
text(cs.tree , pretty =0)



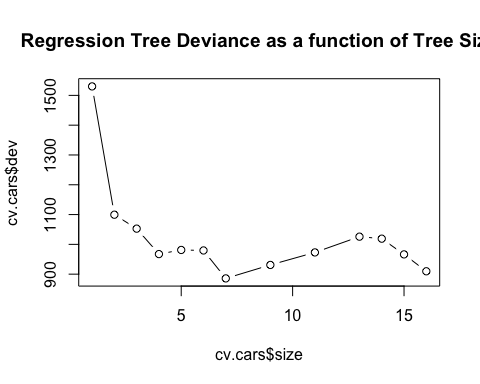
#mean squared error  
pred = predict(cs.tree, Test)  
mean((Test$Sales - pred)^2)

## [1] 4.735924

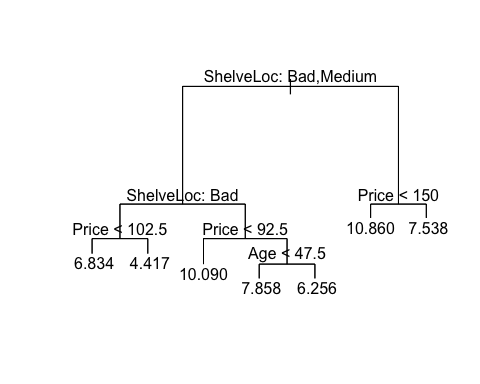
We obtained a test MSE equal to 4.735. The initial split that yielded the most gain in information was the whether or not the ShelveLoc variable was Bad or Medium. If it was, then whether or not ShelveLoc was Bad was the next greatest determinant. If the answer was yes, it came down to price, the either income or CompPrice. If the answer was no, it came down to Price, then possibly Age, Advertising, and CompPrice. On the right hand of the tree (if the ShelveLock was not Bad or Medium), Price, Advertising, then Income were used to determine the predicted Sales value. Predicted values on this side of the tree tended to be larger, with the largest being 12.360 if 110.5 < Price < 150, and Advertising >= 13.5.

Use cross-validation in order to determine the optimal level of tree complexity. Does pruning the tree improve the test MSE?

#using cross validation and set FUN to prune.tree  
cv.cars =cv.tree(cs.tree)  
  
#plot the deviance as a function tree size  
plot(cv.cars$size ,cv.cars$dev ,type='b', main='Regression Tree Deviance as a function of Tree Size')



#via observation. best is when size = 7  
prunned.cs = prune.tree(cs.tree, best = 7)  
plot(prunned.cs)  
text(prunned.cs, pretty = 0)



#create predictions so we can calculate and compare the MSE of the pruned with unprunned  
pred2 = predict(prunned.cs, Test)  
mean((Test$Sales - pred2)^2)

## [1] 5.241447

Based on the first plot above, the optimal tree size is 7. We create a new prunned tree from the original tree with best=7 and predict the test set's values, receiving an MSE of 5.241. Compared to the original tree's MSE of 4.735, this MSE is larger.

Use the bagging approach in order to analyze this data. What test MSE do you obtain? Use the importance() function to determine which variables are most important.

#generate model. mtry = 10 because there are 10 predictors in data. Importance = TRUE so we can obtain importances of features  
bag.cs= randomForest(Sales~.,data=Train, mtry=10,importance =TRUE)  
  
#make predictions  
pred3 = predict(bag.cs, Test)  
mean((Test$Sales - pred3)^2)

## [1] 3.192846

#get feature importance   
importance(bag.cs)

## %IncMSE IncNodePurity  
## CompPrice 18.7849589 110.286494  
## Income 6.0976650 85.922253  
## Advertising 10.9328154 70.382816  
## Population -0.4892713 49.622888  
## Price 45.8306702 310.347726  
## ShelveLoc 72.7990807 596.414587  
## Age 19.3211837 178.446509  
## Education 1.8198728 40.352724  
## Urban 0.1568143 8.899879  
## US 0.3292510 6.459907

Using the bagging method implemented via the randomForest function with mtry = number of predictor variables, we obtain a model that has an MSE equal to 3.15 when predicting sales on the test set. We see that ShelveLoc and Price are the most important predictors, and CompPrice/Age come close to tied for third. Removing them increases, on average, the MSE the most.

Use random forests to analyze this data. What test MSE do you obtain? Use the importance() function to determine which variables are most important. Describe the effect of m, the number of variables considered at each split, on the error rate obtained.

#generate forest. since we are performing regression, use mtry = 10/2 = 5. Importance = TRUE so we can obtain importances of features  
rf.cs = randomForest(Sales ~ ., data = Train, mtry = 5, importance = TRUE)  
  
#make predicitons. Calc MSE  
pred4 = predict(rf.cs, Test)  
mean((Test$Sales - pred4)^2)

## [1] 3.13654

#importances of features  
importance(rf.cs)

## %IncMSE IncNodePurity  
## CompPrice 13.0461197 119.449764  
## Income 3.8685629 96.225078  
## Advertising 10.0027971 94.150361  
## Population -1.6812705 76.773774  
## Price 36.5544778 294.536126  
## ShelveLoc 57.2472615 493.625325  
## Age 16.8877868 202.251707  
## Education 0.3014632 49.402108  
## Urban -3.6072095 9.699726  
## US 2.4797684 9.562143

Using the bagging method implemented via the randomForest function with mtry = number of predictor variables, we obtain a model that has an MSE equal to 3.143. when predicting sales on the test set. This is about the same as the bagging method used in part d. Again, Price and ShelveLoc were the most importance featurse since taking them away cause the most (on average) increase in MSE and node impurity. Age was third but significantly lower with a value of IncMSE at 15.7.