Apurva Shrivastava
UIN: 925009508
ISEN 613 - 600

Assignment# 6

Problem 1

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(ISLR)
> set.seed(1)
> train_data <- sample(1:nrow(Carseats), nrow(Carseats)/2)
> Carseats.train <- Carseats[train_data, ]
> Carseats.test <- Carseats[-train_data, ]
>
```

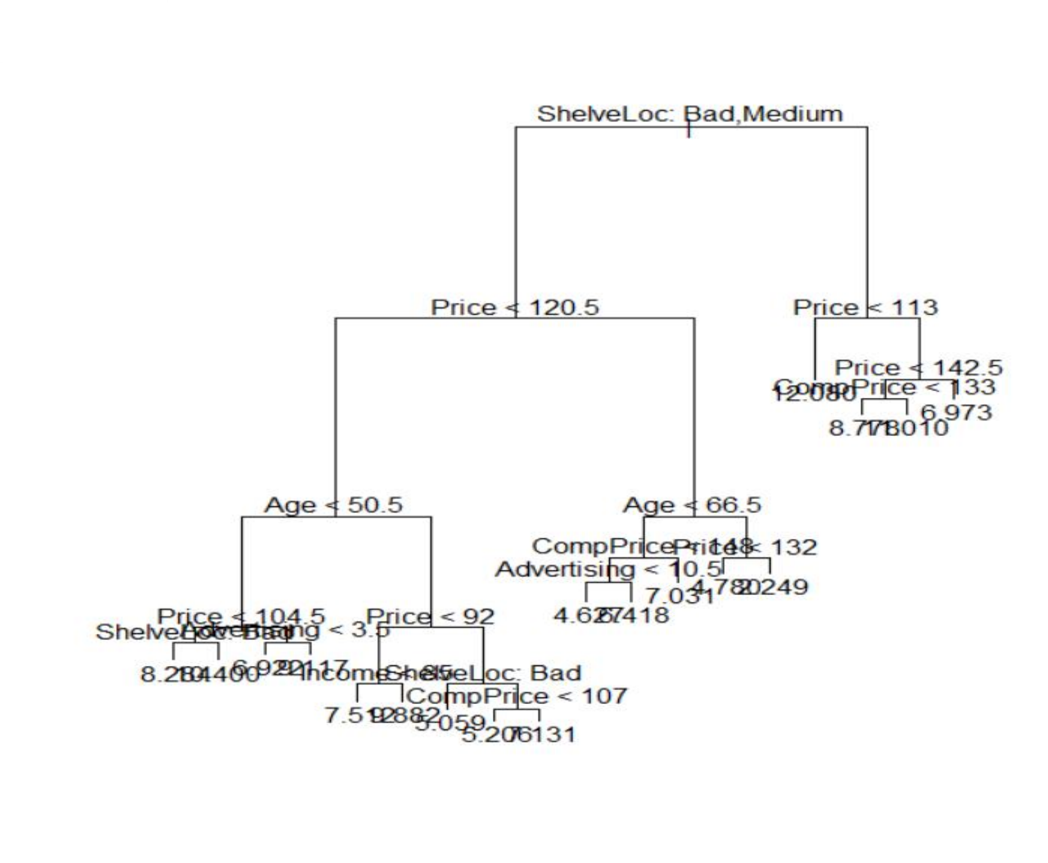
(b) Fit a regression tree to the training set. Plot the tree, and interpret the results. Then compute the test MSE.

```
> library(tree)
> 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" "Income"      "CompPrice"
Number of terminal nodes:  18
Residual mean deviance:  2.36 = 429.5 / 182
Distribution of residuals:
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-4.2570 -1.0360  0.1024  0.0000  0.9301  3.9130
>
```

```
> plot(tree.carseats)
> text(tree.carseats, pretty = 0)
>
```

Apurva Shrivastava
UIN: 925009508
ISEN 613 - 600



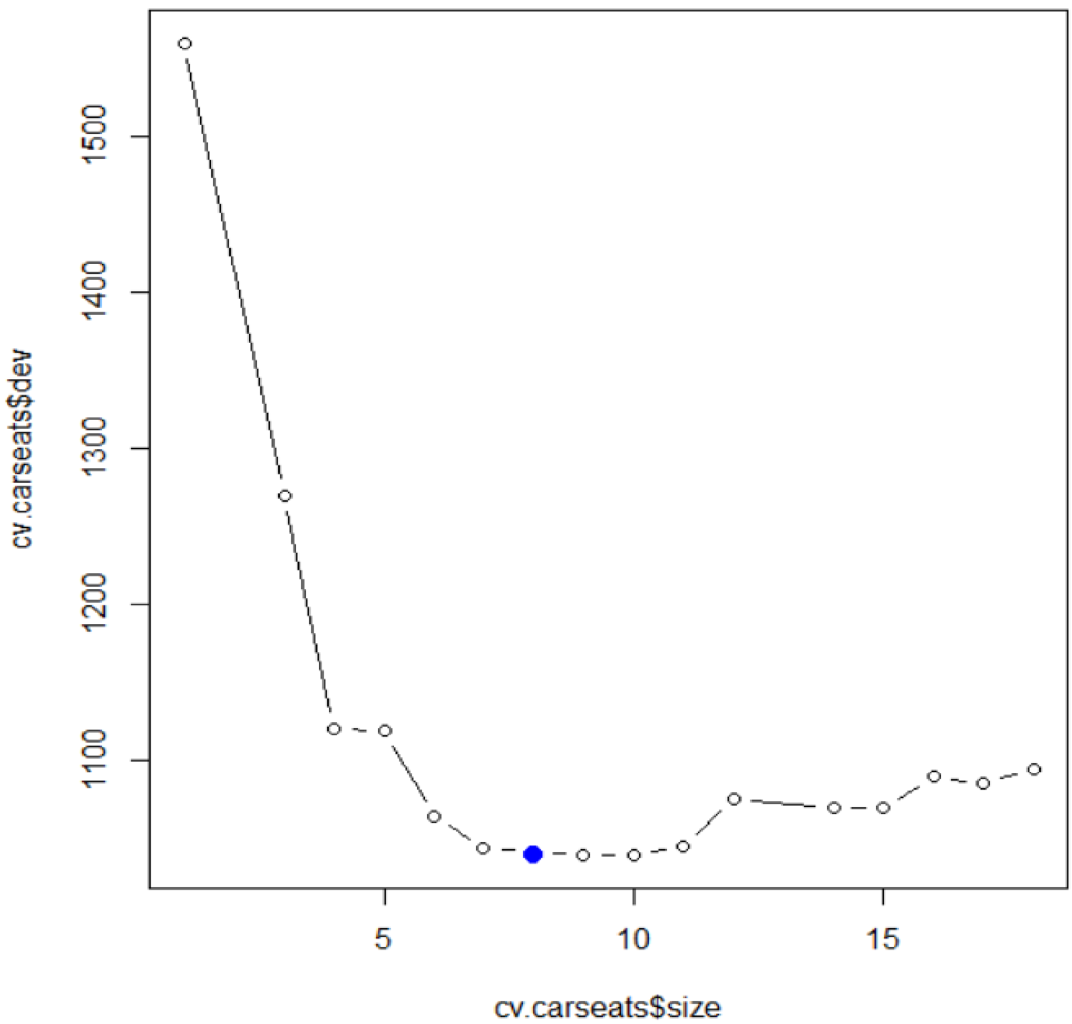

```
>
> ycap <- predict(tree.carseats, newdata = Carseats.test)
> mean((ycap - Carseats.test$Sales)^2)
[1] 4.148897
>
```

We can see that MSE is **4.14**

(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 = "blue", cex = 2, pch = 20)
>
```

Apurva Shrivastava
UIN: 925009508
ISEN 613 - 600

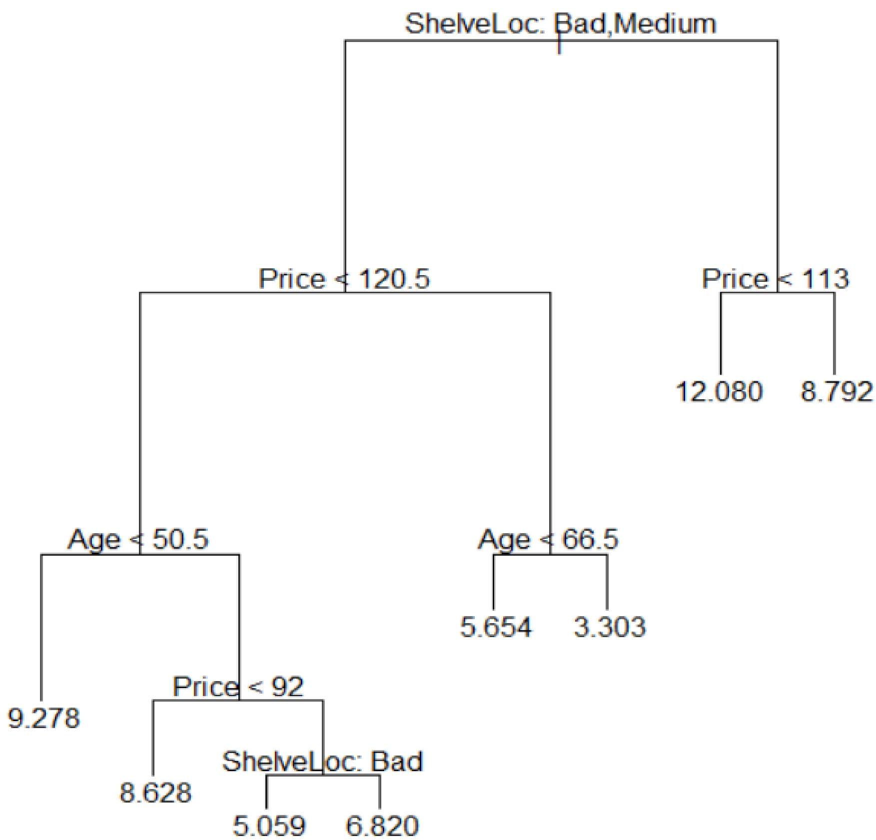


Pruning the tree:

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 = 8)
>   plot(prune.carseats)
>   text(prune.carseats, pretty = 0)
>
```

Apurva Shrivastava
UIN: 925009508
ISEN 613 - 600




```
> ycap <- predict(prune.carseats, newdata = Carseats.test)
>   mean((ycap - Carseats.test$Sales)^2)
[1] 5.09085
>
```

MSE for the prune tree is 5.09. As we can see pruning the tree has increased the MSE from 4.14 to 5.09.

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

```
> bag.carseats <- randomForest(Sales ~ ., data = Carseats.train, mtry = 10, ntree = 500, importance = TRUE)
> ycap.bag <- predict(bag.carseats, newdata = Carseats.test)
> mean((ycap.bag - Carseats.test$Sales)^2)
[1] 2.57295
>
```

We see that bagging decreases the test MSE to **2.57**

Apurva Shrivastava
UIN: 925009508
ISEN 613 - 600

To determine which variable is most important we use **importance()** method.

```
> importance(bag.carseats)
              %IncMSE IncNodePurity
CompPrice   16.297100    130.796172
Income       4.828604     78.208046
Advertising 14.688260    124.933965
Population   2.251331     58.882291
Price       57.016882    517.991476
ShelveLoc   46.096614    319.334615
Age         22.019714    194.098835
Education    2.966678     40.162590
Urban       -2.100855      8.873266
US           7.003729     16.330146
>
```

From the result, we conclude that price and ShelveLoc are the two most important variables. (due to higher %IncMSE values)

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

```
> rf.carseats <- randomForest(Sales ~ ., data = Carseats.train, mtry = 3, ntree = 500, importance = TRUE)
> ycap.rf <- predict(rf.carseats, newdata = Carseats.test)
> mean((ycap.rf - Carseats.test$Sales)^2)
[1] 3.326674
>
```

In this case, test MSE is **3.32**

```
> importance(rf.carseats)
               %IncMSE IncNodePurity
CompPrice    8.0728869     130.15824
Income       3.9001054     121.88353
Advertising 12.4449886     138.63178
Population  -0.9082328      97.30322
Price       36.0489662     384.19473
ShelveLoc   31.2271291     242.38514
Age         16.0387820     195.99128
Education    2.1311852      65.66093
Urban       -3.0169350      16.22909
US           5.2815786      33.08008
>
```

From the results, we see that Price and ShelveLoc are the two most important variables (due to high %IncMSE values).

Apurva Shrivastava
UIN: 925009508
ISEN 613 - 600

Problem 2

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.

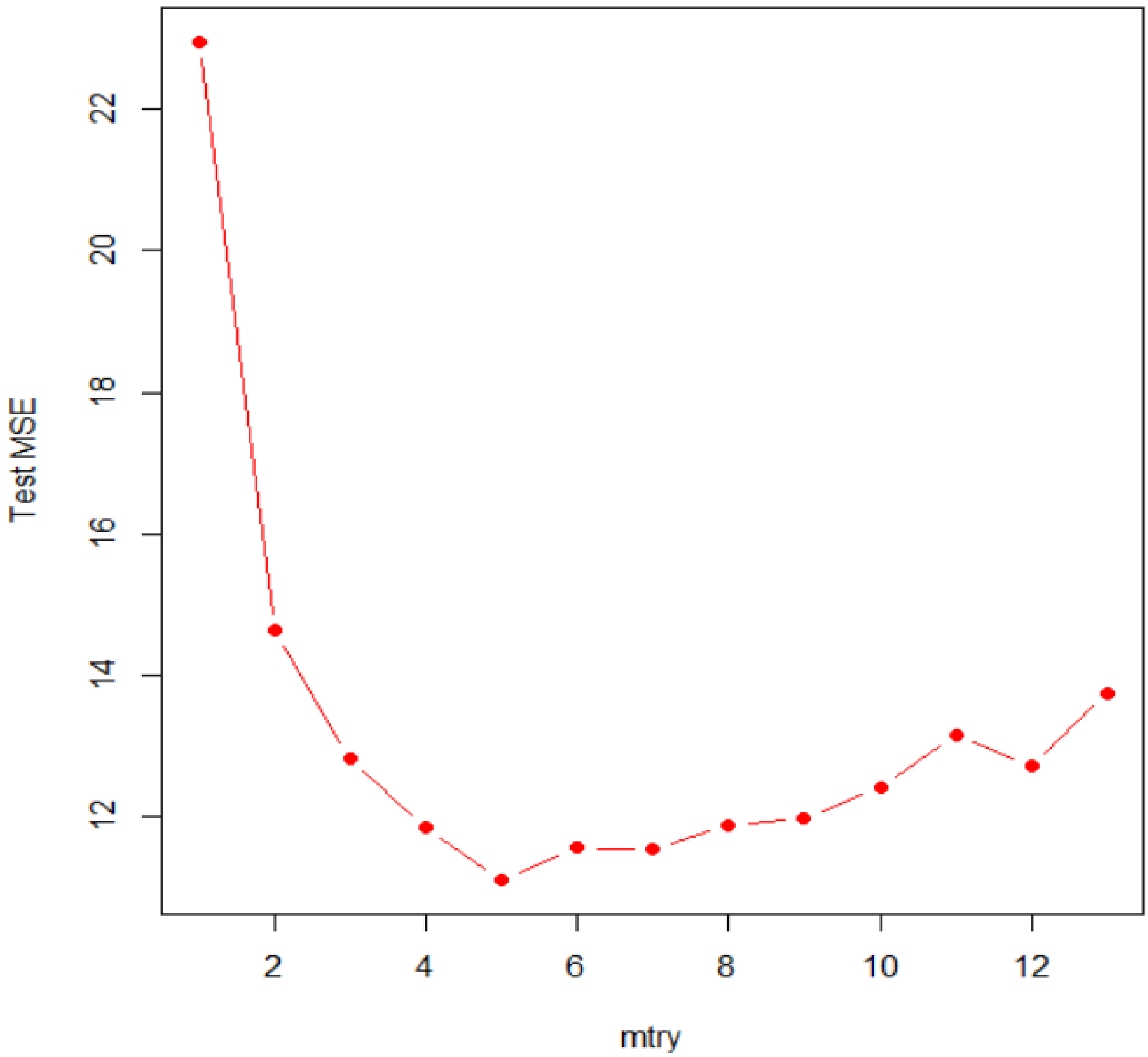
Creating the test and train data sets and mtry and test error vectors

```
> library(MASS)
> attach(Boston)
> set.seed(1)
> train_data <- sample ( 1: nrow(Boston), nrow(Boston) / 2)
> Boston.train <- Boston[train_data,]
> Boston.test <- Boston[-train_data,]
>
> mtry <- c(1:13)
> test.error <- souble(13)
Error in souble(13) : could not find function "souble"
> test.error <- double(13)
> mtry
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13
> test.error
 [1] 0 0 0 0 0 0 0 0 0 0 0 0 0
> 
```

```
> for(i in 1:13){
+ rf.Boston <- randomForest(medv ~ ., Boston.train, mtry = i, ntree = 100, importance = T)
+ ycap.rf <- predict(rf.Boston, Boston.test)
+ test.err[i] <- mean((ycap.rf - Boston.test$medv)^2)
+ }
> test.err
 [1] 22.94959 14.63641 12.82954 11.85036 11.11480 11.56394 11.53832 11.88346 11.99004 12.42750 13.16399 12.73479
[13] 13.73848
> df <- cbind(mtry, test.err)
> plot(df, col = 'red', type = 'b', pch =19, xlab ='mtry', ylab = 'Test MSE')
> 
```

<u>Graph Plot:</u>

Apurva Shrivastava
UIN: 925009508
ISEN 613 - 600



Looking at the graph, we observe that Test MSE is extremely high when mtry =1 and then decreases as the value of mtry increases. However, with increasing mtry Test MSE again starts to increase.

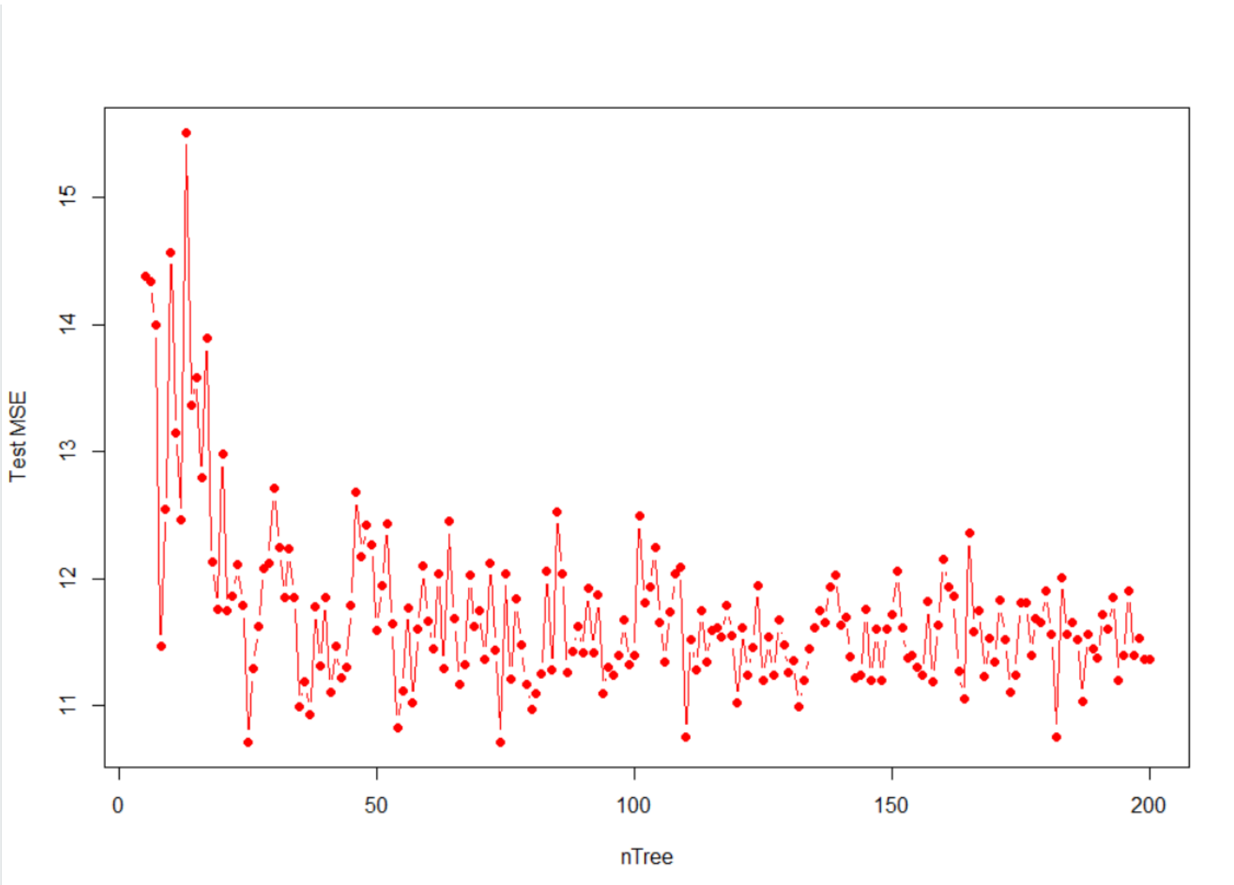
The lowest Test MSE is when **mtry = 5.**

(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.

```
> ntree <- c(5:200)
> test.err <- double(196)
>
> for (i in 5:200){
+ rf.Boston <- randomForest(medv ~ ., Boston.train, mtry = 6, ntree = i, importance = T)
+ ycap.rf <- predict(rf.Boston, Boston.test)
+ test.err[i] <- mean((ycap.rf - Boston.test$medv)^2)
+ }
```

Removing the 0.00 values in test.err and creating data frame nTree and Test MSE:

```
> test.err <- subset(test.err, test.err != 0)
> df <- cbind(ntree, test.err)
> plot(df, col = 'red', type ='b', pch = 19, xlab ='nTree', ylab ='Test MSE')
> |
```



From the plot, we observe that when nTree is below 50 Test MSE is on the higher side. However, on increasing the value of nTree Test MSE is reduced with minimum Test MSE coming for nTree value of **70.**