Assignment #6

Question 1:

In the lab, a classification tree was applied to the Carseats data set after converting Sales into a qualitative response variable. Now we will seek to predict Sales using regression trees and related approaches, treating the response as a quantitative variable.

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

```
library(ISLR)
str(Carseats)
```

```
## 'data.frame':    400 obs. of  11 variables:
##  $ Sales      : num  9.5 11.22 10.06 7.4 4.15 ...
##  $ CompPrice  : num  138 111 113 117 141 124 115 136 132 132 ...
##  $ Income     : num  73 48 35 100 64 113 105 81 110 113 ...
##  $ Advertising: num  11 16 10 4 3 13 0 15 0 0 ...
##  $ Population : num  276 260 269 466 340 501 45 425 108 131 ...
##  $ Price      : num  120 83 80 97 128 72 108 120 124 124 ...
##  $ ShelveLoc  : Factor w/ 3 levels "Bad","Good","Medium": 1 2 3 3 1 1 3 2
3 3 ...
##  $ Age        : num  42 65 59 55 38 78 71 67 76 76 ...
##  $ Education  : num  17 10 12 14 13 16 15 10 10 17 ...
##  $ Urban      : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 1 2 2 1 1 ...
##  $ US         : Factor w/ 2 levels "No","Yes": 2 2 2 2 1 2 1 2 1 2 ...
```

```
# Split data with the ratio 50:50
set.seed(1)
split<- sample(c(rep(0, 0.5 * nrow(Carseats)), rep(1, 0.5 * nrow(Carseats))))
training <- Carseats[split == 0, ]
test <- Carseats[split == 1, ]
dim(training)
```

```
## [1] 200  11
```

```
dim(test)
```

```
## [1] 200  11
```

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

Fitting a regression tree

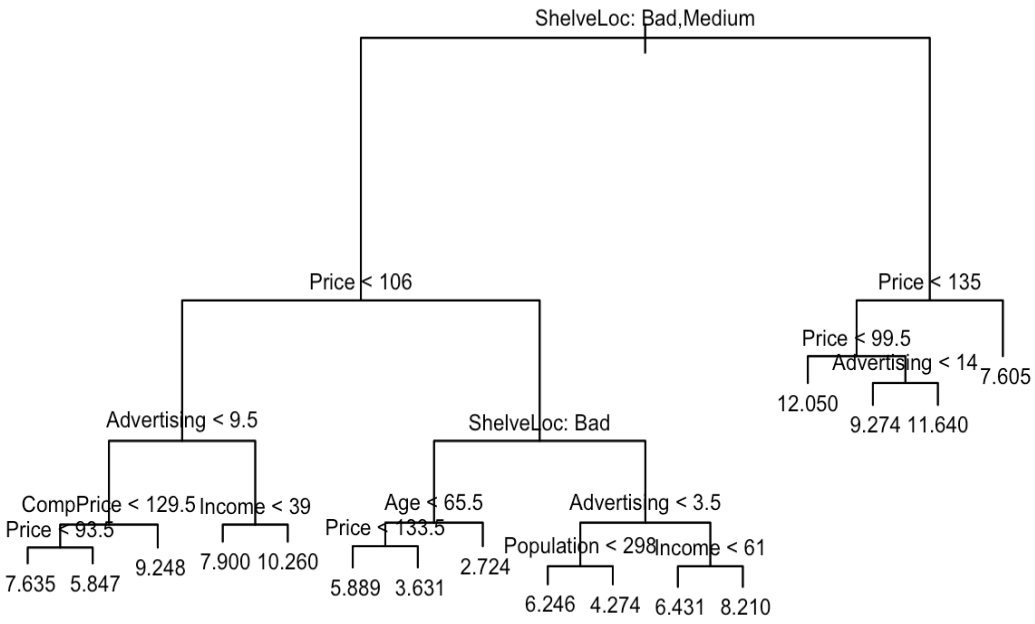
```
library(tree)

# Fit a regression tree to the training set
tree.fit <- tree(Sales~., data = training)
summary(tree.fit)
```

```
## Regression tree:
## tree(formula = Sales ~ ., data = training)
## Variables actually used in tree construction:
## [1] "ShelveLoc"   "Price"       "Advertising" "CompPrice"   "Income"
## [6] "Age"         "Population"
## Number of terminal nodes:  16
## Residual mean deviance:  2.371 = 436.3 / 184
## Distribution of residuals:
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -4.16500 -1.10000 -0.09329  0.00000  1.18600  4.22500
```

Plotting the tree

```
# Plot the tree
plot(tree.fit)
text(tree.fit, pretty = 0, cex = 0.7)
```



The tree indicates that the most important indicator of Sales appears to be shelving location, since the first branch differentiate locations from Bad and Medium locations. However, the number of terminal nodes can be different since it depends on how we split a training and test set. In this case the number of terminal nodes is equal to 16.

Testing MSE

```
# Test MSE
y.pred <- predict(tree.fit, newdata = test)
y.test <- test$Sales
tree.mse <- mean((y.pred - y.test)^2)
sprintf("%s is %0.4f", "Test MSE of tree regression model", tree.mse)
```

```
## [1] "Test MSE of tree regression model is 4.8532"
```

The test MSE of tree regression model is 4.8532.

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

Using cross-validation to determine the optimal level of tree complexity

```
# Use cross-validation to determine the optimal level of tree complexity
cv.carseats <- cv.tree(tree.fit)
cv.carseats
```

```
## $size
##  [1] 16 15 14 13 12 11 10  9  8  7  6  5  4  3  2  1
##
## $dev
##  [1]  931.1620  933.6426  930.3589  938.4850  930.8773  912.4569  907.1038
##  [8]  933.1485  903.7775  902.7721  964.2021 1022.3441 1113.6118 1143.6108
## [15] 1283.2013 1596.5150
##
## $k
##  [1]      -Inf  20.83166  22.37290  23.74626  27.09951  27.91791  28.74719
##  [8]  30.13849  30.89475  35.18539  57.20716  73.32850 107.43225 109.97124
## [15] 184.56084 344.79116
##
## $method
## [1] "deviance"
##
## attr(,"class")
## [1] "prune"         "tree.sequence"
```

```
cv.min <- which.min(cv.carseats$dev)
sprintf("%s is at point %i", "The lowest deviance", cv.min)
```

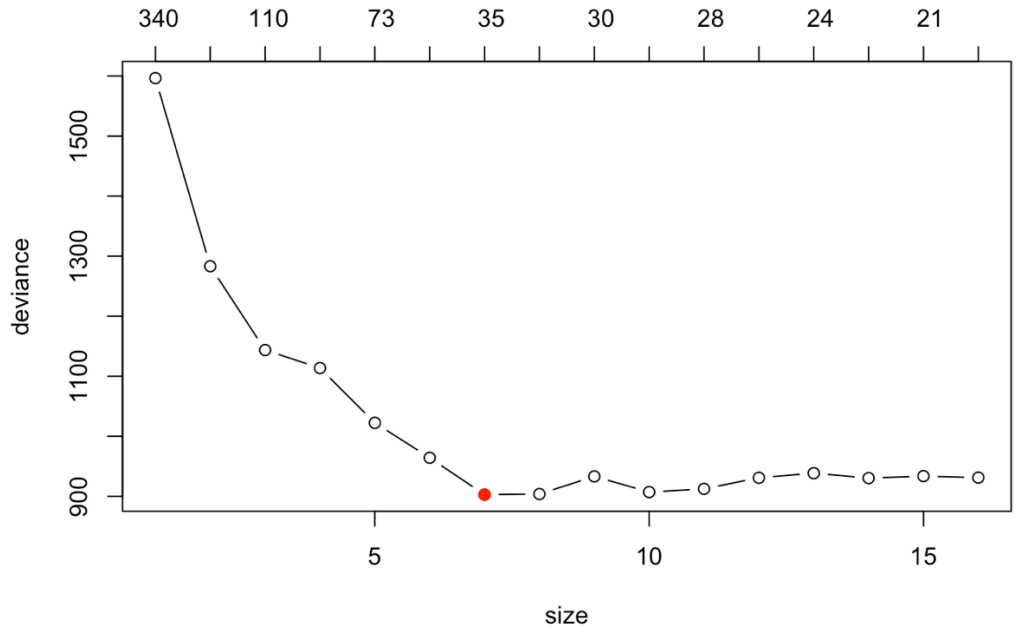
```
## [1] "The lowest deviance is at point 10"
```

```
cv.size<- which(cv.carseats$size == cv.min)
sprintf("%s is %i", "The tree size using cross-validation", cv.size)
```

```
## [1] "The tree size using cross-validation is 7"
```

The lowest deviance is at point 10, which equals 902.7721. At this point, the optimal level of tree complexity is 7.

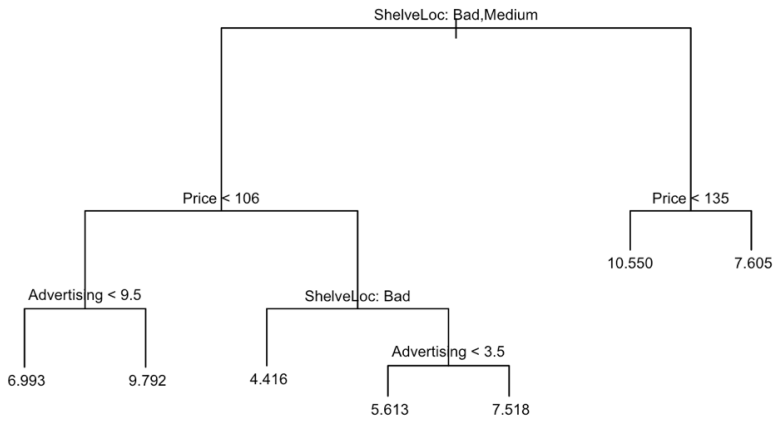
```
# Plot showing the optimal level
plot(cv.carseats,type="b")
points(cv.size, cv.carseats$dev[cv.min], col = "red", cex = 1.5, pch = 20)
```



Therefore, the tree of size 7 is selected by cross-validation, so we will prune the tree to obtain the 7-node tree.

Prunning the tree with the optimal level of tree complexity obtained from cross-validation

```
# Prune the tree to obtain the 7-node tree
prune.carseats <- prune.tree(tree.fit, best = cv.size)
# Plot the tree
plot(prune.carseats)
text(prune.carseats, pretty = 0, cex = 0.7)
```



Testing MSE

```
# Test MSE
y.pred <- predict(prune.carseats, newdata = test)
y.test <- test$Sales
prune.tree.mse <- mean((y.pred - y.test)^2)
sprintf("%s is %0.4f", "Test MSE of pruned tree regression model", prune.tree
.mse)

## [1] "Test MSE of pruned tree regression model is 5.2293"
```

The test MSE of this pruned tree is 5.2293 which is greater than the regression tree model performance. Therefore, pruning the tree does not improve the test MSE since it increases the test MSE from 4.8532 to 5.2293.

(d) 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.

Using the bagging approach to analyze Carseats dataset

Since we have 10 predictors, mtry would be $m = p = 10$. So we would use mtry = 10

```
library(randomForest)
set.seed(1)
# Fitting the model using bagging approach
bag.carseats <- randomForest(Sales~., data = training, mtry = 10, importance
= TRUE )
bag.carseats
##
## Call:
##  randomForest(formula = Sales ~ ., data = training, mtry = 10,      import
ance = TRUE)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 10
##
##           Mean of squared residuals: 2.626527
##                     % Var explained: 66.34
```

Testing MSE

```
# Test MSE
y.pred <- predict(bag.carseats, newdata = test)
y.test <- test$Sales
bag.tree.mse <- mean((y.pred - y.test)^2)
sprintf("%s is %0.4f", "Test MSE of the bagged regression tree", bag.tree.mse
)

## [1] "Test MSE of the bagged regression tree is 2.5810"
```

The test MSE of this bagged regression tree is 2.5810.

Obtaining most important variables

```
importance(bag.carseats)

##                %IncMSE IncNodePurity
## CompPrice   22.0967673    170.794259
## Income      10.8928684     94.914657
## Advertising 24.8139784    200.290001
## Population   4.1384445     76.683150
## Price       48.8083385    394.702345
## ShelveLoc   56.3661707    416.016859
## Age         12.1771446     99.594812
## Education    4.1231386     45.262631
## Urban       -0.5921785      5.436113
## US           2.5592554      6.071524
```

Variable **"ShelveLoc"** is the most important variable using the bagging approach followed by variable Price.

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

Using random forests to analyze Carseats dataset

Since we have 10 predictors, mtry would be $m = \frac{10}{3} \approx 3.3$ when building a random forest of regression trees. So, we would use mtry = 3

```
library(randomForest)
set.seed(1)
# Fitting the model using random forest
rf.carseats <- randomForest(Sales~., data = training, mtry = 3, importance =
TRUE)
rf.carseats

##
## Call:
##  randomForest(formula = Sales ~ ., data = training, mtry = 3,      importa
nce = TRUE)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 3
##
##           Mean of squared residuals: 3.279166
##                     % Var explained: 57.97
```

Testing MSE

```
# Test MSE
y.pred <- predict(rf.carseats, newdata = test)
y.test <- test$Sales
rf.tree.mse <- mean((y.pred - y.test)^2)
```

```
sprintf("%s is %0.4f", "Test MSE of the random forest of regression tree", rf
.tree.mse)

## [1] "Test MSE of the random forest of regression tree is 3.2542"
```

The test MSE of this random forest of regression tree is 3.2542.

Obtaining most important variables

```
importance(rf.carseats)

##                %IncMSE IncNodePurity
## CompPrice   11.2615192     159.25081
## Income       4.9401807     129.65272
## Advertising 18.9598909     186.80680
## Population  -0.6956409     115.95449
## Price       30.9839118     325.34083
## ShelveLoc   36.2275128     319.46378
## Age          6.5238689     135.28333
## Education    2.1396693      63.46654
## Urban        1.0122455      13.20398
## US           2.8146414      22.09997
```

Variable **"ShelveLoc"** is the most important variable using the random forest followed by variable Price.

Conclusion:

The model using bagging approach has the lowest test error.