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 the data set into a training set and a test set

The total number of observations in the dataset is 400. We use the sample function to generate a random sample for train containing 200 observations. The R code generated is shown below.

```
> library(ISLR)
> set.seed(1)
> attach(Carseats)
> train<-sample(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. What test error rate do you obtain?

We use the *tree* function from the library tree to create a regression tree on the training data. The code and the output generated are shown below.

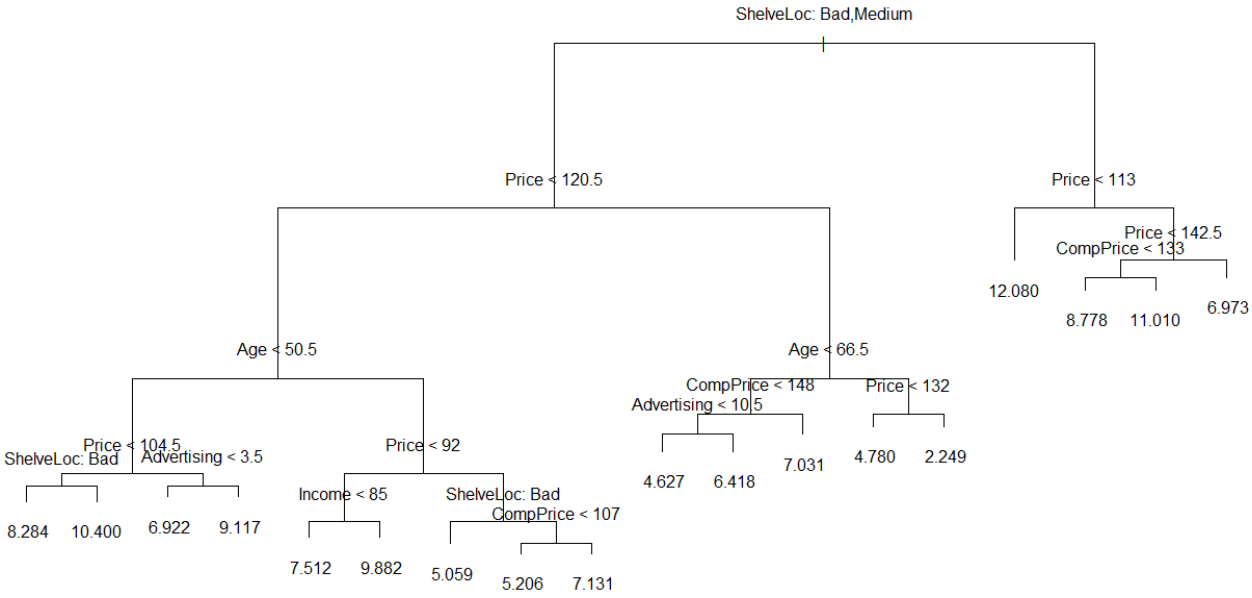
```
> library(tree)
> tree.carseats<-tree(Sales~.,Carseats,subset=train)
> summary(tree.carseats)

Regression tree:
tree(formula = Sales ~ ., data = Carseats, subset = train)
Variables actually used in tree construction:
[1] "ShelveLoc"   "Price"       "Age"         "Advertising" "Income"
[6] "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
```

Using the *plot* and *text* functions we plot the regression tree.

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

The variables used in the tree construction are *ShelveLoc*, *Price*, *Age*, *Advertising*, *Income* and *CompPrice*. The Sales is stratified into 18 regions (R_1, R_2… R_{18}). The *ShelveLoc* Bad, Medium are assigned to the left branch and the *ShelveLoc* Good is assigned to the right branch. Similarly, in the second level, the group is further subdivided by *Price* on both the branches. In each of the subsequent branches the groups are further subdivided by the variables that are of importance in the construction of the tree. The plot of the tree is given below:



Now we calculate the *Sales.Hat* that is the predicted values using the regression tree on the test dataset.

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

The mean squared error (Test MSE) is *4.15*

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

We use the *cv.tree* function to cross-validate the model since large tree classifiers tend to over-fit the training data. The code and output is as below:

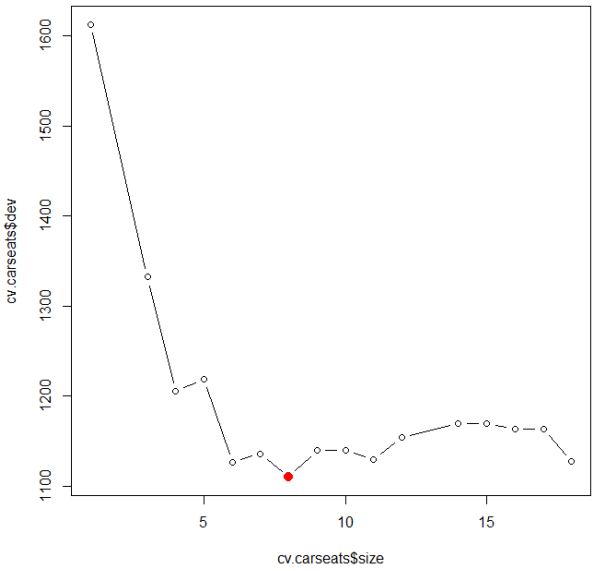
```
> cv.carseats<-cv.tree(tree.carseats,FUN=prune.tree)
> names(cv.carseats)
[1] "size"   "dev"    "k"      "method"
> cv.carseats
$size
 [1] 18 17 16 15 14 12 11 10  9  8  7  6  5  4  3  1

$dev
 [1] 1023.659 1033.957 1033.919 1023.382 1023.382 1038.603 1062.313 1033.461 1033.461
[10] 1040.792 1030.011 1069.573 1125.023 1143.986 1179.822 1566.489

$k
 [1]      -Inf  15.48181  15.53599  18.69038  18.74886  21.05038  23.79480  25.78579
 [9]  26.01210  30.10435  32.74801  53.28569  72.33061  78.19599 141.73781 251.22901

$method
[1] "deviance"

attr(,"class")
[1] "prune"         "tree.sequence"
```



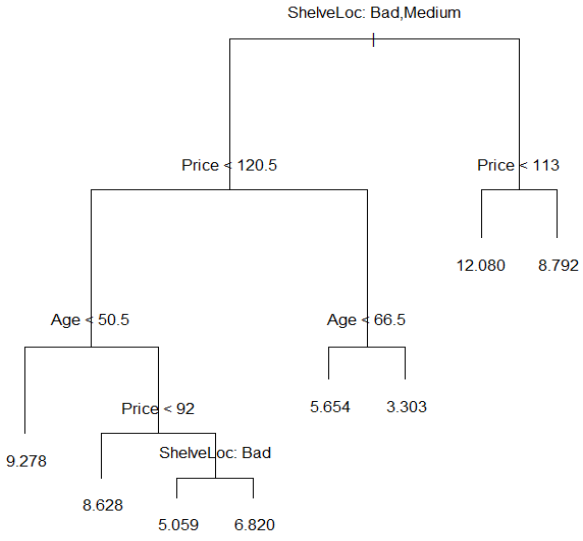
We then plot the tree size versus the deviance plot to figure out the optimal number regions that the predictor space should be divided into.

In the graph between deviance and size we see that the least deviance corresponds to size 8.

So, we select the best size as 8 when we prune the tree to obtain an 8-node tree. Below is the code to obtain the pruned tree

```
> prune.carseats= prune.tree(tree.carseats,best=8)
> plot(prune.carseats)
> text(prune.carseats,pretty=0,xpd=TRUE)
> summary(prune.carseats)

Regression tree:
snip.tree(tree = tree.carseats, nodes = c(39L, 11L, 8L, 18L,
7L, 10L))
Variables actually used in tree construction:
[1] "ShelveLoc" "Price"     "Age"
Number of terminal nodes:  8
Residual mean deviance:  3.363 = 645.7 / 192
Distribution of residuals:
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-5.28400 -1.12000  0.02695  0.00000  0.99010  4.59800
```



After the pruning has been performed we see that the terminal nodes have been reduced from 18 to 8. The pruned tree increases the interpretability of the model as compared to the original tree. If we observe the summary of the pruned model, the variables used have been reduced from 6 to 3.

The pruned tree test MSE is calculated as below:

```
> prune.sales.Hat<-predict(prune.carseats,carseats.test)
> mean((prune.sales.Hat - carseats.test$Sales)^2)
[1] 5.09085
```

Pruning does not improve the test MSE as it has increased from *4.15* to *5.09*.

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

Bagging is a technique used to improve the prediction accuracy of the regression tree. We use the *randomForest* function with *mtry=10* (including all the predictor variables).

```
> bagging.sales<-randomForest(Sales~.,data=carseats.train,mtry=10,importance=T)
> bagging.sales

Call:
 randomForest(formula = Sales ~ ., data = carseats.train, mtry = 10,      importance = T
)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 10

          Mean of squared residuals: 2.820522
                    % Var explained: 63.05
> bagging.pred = predict(bagging.sales, carseats.test)
> mean((bagging.pred-carseats.test$Sales)^2)
[1] 2.57295
> importance(bagging.sales)
              %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
```

The test error rate obtained is *2.57295* is lower than the test error of both the normal regression tree as well as the pruned tree. The *importance* function tells us that *Price*, *ShelveLoc* and *Age* are the three most important variables in the tree.

(e) Use random forests to analyze this data. What test error rate 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.

The total number of predictors in the data set is *p=10*. However, while implementing the random forests, we have to consider random selection of m predictors (*m<p* and $m=\sqrt{p}$). Since *p=10*, the ideal value for *m=sqrt(10)≈3.16*. By using the random forest implementation with *m=3*:

```
> rf.carseats= randomForest(Sales~.,data=carseats.train,mtry=3,importance=TRUE)
> saleshat.rf <- predict(rf.carseats, carseats.test)
> mean((saleshat.rf - carseats.test$Sales)^2)
[1] 3.276259
```

The test MSE is *3.276259*, which is higher than that obtained by using bagging. This increase in test MSE is due to the reduction in number of predictor variables used at each split. As we increase the value of m the test MSE decreases, however the trees obtained are highly correlated which does not aid in the reduction of variance.

By using the importance function we can conclude that *Price*, *ShelveLoc* and *Age* are the important variables.

```
> importance(rf.carseats)
               %IncMSE IncNodePurity
CompPrice    6.0194482     129.34280
Income       5.1108217     123.32831
Advertising 13.7584451     139.55570
Population  -0.1780729      99.93286
Price       39.4861834     387.76316
ShelveLoc   31.4141129     240.46217
Age         18.2368950     199.15419
Education    3.3365768      67.61640
Urban        0.6825523      14.37889
US           7.0717217      31.98968
```