HW4

Liyuan Tang

5/17/2020

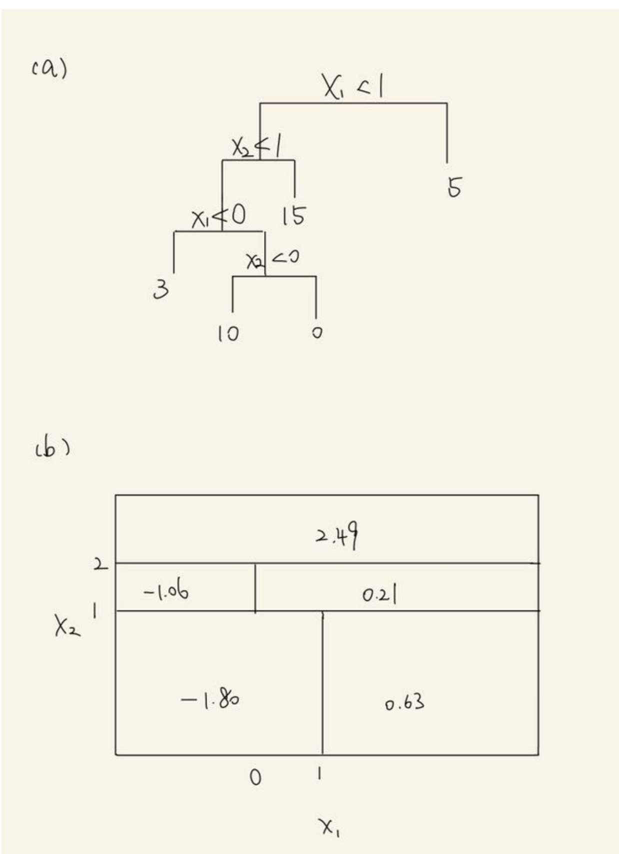
Question 2

Based on algorithm 8.2, we first set $\hat{f}(x) = 0$ and $r_i = y_i$ for all i in the training set. And each time, we will update $\hat{f}(x)$ as $\hat{f}(x) + \lambda \hat{f}^b(x)$

So, the output of the boosted model is $\hat{f}(x) = \sum_{b=1}^{B} \lambda \hat{f}^b(x)$. Every $\hat{f}^b(x)$ is fitted by a depth-one tree, we will get 1 split and 2 terminal nodes. And $\hat{f}^b(x) = c_1 I(x_b < t)$. Since the split only depends on one predictor, the summation of $\hat{f}^b(x)$ is additive.

Question 4

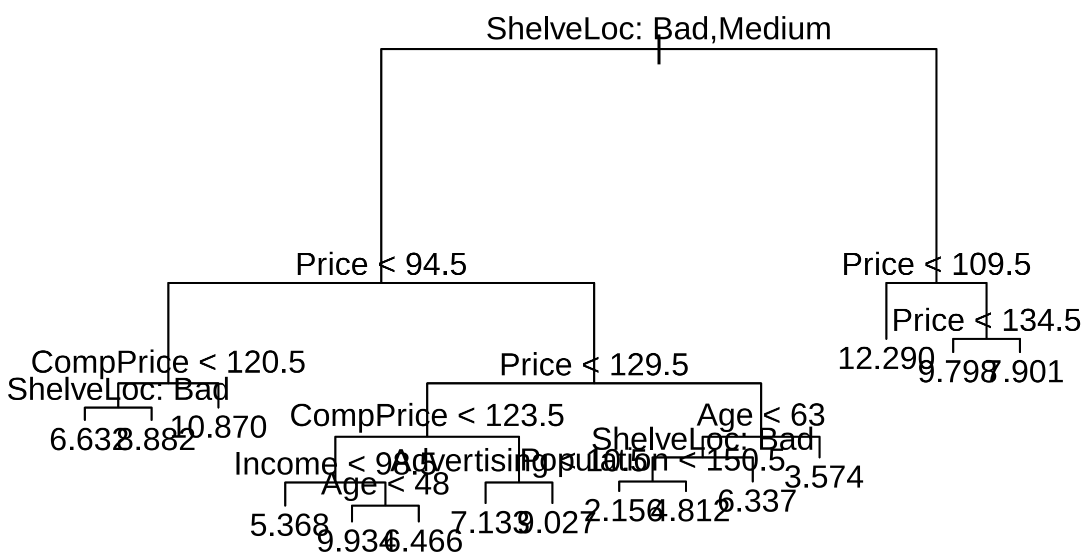
See the picture below.



Question 8

(b)

```r
source('Carseats-split.r')
library(tree)
tree.carseats=tree(Sales~., data = Carseats.train)
plot(tree.carseats)
text(tree.carseats,pretty=0)
```




```r
summary(tree.carseats)
```

```
##
## Regression tree:
## tree(formula = Sales ~ ., data = Carseats.train)
## Variables actually used in tree construction:
## [1] "ShelveLoc"   "Price"       "CompPrice"   "Income"      "Age"
## [6] "Advertising" "Population"
## Number of terminal nodes:  15
## Residual mean deviance:  2.506 = 714.3 / 285
## Distribution of residuals:
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -3.94800 -1.03000 -0.02731  0.00000  1.14400  3.97600
```

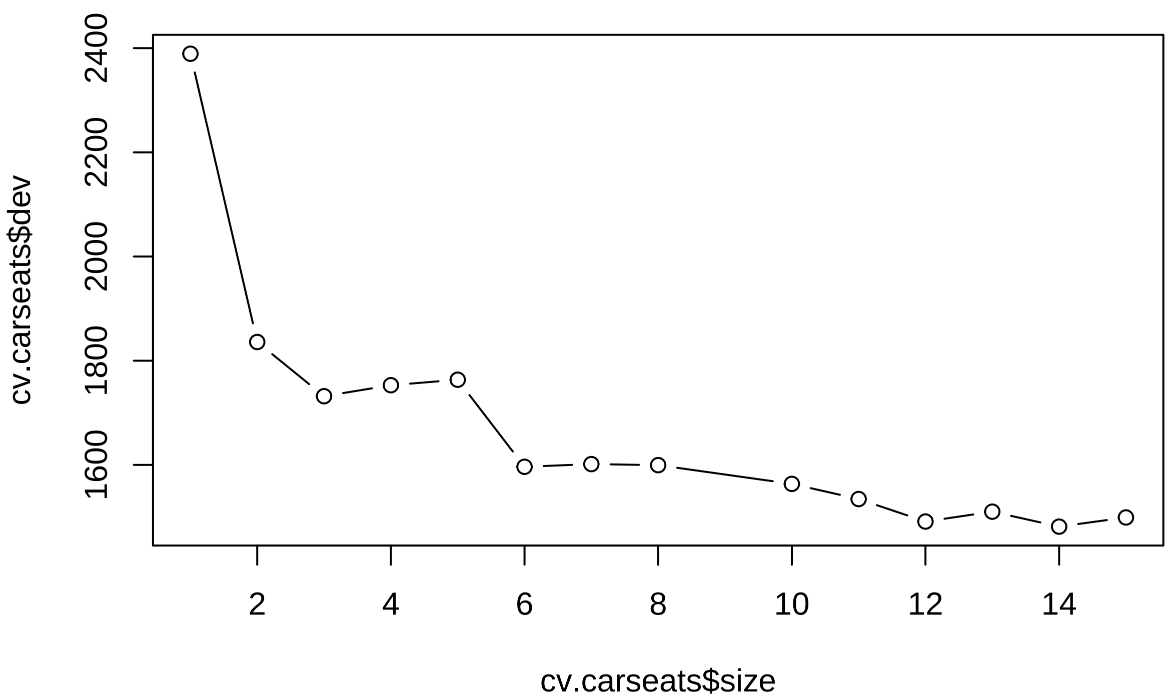
```r
# test MSE
pred.val = predict(tree.carseats, newdata = Carseats.test)
t.mse = sum((Carseats.test$Sales - pred.val)^2) / length(pred.val)
```

The test MSE is 4.9659091. 'ShelveLoc' is the most important factor in determining 'Sales'. In this graph, it shows that a good quality of shelving location for the car seats at each site has more unit sales comparing to the bad or medium shelving location. Then, if the child car seats have equal quality of shelving location, the price will affect the number of unit sales.

For part (c), (d) and (e), I used set.seed(1)

(c)

```r
set.seed(1)
cv.carseats <- cv.tree(tree.carseats)
plot(cv.carseats$size,cv.carseats$dev,type='b')
```

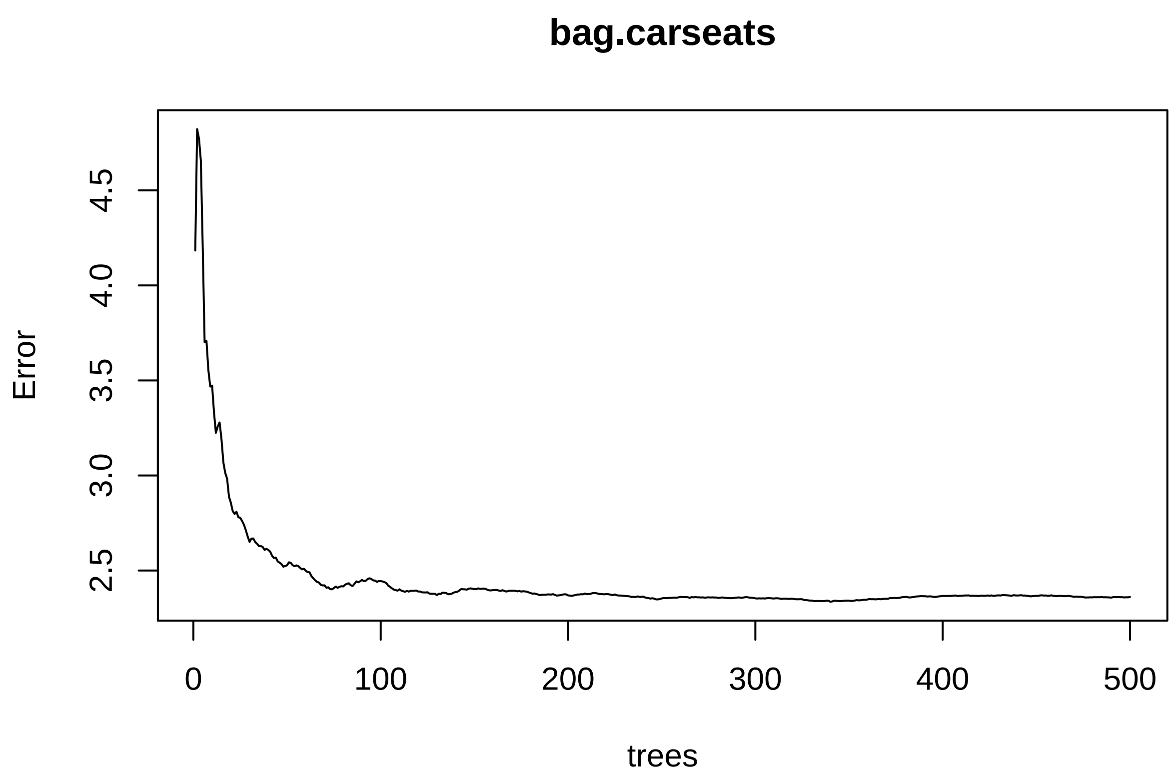



```r
tree.opt = cv.carseats$size[which.min(cv.carseats$dev)]

# Pruning the tree
prune.carseats=prune.tree(tree.carseats,best=tree.opt)
pred.val.improved = predict(prune.carseats, newdata = Carseats.test)
t.mse.improved = sum((Carseats.test$Sales - pred.val.improved)^2) / length(pred.val.improved)
```

The optimal level of tree complexity is 14. The improved test MSE is 4.9933491. It does not improve the test MSE.

(d)

```r
library(randomForest)
set.seed(1)
bag.carseats = randomForest(Sales~., data = Carseats.train, mtry = 10, importance = T)
plot(bag.carseats)
```



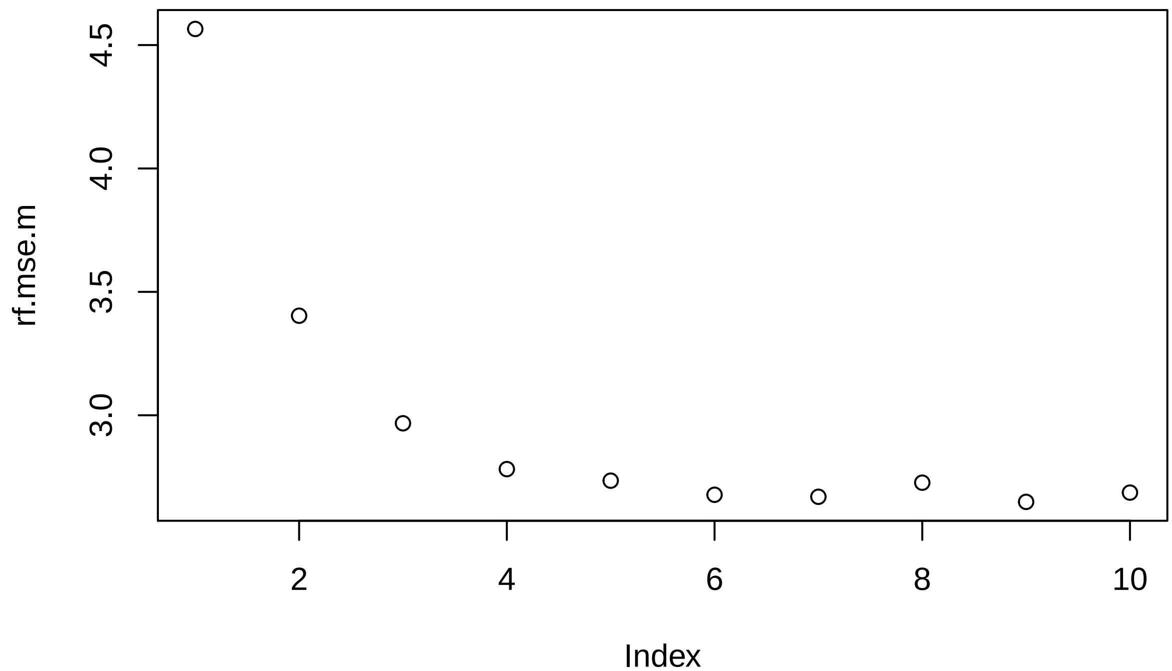

```r
bag.pred = predict(bag.carseats, newdata = Carseats.test)
bag.mse = sum((Carseats.test$Sales - bag.pred)^2) / length(bag.pred)

# importance function
importance(bag.carseats)
```

```
##                %IncMSE IncNodePurity
## CompPrice   32.5702673     258.76798
## Income      13.6077460     134.94835
## Advertising 21.3896789     157.42386
## Population   0.4843055      74.42946
## Price       74.3886644     726.20831
## ShelveLoc   80.4749889     689.35321
## Age         22.1441481     192.99213
## Education    0.3489822      59.26904
## Urban       -0.8450374      13.60746
## US           4.3414150      10.37429
```

The test MSE is 2.6549389. 'Price' and 'ShelveLoc' are the most important variables.

(e)

```
set.seed(1)
rf.mse.m = numeric(10)
for (m in 1:10) {
  rf.carseats = randomForest(Sales~., data = Carseats.train, mtry = m, importance = T)
  rf.pred = predict(rf.carseats, newdata = Carseats.test)
  rf.mse.m[m] = mean((Carseats.test$Sales - rf.pred)^2)
}
plot(rf.mse.m)
```




```
set.seed(1)
rf.carseats = randomForest(Sales~., data = Carseats.train, mtry = 3, importance = T)
rf.pred = predict(rf.carseats, newdata = Carseats.test)
rf.mse = mean((Carseats.test$Sales - rf.pred)^2)
importance(rf.carseats)
```

```
##                %IncMSE IncNodePurity
## CompPrice   16.9845091     216.84511
## Income       6.6130685     190.07964
## Advertising 13.2361438     178.78415
## Population  -0.4352699     145.68840
```

```
## Price        46.3745161     572.67348
## ShelveLoc    49.6443122     551.07641
## Age          14.7573564     244.78159
## Education     3.7088092      95.19782
## Urban         1.9709805      20.06748
## US            4.6522259      25.76327
```

The test MSE by the random forest is 2.9696593. 'Price' and 'ShelveLoc' are the most important variables. From the graph we can see as m increases, the test MSE will decrease dramatically when reaching 3. But as m continues increasing, the rate of change decreases. We can also see a fluctuation of MSE when m is between 6 and 10.