Stats 101C HW 5

Charles Liu (304804942)

11/17/2020

Loading Necessary Packages

```
library(ISLR)
library(tree)
library(randomForest)
```

Problem 1 (Exercise 8.4.2)

It is mentioned in Section 8.2.3 that boosting using depth-one trees (or stumps) leads to an additive model: that is, a model of the form

$$f(X) = \sum_{j=1}^{p} f_j(X_j)$$

Explain why this is the case. You can begin with (8.12 shown below) in Algorithm 8.2.

$$\hat{f}(X) = \sum_{b=1}^{B} \lambda \hat{f}^b(x)$$

ANSWER: Let's say we start off with $d = 1$ for in algorithm 8.12. Then, we know that for every term will be based off a single predictor. When we sum up all these terms, we find that it'll become an additive model. The reason comes from that when we boost multiple trees, we'll be fitting the residuals from the previous model on each iteration. Then, the models are added together, and the parameter d determines the number of splits. With this, we find the terminal nodes to be $d + 1$.

Problem 2 (Exercise 8.4.5)

Suppose we produce ten bootstrapped samples from a data set containing red and green classes. We then apply a classification tree to each bootstrapped sample and, for a specific value of X, produce 10 estimates of P(Class is Red|X):

0.1, 0.15, 0.2, 0.2, 0.55, 0.6, 0.6, 0.65, 0.7, and 0.75.

There are two common ways to combine these results together into a single class prediction. One is the majority vote approach discussed in this chapter. The second approach is to classify based on the average probability. In this example, what is the final classification under each of these two approaches?

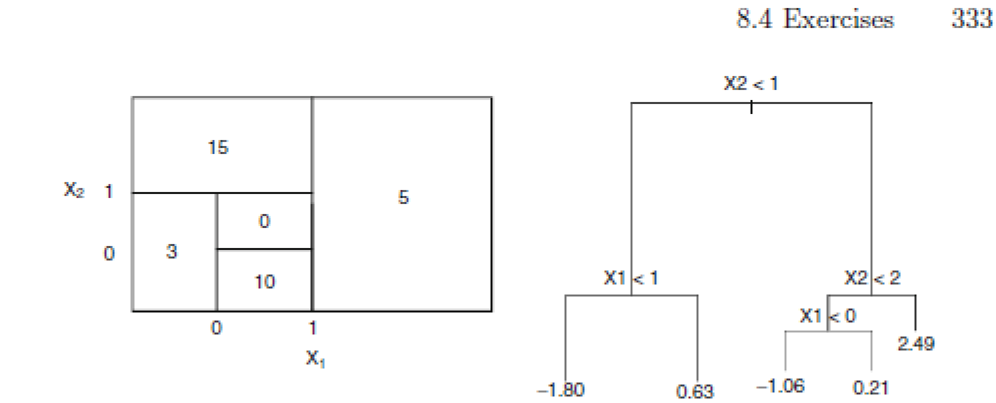


FIGURE 8.12. Left: *A partition of the predictor space corresponding to Exercise 4a.* Right: *A tree corresponding to Exercise 4b.*

Figure 1: 8.4.5 Figure

```
bootstrapped_samples <- c(0.1, 0.15, 0.2, 0.2, 0.55,
                          0.6, 0.6, 0.65, 0.7, 0.75)

method_1 <- max(ifelse(bootstrapped_samples <= 0.5, "Green", "Red")) # take majority vote

method_2 <- ifelse(mean(bootstrapped_samples) <= 0.5, "Green", "Red") # take mean probability


Q5 <- rbind(method_1, method_2)
colnames(Q5) <- "Color Results"
rownames(Q5) <- c("Majority Vote", "Average Probability")
Q5
```

```
##                     Color Results
## Majority Vote       "Red"
## Average Probability "Green"
```

ANSWER: We can see with the first approach, we have *"Red"* as the Majority Vote because we can clearly see 6 out of the 10 will end up *"Red"*. As for the Average Probability approach, we find the final results to be *"Green"*.

Problem 3 (Exercise 8.4.8)

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.

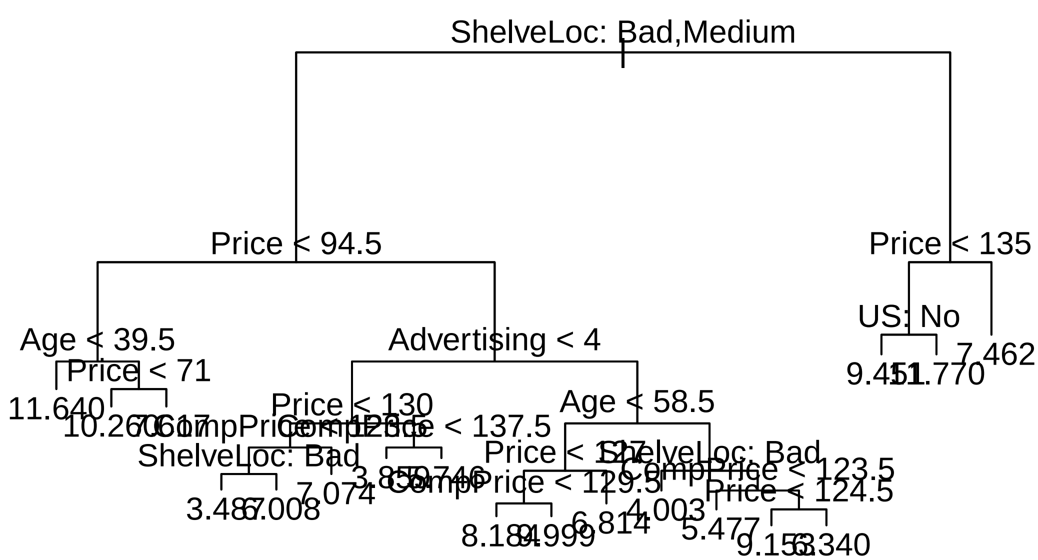
(a) Split the data set into a training set and a test set.

```r
# Load data and set.seed(...) for sampling
data(Carseats)
set.seed(1)

# 50% of data for train and 50% of data for test
train_size <- floor(0.5 * nrow(Carseats))
train_ind <- sample(seq_len(nrow(Carseats)), size = train_size)
car_train <- Carseats[train_ind, ]
car_test <- Carseats[-train_ind, ]
```

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

```r
car_tree <- tree(Sales ~ ., data = car_train)
plot(car_tree)
text(car_tree, pretty = 0)
```




```r
summary(car_tree)
```

```
##
## Regression tree:
## tree(formula = Sales ~ ., data = car_train)
## Variables actually used in tree construction:
## [1] "ShelveLoc"   "Price"       "Age"         "Advertising" "CompPrice"
```

```
## [6] "US"
## Number of terminal nodes:  18
## Residual mean deviance:  2.167 = 394.3 / 182
## Distribution of residuals:
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -3.88200 -0.88200 -0.08712  0.00000  0.89590  4.09900
```

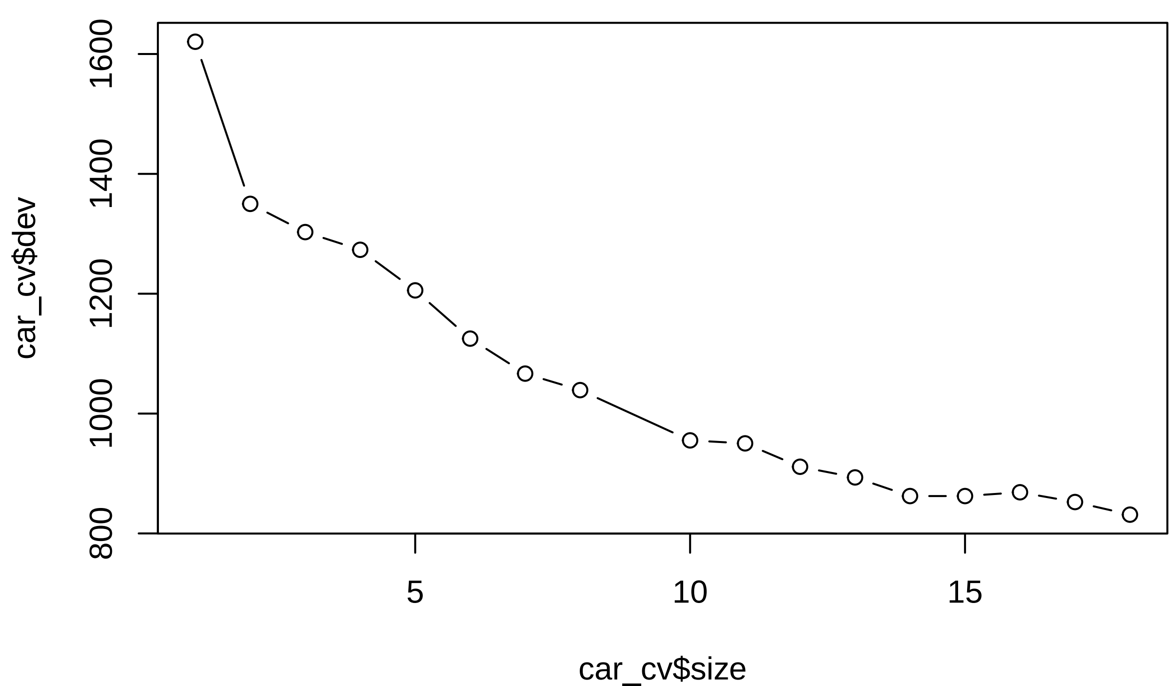
```r
car_pred <- predict(car_tree, car_test)
MSE_tree <- mean((car_test$Sales - car_pred)^2)
MSE_tree
```

```
## [1] 4.922039
```

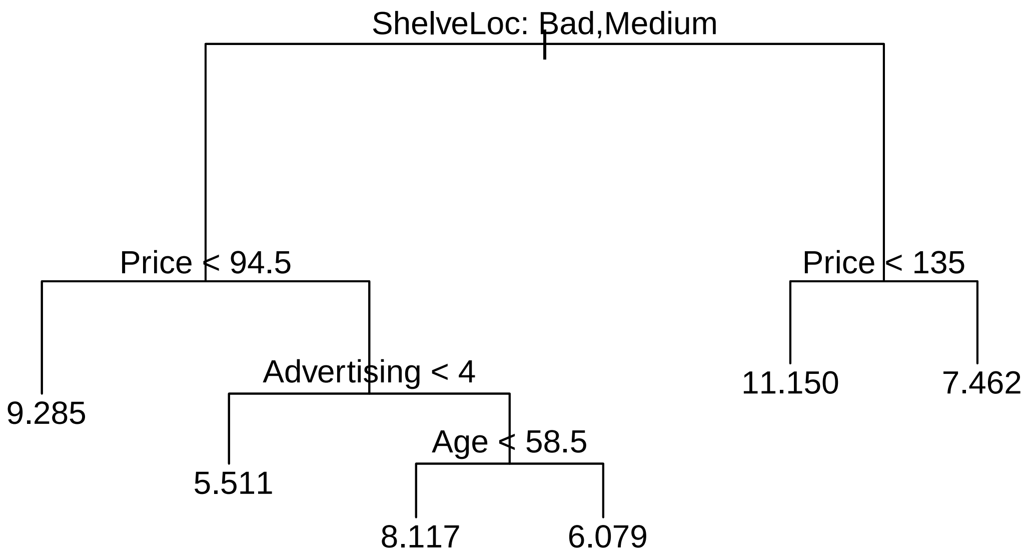
COMMENTS: We can see that we have a total of 18 terminal nodes using 6 variables. The Residual Mean Deviance is a measure of the error remaining in the tree after construction and is related to the MSE. FOr the MSE results, we have 4.922039.

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

```r
car_cv <- cv.tree(car_tree)
plot(car_cv$size ,car_cv$dev ,type="b") # it would seem best would be about 6
```



```r
car_pruned <- prune.tree(car_tree, best = 6)
plot(car_pruned)
text(car_pruned, pretty = 0)
```




```r
car_pred_pruned <- predict(car_pruned, car_test)
MSE_tree_pruned <- mean((car_test$Sales - car_pred_pruned)^2)
MSE_tree_pruned
```

```
## [1] 5.318073
```

ANSWER: No, pruning does *NOT* improve the MSE for this case. The MSE with pruned is 5.318073.

(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.

```r
p <- dim(Carseats)[2]-1
car_bag <- randomForest(Sales ~ .,
                        data = Carseats,
                        subset = train_ind,
                        mtry = p,
                        importance = TRUE)
car_bag
```

```
##
## Call:
##  randomForest(formula = Sales ~ ., data = Carseats, mtry = p,      importance = TRUE, subset = train_
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 10
##
##           Mean of squared residuals: 2.931324
##                     % Var explained: 62.72
```

```r
importance(car_bag)
```

```
##                  %IncMSE IncNodePurity
## CompPrice   23.07909904    171.185734
## Income       2.82081527     94.079825
## Advertising 11.43295625     99.098941
## Population  -3.92119532     59.818905
## Price       54.24314632    505.887016
## ShelveLoc   46.26912996    361.962753
## Age         14.24992212    159.740422
## Education   -0.07662320     46.738585
## Urban        0.08530119      8.453749
## US           4.34349223     15.157608
```

```r
car_pred_bag <- predict(car_bag, car_test)
MSE_tree_bag <- mean((car_test$Sales - car_pred_bag)^2)
MSE_tree_bag
```

```
## [1] 2.657296
```

ANSWER: The MSE obtained from bagging is 2.610255. We can see that CompPrice, ShelveLoc, and Price variables are the most important variables.

(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. Describe the effect of m, the number of variables considered at each split, on the error rate obtained.

```r
m <- 9
set.seed(1)
car_importance_1 <- randomForest(Sales ~ .,
                        data = Carseats,
                        subset = train_ind,
                        mtry = m-1,
                        importance = TRUE)

car_importance <- randomForest(Sales ~ .,
                        data = Carseats,
                        subset = train_ind,
                        mtry = m,
                        importance = TRUE)

car_pred_importance_1 <- predict(car_importance_1, car_test)
```

```r
MSE_tree_importance_1 <- mean((car_test$Sales - car_pred_importance_1)^2)
MSE_tree_importance_1
```

```
## [1] 2.647116
```

```r
car_pred_importance <- predict(car_importance, car_test)
MSE_tree_importance <- mean((car_test$Sales - car_pred_importance)^2)
MSE_tree_importance
```

```
## [1] 2.625435
```

```r
car_importance
```

```
##
## Call:
##  randomForest(formula = Sales ~ ., data = Carseats, mtry = m,      importance = TRUE, subset = train
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 9
##
##           Mean of squared residuals: 2.889193
##                     % Var explained: 63.26
```

```r
importance(car_importance)
```

```
##                %IncMSE IncNodePurity
## CompPrice   25.3795157    169.734708
## Income       5.9522997     91.001058
## Advertising 13.0188532    106.862436
## Population   0.7391527     61.885054
## Price       57.2870276    499.549240
## ShelveLoc   45.4314743    359.082376
## Age         18.1705705    161.278123
## Education   -0.2624495     44.528915
## Urban       -1.0365088      8.927086
## US           5.3678873     18.490983
```

ANSWER: The MSE is 2.625435 for the new importance for $m = 9$. We know that $m = \sqrt{p}$, so the number of variables we did at each split was 9. As our m increases closer to 10, we see that we'll have a lower MSE. We see that using $m = 8$ offers MSE of 2.647116, which is higher than when $m = 9$. Finally, we can conclude that Price and ShelveLoc are truly the most important variables.