# Homework 10 - Predictive Modeling in Finance and Insurance

Dennis Goldenberg

2024-04-11

## 1. Building a decision tree

### a. Determining roots and first split

The split at the root node has to put at least one data point into each child node. Since the heights have the possible values $\{1.4, 1.5, 1.6, 1.7, 1.8\}$, there are 4 possible splits on height, and since gender has the possible values $\{M, F\}$, there is 1 possible split on gender. I examine each of these and their corresponding $SSE = SSE_l + SSE_r$:

- Splitting on Height $\leq$ 1.4: In this case, the left child has one data point, data point 6; the mean response is just $\bar{y}_l = y_6 = 55$, so $SSE_l = 0$. The other side contains the other 6 points; here the mean response is $\bar{y}_r = \frac{88+82+60+73+77+80}{6} = \frac{230}{3}$. Thus, $SSE_r = \sum_{i \in R}(y_i - \bar{y}_r)^2 = 459.33$. Therefore, $SSE = 0 + 459.33 = 459.33$.
- Splitting on Height $\leq$ 1.5: In this case, the left child has 3 data points: 3, 4, and 6. Therefore, the mean response is $\bar{y}_l = \frac{60+77+55}{3} = 64$. Therefore, $SSE_l = \sum_{i \in L}(y_i - \bar{y}_l)^2 = 266$. The other side contains the other 4 data points, and the mean response is $\bar{y}_r = \frac{88+82+73+80}{4} = 80.75$. Here, $SSE_R = \sum_{i \in R}(y_i - \bar{y}_r)^2 = 114.75$. So, $SSE = 266 + 114.75 = 380.75$.
- Splitting on Height $\leq 1.6$: In this case, the left child has 4 data points: 1,3,5, and 6. The mean response is $\bar{y}_l = \frac{88+60+77+55}{4} = 70$. Consequently, $SSE_l = \sum_{i \in L}(y_i - \bar{y}_l)^2 = 698$. Then, the right child has 3 data points: 2, 5, and 7. The mean response is $\bar{y}_r = \frac{82+73+80}{3} = \frac{235}{3}$. So, $SSE_r = \sum_{i \in R}(y_i - \bar{y}_r)^2 = 44.67$. Finally, $SSE = 698 + 44.67 = 742.67$.
- Splitting on Height $\leq$ 1.7: In this case, the left child has 6 data points: 1,2,3,5,6,7. The mean response is $\bar{y}_l = \frac{88+82+60+77+55+80}{6} = \frac{221}{3}$. Then, $SSE_l = \sum_{i \in L}(y_i - \bar{y}_l)^2 = 861.33$. The right child only has one data point: point 4. So, the mean response is $\bar{y}_r = y_4 = 73$, and $SSE_r = 0$. Finally, $SSE = 861.33 + 0 = 861.33$.
- Splitting on Gender $< 0.5$: Encode $\{M : 0, F : 1\}$. Thus, all males (points 1,4,5, 7) are in the left child, and all females (points 2,3, 6) are in the right child. Thus, $\bar{y}_l = \frac{88+73+77+80}{4} = 79.5$, and $SSE_l = \sum_{i \in L}(y_i - \bar{y}_l)^2 = 121$. Similarly, $\bar{y}_r = \frac{82+60+55}{3} = 65.67$ and $SSE_r = \sum_{i \in R}(y_i - \bar{y}_r)^2 = 412.67$. Thus, $SSE = 121 + 412.67 = 533.67$.

Summarizing in a table:

| Split | Points in L | Points in R | $SSE_l$ | $SSE_r$ | $SSE$ |
|---|---|---|---|---|---|
| Height $\leq$ 1.4 | 6 | 1,2,3,4,5,7 | 0 | 459.33 | 459.33 |
| Height $\leq$ 1.5 | 3,5,6 | 1,2,4,7 | 266 | 114.75 | 380.75 |
| Height $\leq$ 1.6 | 1,3,5,6 | 2,4,7 | 698 | 44.67 | 742.67 |
| Height $\leq$ 1.7 | 1,2,3,5,6,7 | 4 | 861.33 | 0 | 861.33 |
| Gender $<$ 0.5 | 1,4,5,7 | 2,3,6 | 121 | 412.67 | 533.67 |

Therefore, the root node is split on Height $\leq$ **1.5**, the left leaf contains $3, 5, 6$, the right leaf contains $1, 2, 4, 7$, and their SSE's are 266 and 114.75 respectively, with an association $SSE$ of 380.75.

## b. Determining second split

At this juncture, it is possible that either leaf is subject to the next split, as both still have more than 2 data points. However, $SSE_l > SSE_r$; if I were to find a split that reduced $SSE_l$ by more than the totality of $SSE_r$, I can verify that $L$ would be split next, as this would reduce total $SSE$ by more than any potential split in $R$. In leaf $L$, I notice that the heights are $\{1.4, 1.5\}$ and the genders are $\{M, F\}$, so there are two possible splits (I'll refer to the left child as $L_1$ and the right as $L_2$):

- Splitting on Height $\leq 1.4$: In this case, only data point 6 is in $L_1$, so $\bar{y}_{L_1} = y_6 = 55$ and $SSE_{L_1} = 0$. Meanwhile, data points 3 and 5 are in $L_2$, so $\bar{y}_{L_2} = \frac{60+77}{2} = 68.5$ and $SSE_{L_2} = \sum_{i \in L_2}(y_i - \bar{y}_{L_2})^2 = 144.5$. So, $SSE = SSE_{L_1} + SSE_{L_2} = 144.5$.
- Splitting on Gender $< 0.5$: In this case, the males, or only data point 5, are in $L_1$; therefore, $\bar{y}_{L_1} = y_5 = 77$ and $SSE_{L_1} = 0$. Then, the females, or data points 3 and 6, are in $L_2$; therefore, $\bar{y}_{L_2} = \frac{60+55}{2} = 57.5$ and $SSE_{L_2} = \sum_{i \in L_2}(y_i - \bar{y}_{L_2})^2 = 12.5$. So, $SSE = SSE_{L_1} + SSE_{L_2} = 12.5$.

Note that, when the left leaf is split on gender, the new $SSE$ on that side of the tree is 12.5. Note that $SSE_L - SSE = 266 - 12.5 = 243.5$, a reduction in $SSE$ greater than the entirety of the $SSE$ in the right leaf. Thus, as this split maximizes reduction in $SSE$ of the data points in the left leaf, and the reduction is greater than the entirety of the $SSE$ in the right leaf, we conclude that:

- The second split is on the left leaf, and the split is Gender $< 0.5$.
- Data point 5 is in $L_1$, and data points 3 and 6 are in $L_2$.
- $SSE_{L_1} = 0$ and $SSE_{L_2} = 12.5$
- $SSE = SSE_{L_1} + SSE_{L_2} = 12.5$; $SSE_{\text{total}} = 12.5 + 144.75 = 157.25$.

## c. Complete decision tree build

Note that $L_1$ has 1 data point, and $L_2$ has 2 data points. Therefore, both have reached the minimum of data points in a leaf, and cannot be split any more. However, leaf $R$ has 4 data points (1,2,4,7); therefore, it can be split once again. The heights $\{1.6, 1.7, 1.8\}$ and the genders are $\{M, F\}$ among the 4 data points; therefore, there can be $2 + 1 = 3$ possible splits (calling the left child $R_1$ and right child $R_2$).

- Splitting on Height $\leq 1.6$: Here, data point 1 is in $R_1$, so $\bar{y}_{R_1} = y_1 = 88$ and $SSE_{R_1} = 0$. Then, data points 2,4, and 7 are in $R_2$, so $\bar{y}_{R_2} = \frac{82+73+80}{3} = \frac{235}{3} = 78.33$; thus, $SSE_{R_2} = \sum_{i \in R_2}(y_i - \bar{y}_{R_2})^2 = 44.67$. So $SSE_R = 0 + 44.67 = 44.67$.
- Splitting on Height $\leq 1.7$: Here, data points 1,2, and 7 are in $R_1$; therefore, $\bar{y}_{R_1} = \frac{88+82+80}{3} = \frac{250}{3} = 83.33$. So, $SSE_{R_1} = \sum_{i \in R_1}(y_i - \bar{y}_{R_1})^2 = 34.67$. The only data point remaining is data point 4, which is in $R_2$; so $\bar{y}_{R_2} = y_4 = 73$ and $SSE_{R_2} = 0$. Thus, $SSE_R = 34.67 + 0 = 34.67$.
- Splitting Gender $< 0.5$: Here, all the males are in $R_1$, or data points 1,4, and 7. Here, $\bar{y}_{R_1} = \frac{88+73+80}{3} = \frac{241}{3} = 80.33$. Thus, $SSE_{R_1} = \sum_{i \in R_1}(y_i - \bar{y})^2 = 112.67$. There is only one female in $R$: data point 2. As a result, $\bar{y}_{R_2} = y_2 = 82$ and $SSE_{R_2} = 0$. So, $SSE_R = 112.67 + 0 = 112.67$.

The best split here would be on Height $\leq 1.7$, as this minimizes the $SSE_R$. With this split, $SSE = SSE_L + SSE_R = 12.5 + 34.67 = \mathbf{47.17}$. However, $R_1$ still has 3 data points, so it must be split once more to reach the minimum. $R_1$ contains the points 1,2, and 7. The potential heights are $\{1.6, 1.7\}$ and genders are $\{M, F\}$, thus giving way to two final splits (I will call the left child $R_{1_a}$ and $R_{1_b}$):

- Splitting on Height $\leq 1.6$: Here, point 1 is the only point in $R_{1_a}$, so $\bar{y}_{R_{1_a}} = y_1 = 88$ and $SSE_{R_{1_a}} = 0$. Then, points 2 and 7 are in $R_{1_b}$, so $\bar{y}_{R_{1_b}} = \frac{82+80}{2} = 81$ and $SSE_{R_{1_b}} = \sum_{i \in R_{1_b}}(y_i - \bar{y}_{R_{1_b}})^2 = 2$. So, $SSE_{R_1} = 0 + 2 = 2$.
- Splitting on Gender $< 0.5$: All the males, or points 1 and 7, would be in $R_{1_a}$. Thus, $\bar{y}_{R_{1_a}} = \frac{88+80}{2} = 84$ and $SSE_{R_{1_a}} = \sum_{i \in R_{1_a}}(y_i - \bar{y}_{R_{1_a}})^2 = 32$. This would leave the only female in the subset, or data point 2, in $R_{1_b}$, so $\bar{y}_{R_{1_b}} = y_2 = 82$ and $SSE_{R_{1_b}} = 0$. Finally, $SSE_{R_1} = 0 + 32 = 32$

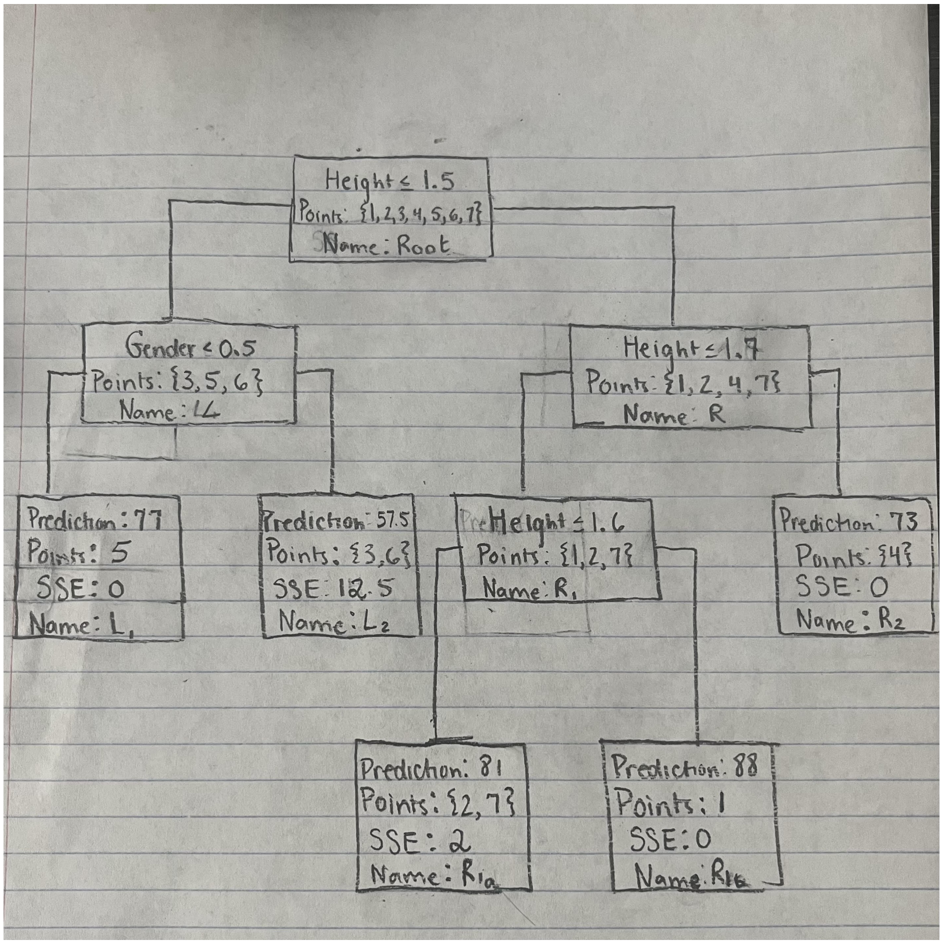
Clearly, splitting on Height $\leq 1.6$ reduces the $SSE$ the most. Therefore, this split is undertaken, and $R_{1_a}$ has 1 data point while $R_{1_b}$ has 2, so each leaf node has 2 or less data points. The tree is complete, with final $SSE = SSE_L + SSE_R = 12.5 + SSE_{R_1} + SSE_{R_2} = 12.5 + 2 + 0 = 14.5$. The leaves, the condition to be in those leaves, the prediction (sample mean), the points in the leaves, and the corresponding SSE can be found in the table below:

| Leaf Name | Condition | Prediction | Points | SSE |
|---|---|---|---|---|
| $L_1$ | Height $\leq 1.5 \wedge$ Gender $=$ M | 77 | 5 | 0 |
| $L_2$ | Height $\leq 1.5 \wedge$ Gender $=$ F | 57.5 | 3,6 | 12.5 |
| $R_{1_a}$ | Height $\in (1.5, 1.6]$ | 88 | 1 | 0 |
| $R_{1_b}$ | Height $\in (1.6, 1.7]$ | 81 | 2,7 | 2 |
| $R_2$ | Height $> 1.7$ | 73 | 4 | 0 |

## d. depict entire tree

The tree is shown below:

```r
knitr::include_graphics("tree.jpg")
```



## e. Predicting weight for male, height 1.45 m

Using this tree, and the fact that male encodes for 0 in gender, I note that a male of height 1.45 meters falls into leaf $L_1$. Thus, the predicted weight for said male would be **77** kg.

## f. What is sequence of $\alpha_T$?

To determine the list of $\alpha_T$ values that prune nodes, I have to first know the total SSE after each split. Therefore, I list the trees in order after each split, the number of leaves they have, and the SSE at the current time:

| Tree | Latest Split | SSE | Number of Leaves |
|---|---|---|---|
| 1 | N/A | 861.71 | 1 |
| 2 | Height $\leq 1.5$ | 380.75 | 2 |
| 3 | Gender $< 0.5$ | 157.25 | 3 |
| 4 | Height $\leq 1.7$ | 47.17 | 4 |
| 5 | Height $\leq 1.6$ | 14.5 | 5 |

The Tree score formula is $SSE + \alpha|T|$, where $T$ is the number of leaves. Therefore, I pick tree 4 over 5 when:

$$47.17 + \alpha_1 * 4 < 14.5 + \alpha_1 * 5 \rightarrow \alpha_1 > 47.17 - 14.5 = 32.67$$

I do a similar procedure for the other consecutive trees:

$$157.25 + \alpha_2 * 3 < 47.17 + \alpha_2 * 4 \rightarrow \alpha_2 > 157.25 - 47.17 = 110.08$$
$$380.75 + \alpha_3 * 2 < 157.25 + \alpha_3 * 3 \rightarrow \alpha_3 > 380.75 - 157.25 = 223.5$$
$$861.71 + \alpha_4 * 1 < 380.75 + \alpha_4 * 2 \rightarrow \alpha_4 > 861.71 - 380.75 = 480.96$$

Rounding to the nearest whole number above the minimum for pruning a given leaf, I generate the following $\alpha_T$ values under which each tree is optimal, (starting from the most complex tree and iteratively pruning back to the root):

$$\alpha_T = \{0, 33, 111, 224, 481\}$$

## 2. Building Decision Tree with Train-Test Split

```
library(MASS)
library(randomForest)
library(rpart)
library(rpart.plot)
library(tree)
library(gbm)
```

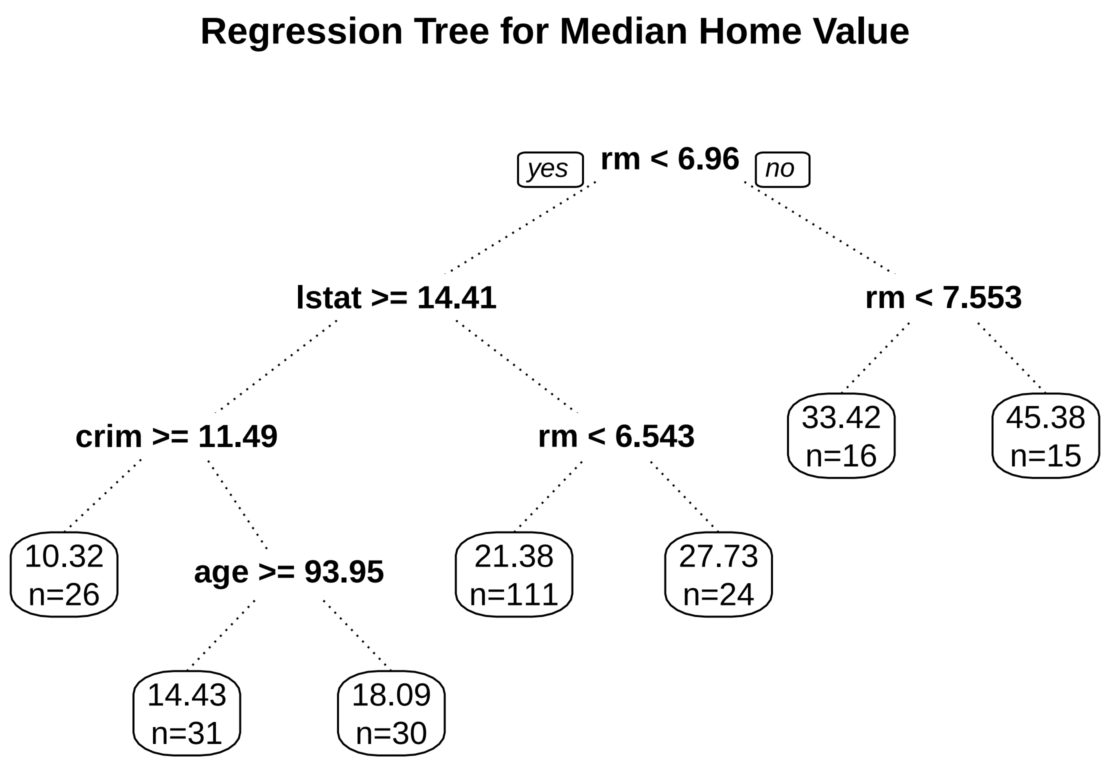
### a. Splitting dataset

```
set.seed(1)
train_sample <- sort(sample(seq_len(nrow(Boston)),
                     size = floor(0.5 * nrow(Boston))))
trainData <- Boston[train_sample,]
testData <- Boston[-train_sample,]
```

### b. Building decision tree

#### i. Fitting Regression Tree

```
rtree <- rpart(medv ~ ., data = trainData, method = "anova")
prp(rtree, main = "Regression Tree for Median Home Value", roundint = FALSE,
    extra = 1, digits = 4, branch.lty = 3)
```



Note that the number of rooms is split on first, as well as being the only variable split on multiple times. So, *rm* is the most important variable. The other variables split on (ordered by depth in tree at which they are split on) are *lstat*, *crim*, and *age*. So the order of importance for variables is $1.rm, 2.lstat, 3.crim, 4.age$.

### ii. Calculating test MSE

I predict using the fitted tree, and calculate test MSE:

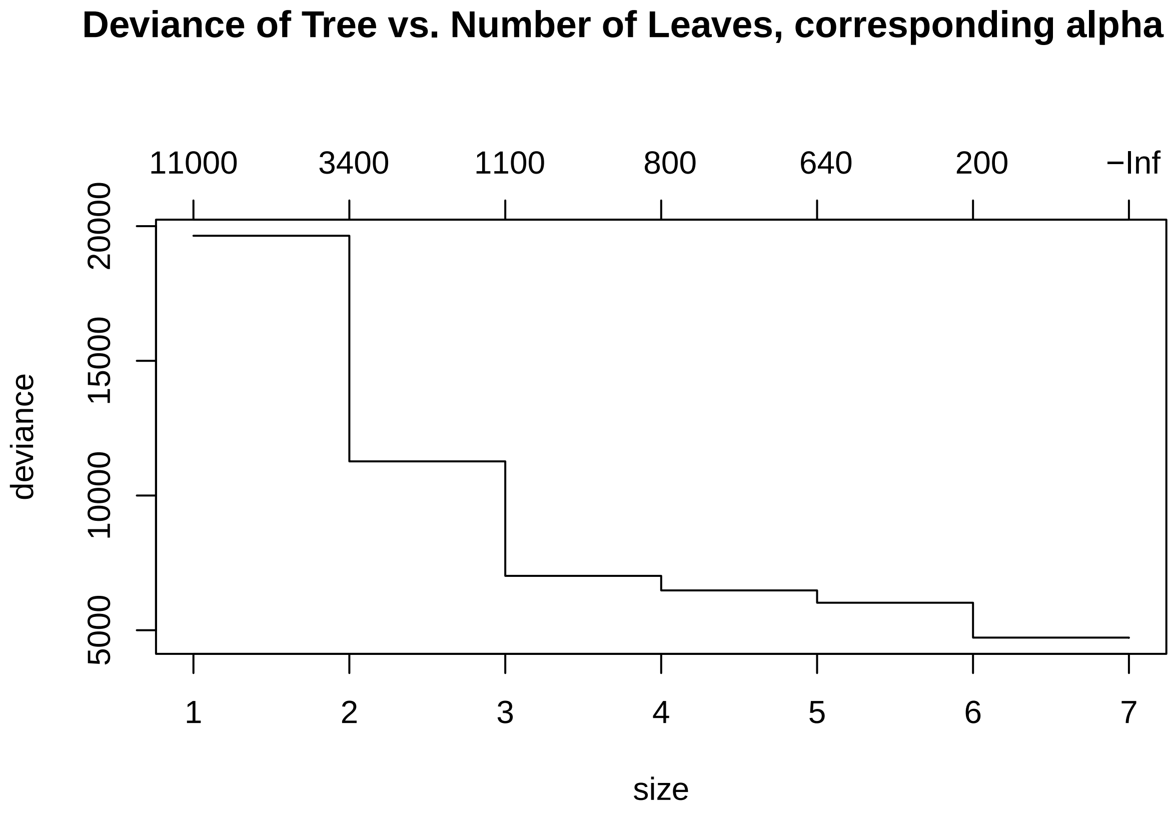
```
predictions <- predict(rtree, newdata = testData)
SSE <- sum((predictions - testData$medv)^2)
MSE = SSE/(length(predictions))
sprintf("Test MSE: %.5f", MSE)
```

```
## [1] "Test MSE: 35.28688"
```

### iii. Cross-Validation for optimal Tree

I do 10-fold cross validation and plot the deviance (note that I use the tree method instead of rpart, as rpart plots better):

```
set.seed(1)
rtree <- tree(medv ~ ., data = trainData)
cv <- cv.tree(rtree, K = 10, FUN = prune.tree)
par(oma = c(0,0,2,0))
plot(cv)
title(main = "Deviance of Tree vs. Number of Leaves, corresponding alpha",
      outer = TRUE)
```



The deviance is very close between the tree with 6 leaves and 7 leaves. The seven leaf tree is the one already created; therefore, I examine the test error of the 6 leaf tree:

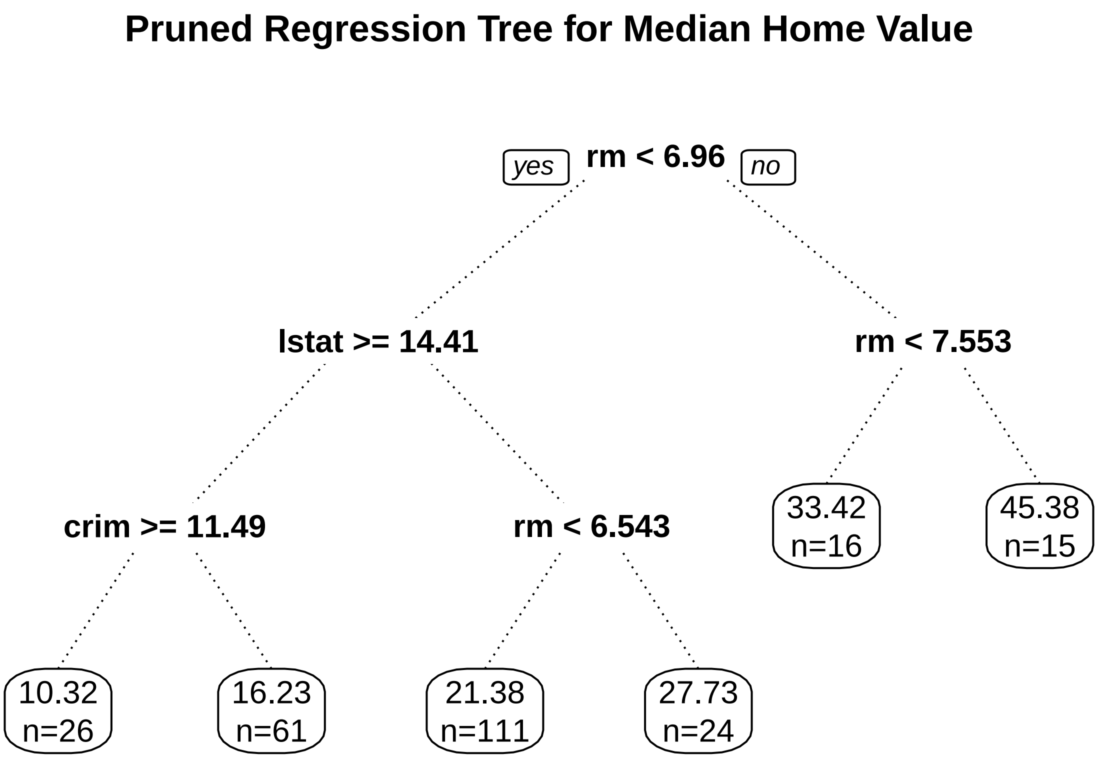
```
rtree6 <- prune.tree(rtree, best = 6)
pred6 <- predict(rtree6, newdata = testData)
SSE6 <- sum((pred6 - testData$medv)^2)
MSE6 = SSE6/(length(pred6))
sprintf("Test MSE of 6 leaf tree: %.5f", MSE6)
```

```
## [1] "Test MSE of 6 leaf tree: 35.16439"
```

Note that $\text{MSE}_6 = 35.16439 < 35.28688 = \text{MSE}_7$. Thus, the the tree with **6 leaves** is the best.

#### iv. Pruning the tree

```
replot <- rpart(medv ~ ., data = trainData, method = "anova")
replot <- prune(replot, cp = 0.02)
prp(replot, main = "Pruned Regression Tree for Median Home Value",
    roundint = FALSE, extra = 1, digits = 4, branch.lty = 3)
```



#### v. Test error

The test error is was calculated as: $\text{MSE}_6 = 35.16439$.

### c. Building Bagging Model

I create a bagging model, and calculate the test MSE:

```
set.seed(1)
rB <- randomForest(medv ~., ntree = 500, mtry = dim(Boston)[2] - 1,
                   data = trainData, importance = TRUE)
predictionsrB <- predict(rB, newdata = testData)
MSErB <- mean((predictionsrB - testData$medv)^2)
sprintf("Test MSE of Bagging Model: %.5f", MSErB)
```

```
## [1] "Test MSE of Bagging Model: 23.27780"
```

I also return the top three variables in importance as measured by the mean decrease in MSE:

```
sortedImportance <- sort(rB$importance[,2], decreasing = TRUE)
sortedImportance[1:3]
```

```
##         rm      lstat       crim
## 11998.3365  4967.9514   814.6526
```

So, this suggests that *rm*, *lstat*, and *crim* are the 3 most important variables in decreasing MSE.

## d. Building Random Forest

I create a random Forest, and calculate the test MSE:

```
set.seed(1)
rF <- randomForest(medv ~., ntree = 500, data = trainData, importance = TRUE)
predictionsrF <- predict(rF, newdata = testData)
MSErF <- mean((predictionsrF - testData$medv)^2)
sprintf("Test MSE of random Forest Model: %.5f", MSErF)
```

```
## [1] "Test MSE of random Forest Model: 18.73938"
```

## e. Building boosted tree

### i. Building Model, Calculating testMSE

I build the tree, and then calculate:

```
set.seed(1)
gbMod <- gbm(medv ~ ., data = trainData, distribution = "gaussian",
             n.trees = 1000, interaction.depth = 4)
predictionsgB <- predict(gbMod, newdata = testData)
MSEgB <- mean((predictionsgB - testData$medv)^2)
sprintf("Test MSE of gradient Boosted Model: %.5f", MSEgB)
```

```
## [1] "Test MSE of gradient Boosted Model: 18.47792"
```

### ii. Determining optimal interaction depth d

I generate each of the gradient boosted models, and find the interaction depth parameter that yields the minimum MSE:

```
set.seed(1)
mseVec <- c()
for(i in 2:10){
  gbTest <- gbm(medv ~ ., data = trainData, distribution = "gaussian",
             n.trees = 1000, interaction.depth = i)
  predgBTest <- predict(gbTest, newdata = testData)
  MSEgBTest <- mean(((predgBTest - testData$medv)^2))
  mseVec <- append(mseVec, MSEgBTest)
}
```

```
names(mseVec) <- 2:10
optD <- as.numeric(names(which.min(mseVec)))
optDMSE <- mseVec[as.character(optD)]
sprintf("Optimal depth: %.0f, Test MSE: %.5f", optD, optDMSE)
```

```
## [1] "Optimal depth: 5, Test MSE: 17.60284"
```

### iii. Determining optimal shrinkage parameter

I iterate over different possible shrinkage parameters, fitting the model with each and the optimal interaction depth parameter from above to find the minimum MSE:

```
set.seed(1)
mseVecL <- c()
for(l in seq(0.001, 0.2, by = 0.001)){
 gbTestL <- gbm(medv ~ ., data = trainData, distribution = "gaussian",
             n.trees = 1000, shrinkage = l, interaction.depth = optD)
 predgBTestL <- predict(gbTestL, newdata = testData)
 MSEgBTestL <- mean((predgBTestL - testData$medv)^2)
 mseVecL <- append(mseVecL, MSEgBTestL)
}
names(mseVecL) <- seq(0.001, 0.2, by = 0.001)
optL <- as.numeric(names(which.min(mseVecL)))
optLMSE <- mseVecL[as.character(optL)]
sprintf("Optimal shrinkage: %.3f, Test MSE: %.5f", optL, optLMSE)
```

```
## [1] "Optimal shrinkage: 0.164, Test MSE: 16.60171"
```

### iv. Comparing test MSEs

I summarize the results of the above 3 procedures:

```
mseSum <- as.data.frame(cbind(c("Base Boosted Model", "Boosted + Optimal depth",
                                "Boosted + Optimal depth and shrinkage"),
                              unname(c(MSEgB, optDMSE, optLMSE))))
colnames(mseSum) <- c("Model Specifications", "Test MSE")
mseSum
```

```
##                    Model Specifications         Test MSE
## 1                    Base Boosted Model 18.4779230217593
## 2               Boosted + Optimal depth 17.6028379650112
## 3 Boosted + Optimal depth and shrinkage 16.6017066320589
```

With each optimization of parameters, the test MSE was reduced, improving the overall accuracy of the model.

# 3. Gradient Boosted Tree for Q1 data

## a. Building the First tree

The first tree is just one leaf, with no splits. Therefore, the prediction at that tree is simply the average of all weights, or $\hat{y} = \bar{y} = \frac{88+82+60+73+77+55+80}{7} = 73.57$. I calculate the residuals by generating a data frame of table 1 data, and then subtracting $\bar{y}$:

```r
Height <- c(1.6,1.7,1.5,1.8,1.5,1.4,1.7)
Gender <- c(0,1,1,0,0,1,0)
Weight <- c(88,82,60,73,77,55,80)
table1 <- as.data.frame(cbind(Height, Gender, Weight))
table1$Gender <- factor(table1$Gender)
residual <- table1$Weight - mean(table1$Weight)
resid <- as.data.frame(cbind(1:7, Height, Gender, residual))
colnames(resid) <- c("Observation", "Height", "Gender","Residual")
resid
```

```
##   Observation Height Gender    Residual
## 1           1    1.6      0  14.4285714
## 2           2    1.7      1   8.4285714
## 3           3    1.5      1 -13.5714286
## 4           4    1.8      0  -0.5714286
## 5           5    1.5      0   3.4285714
## 6           6    1.4      1 -18.5714286
## 7           7    1.7      0   6.4285714
```

## b. Creating the 2nd tree

## c. Creating the 3rd tree

## d. Drawing whole gradient boosted tree

## e. Predicting weight for male with height 1.45m