Homework 5

James Young

Resources: Stack overflow, Github

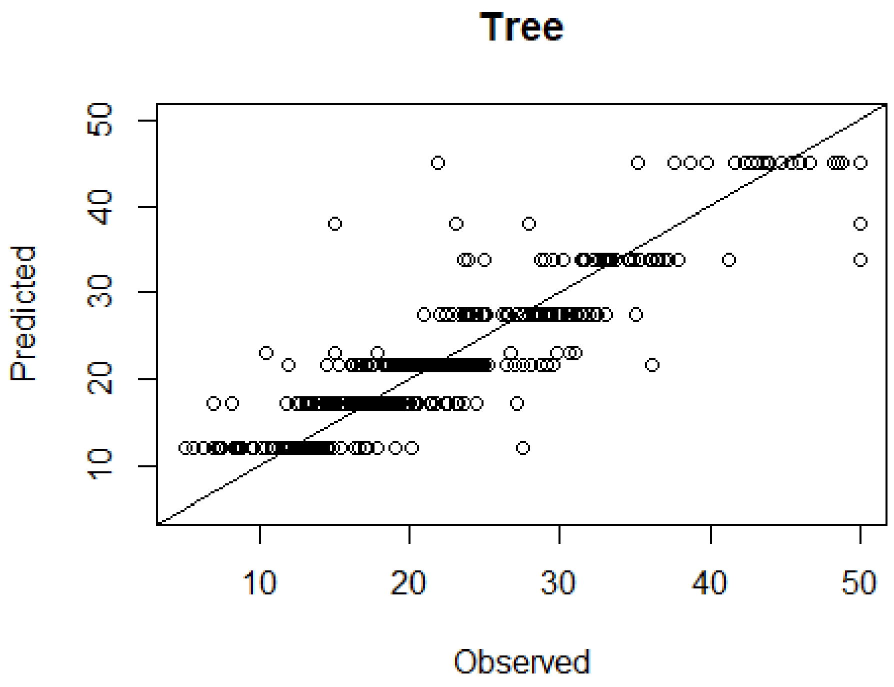
*Here we load in the required packages and the “BostonHousing” data that will be used.

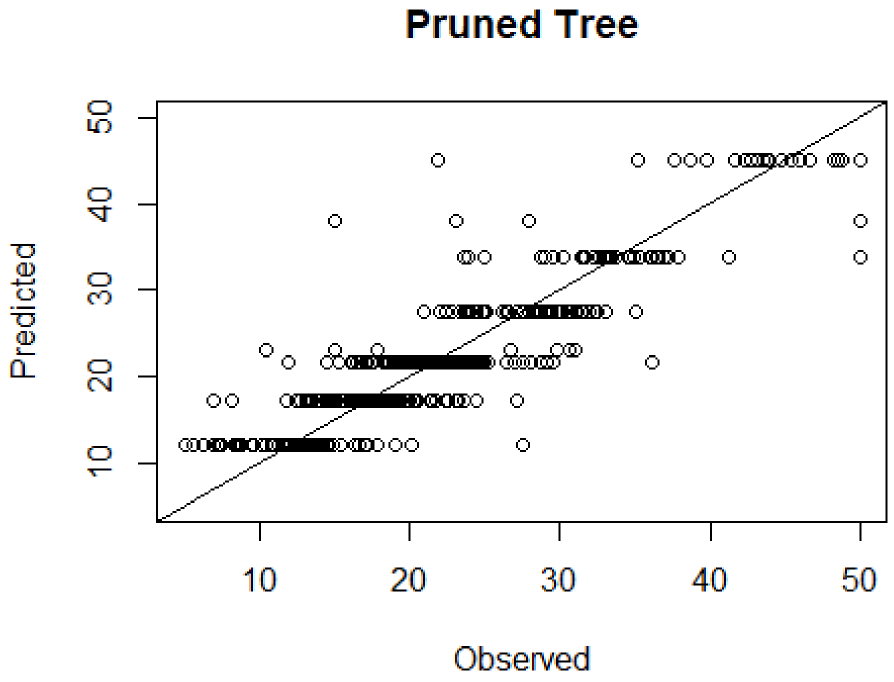
```
library(mlbench)
library(rpart)
library(partykit)
library(TH.data)
library(ada)
library(adabag)
library(rattle.data)
library(ggplot2)
data(BostonHousing)
names(BostonHousing)

##  [1] "crim"    "zn"      "indus"   "chas"    "nox"     "rm"      "age"
##  [8] "dis"     "rad"     "tax"     "ptratio" "b"       "lstat"   "medv"

attach(BostonHousing)
```

1. The dataset reported by Harrison and Rubinfeld (1978) is available as data.frame package (Leisch and Dimitriadou, 2009). The goal here is to predict the median value of owner-occupied homes (medv variable, in 1000s USD) based on other predictors in the dataset. Use this dataset to do the following

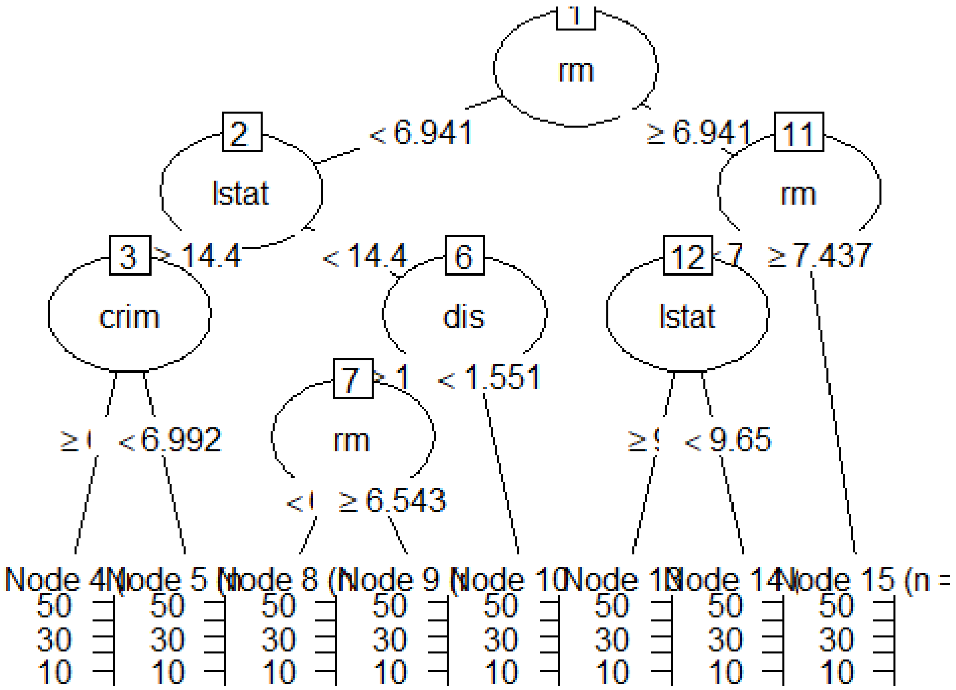
   a.) Construct a regression tree using rpart(). The following need to be included in your discussion. How many nodes did your tree have? Did you prune the tree? Did it decrease the number of nodes? What is the prediction error (calculate MSE)? Provide a plot of the predicted vs. observed values. Plot the final tree.

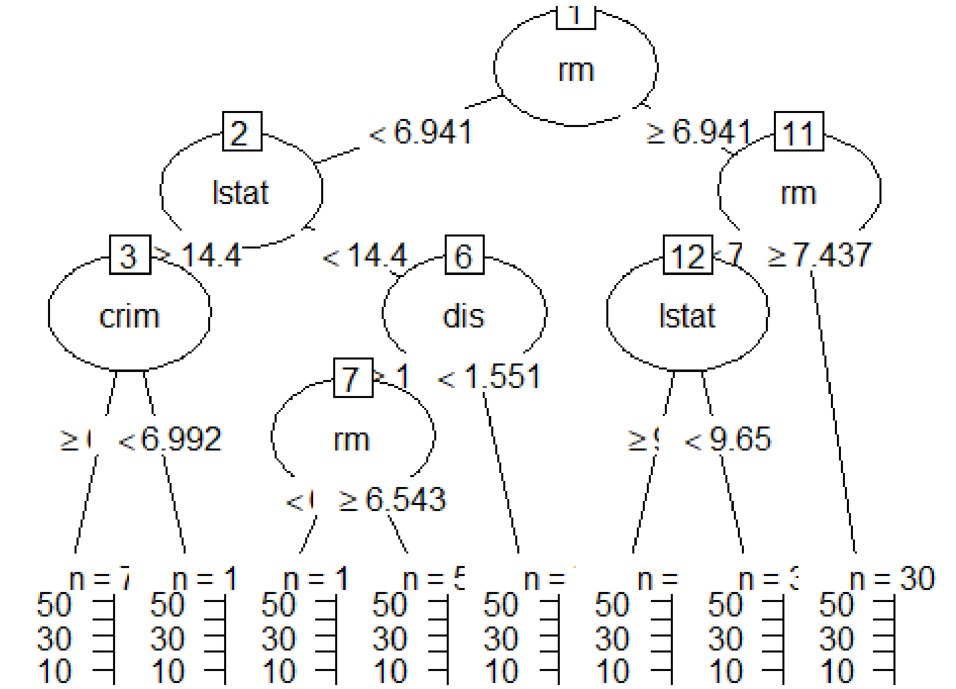
```
## Tree:

## [1] 16.24467

##
## Tree with Pruning:

## [1] 16.24467
```





Here we see that the error rate for trees, both pruned and unpruned, are the same and the predicted vs. observed plots do not change. Lets check out the trees below to check for a node difference.

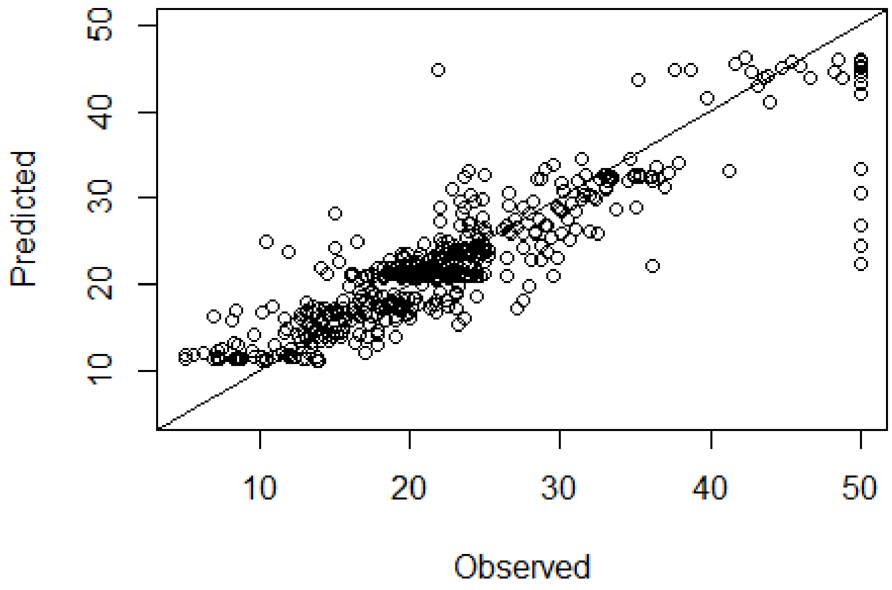




Both pruned and unpruned trees have the same number of nodes, which is seven nodes.

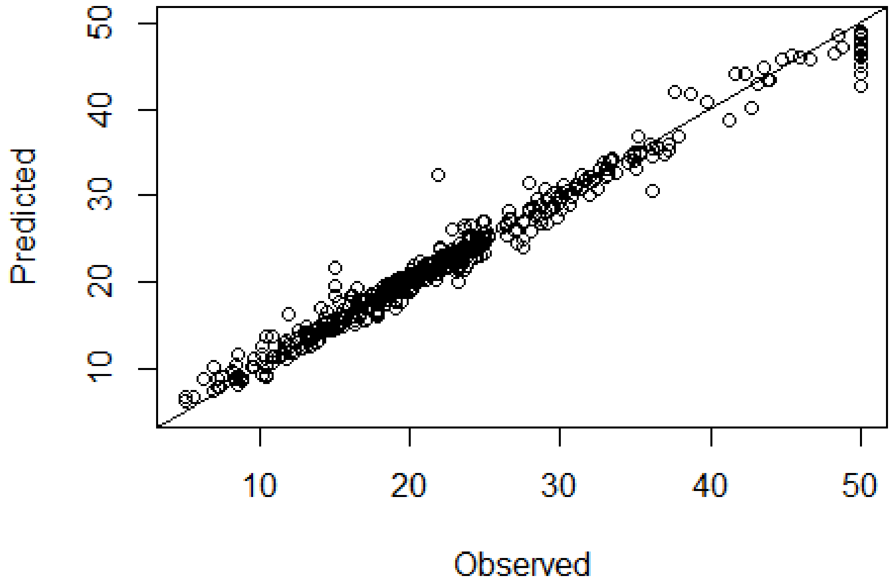
```
b) Perform bagging with 50 trees. Report the prediction error (MSE). Provide
the predicted vs observed plot.
```

```
##
## lstat    rm
##    25    25

## [1] 17.6975
```



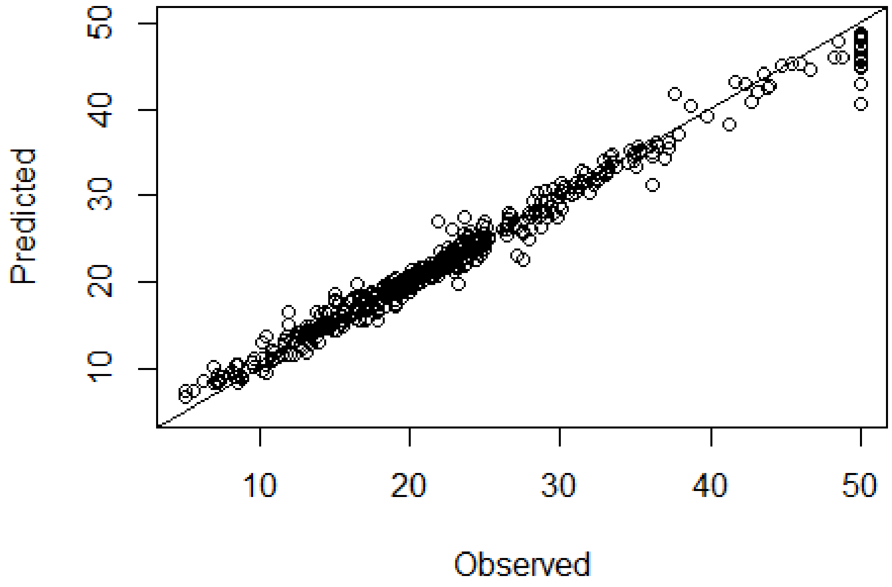
Here we see MSE actually rises compared to the singular trees we previously made.

```
c) Use randomForest() function in R to perform bagging. Report the prediction
error (MSE). Was it the same as (b)? If they are different what do you think
caused it?  Provide a plot of the predicted vs. observed values.

## [1] 1.957831
```



The random forest bagging model is much better in terms of MSE versus the rpart trees (bagged or unbagged). I think this may because of the difference in the way random forest and rpart work. Each random forest tree is itself an ensemble of "weak-learners" that together have strong predictive ability where each tree of rpart is just a tree.

```
d) Use randomForest() function in R to perform random forest. Report the
prediction error (MSE).  Provide a plot of the predicted vs. observed values.

## [1] 2.046214
```



Here we see pretty similar performance to the random forest bagged tree in terms of MSE. We need to keep in mind that this is the MSE for data that the model has already seen and does not imply that it would strongly generalize.

```
e) Provide a table containing each method and associated MSE. Which method is
more accurate?

##                     Method Mean_Square_Error
## 1 Single Tree (un-pruned)         16.244674
## 2    Single Tree (pruned)         16.244674
## 3        50 Tree Ensemble         17.697496
## 4   Random Forest Bagging          1.957831
## 5           Random Forest          2.046214
```

In my models, with random.seed set to 1, the most accurate model was the random forest bagging model which had the lowest MSE. In fact, random forest bagging was about 8-fold better than all rpart trees (bagged or unbagged) and slightly better than a "out of the box" random forest. However, it may be possible that this random forest model is overfitting the data it was given and may not generalize well to other data, say from another city.

2. Consider the glaucoma data (data = "", package = "").

   a) Build a logistic regression model. Note that most of the predictor variables are highly correlated. Hence, a logistic regression model using the whole set of variables will not work here as it is sensitive to correlation.

The solution is to select variables that seem to be important for predicting the response and using those in the modeling process using GLM. One way to do this is by looking at the relationship between the response variable and predictor variables using graphical or numerical summaries - this tends to be a tedious process. Secondly, we can use a formal variable selection approach. The *step*() function will do this in R. Using the *step* function, choose any direction for variable selection and fit logistic regression model. Discuss the model and error rate.

Do not print out the summaries of every single model built using variable selection. That will end up being dozens of pages long and not worth reading through. Your discussion needs to include the direction you chose. You may only report on the final model, the summary of that model, and the error rate associated with that model.

```
## 
## Call:
## glm(formula = Class ~ phct + phci + hvc + mr + rnf + emd, family = 
"binomial", 
##     data = glau1)
## 
## Deviance Residuals: 
##      Min        1Q    Median        3Q       Max  
## -2.13784  -0.63261  -0.00377   0.60628   2.50871  
## 
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)    
## (Intercept)    6.968      2.289   3.044 0.002337 ** 
## phct           4.306      2.989   1.440 0.149759    
## phci           9.242      2.643   3.497 0.000470 ***
## hvc           -5.405      1.981  -2.729 0.006350 ** 
## mr            -6.423      2.479  -2.591 0.009569 ** 
## rnf          -12.973      3.546  -3.659 0.000253 ***
## emd            9.688      2.525   3.836 0.000125 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 271.71  on 195  degrees of freedom
## Residual deviance: 154.13  on 189  degrees of freedom
## AIC: 168.13
## 
## Number of Fisher Scoring iterations: 6

## 
## Error Rate from step glm:

## [1] 0.4846939
```

I tried going in forward, reverse, and both directions and settled on the variables used in the final model. However, I also tried filtering out variables based on covariance first and got a better (lower) error rate. The final model has picked 6 variables out of the starting pool which are all significant at p <0.05 or close to it. This is a binary prediction and my model shows an error rate of 0.49.

```
b) Build a logistic regression model with K-fold cross validation (k = 10).
Report the error rate.

## K-fold cross validation (K=10) error rate:

## [1] 0.7295918
```

We see the 10-fold cross validation is pretty high and should not be considered for clinical use in its current state.

```
c) Find a function (package in R) that can conduct the "adaboost" ensemble
modeling. Use it to predict glaucoma and report error rate. Be sure to
mention the package you used.

## Error Rate from whole data boosting:

## [1] 0

##
## Error Rate from train boosting:

## [1] -0.0212766

d) Report the error rates based on single tree, bagging and random forest. (A
table would be great for this).

##                           Method Error.Rate
## 1                       Step GLM  0.4846939
## 2                      K-Fold CV  0.7295918
## 3 Adaptive Boosting: Whole Data -0.0212766
## 4              Adaptive Boosting  0.0000000

e) Write a conclusion comparing the above results (use a table to report
models and corresponding error rates). Which one is the best model?
```

The table shows that the adaboost technique worked the best, even when we “hid” some of the data with a train test split. I judge this based on lowest error rate. This may be because of how adaboost works, giving different trees different weights within the ensemble based on their performance. The glm model, the simplest of the models, had the worst performance.

```
f) From the above analysis, which variables seem to be important in
predicting Glaucoma?

##       emd       hvc        mr        mv      phci      phct       rnf
## 15.069167 12.725215 13.325029  9.995113 18.907193 14.225724 15.752559
```

The above analysis shows that phci, emd, hvc, mr, mv, phct, and rnf all have importance numbers and we see rnf was the most important while mv was the least important and we also see some difference in ranking when comparing to the glm (if we consider p-value a pseudo-rank).