Lab 13 MATH 4322

Bagging, Random Forest and Boosting

Fall 2023

- We will apply bagging, random forests and boosting to the `Boston` data, using the `randomForest` package.
- *Note*: The exact results obtained in this lab may depend on the version of `R` and the version of the `randomForest` package installed on your computer. Give the results from your computer.
- You can use the `Rmarkdown` script given or write down your answers and scan them as a pdf file to upload in Canvas similar to your homework.
- Possible points: 10.

Question 1: For any data that has p predictors **bagging** requires that we consider how many predictors at each split in a tree?

There are p predictors.

First, we call the data and create training/testing sets.

```
library(ISLR2)
set.seed(1)
train = sample(1:nrow(Boston),nrow(Boston)/2)
boston.test = Boston[-train,"medv"]
```

Bagging

We perform bagging as follows:

```
library(randomForest)
set.seed(10)
bag.boston = randomForest(medv~., data = Boston,
                          subset = train,
                          mtry = ncol(Boston) - 1,
                          importance = TRUE)
bag.boston
```

Question 2: What is the *MSE* based on the training set?

The MSE is equal to 11.69993.

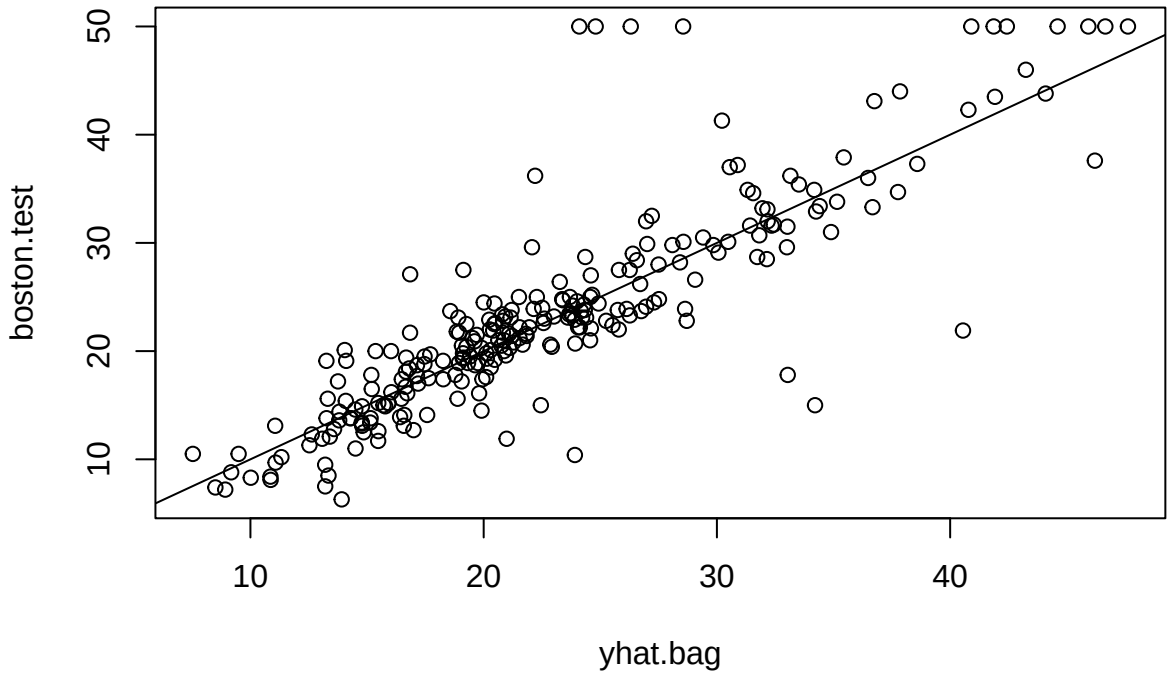
How well does this bagged model perform on the test set?

Question 3: What is the formula to determine the *MSE*?

MSE = (1/n) * sum((yi-$^{yi)}$2)

Run the following in R.

```
yhat.bag = predict(bag.boston,newdata = Boston[-train,])
plot(yhat.bag,boston.test)
abline(0,1)
```




```
mean((yhat.bag - boston.test)^2)
```

Question 4: What is the *MSE* of the test data set?

The MSE of the test data set is 23.23877.

We could change the number of trees grown by `randomForest()` using the `ntree` argument:

```
bag.boston = randomForest(medv ~ ., data = Boston,
                          subset = train,
                          mtry = ncol(Boston) - 1,
                          ntree = 25)
bag.boston
yhat.bag = predict(bag.boston,newdata = Boston[-train,])
mean((yhat.bag - boston.test)^2)
```

Question 5: What method do we use to get the different trees?

We use bootstrapping to get the different trees.

Random Forests

Question 6: For a building a random forest of regression trees, what should be `mtry` (number of predictors to consider at each split)?

(Number of predictors) / 3

Type and run the following in `R`:

```
set.seed(10)
rf.boston = randomForest(medv ~., data = Boston,
                         subset = train,
                         mtry = (ncol(Boston)-1)/3,
                         importance = TRUE)
yhat.rf = predict(rf.boston,newdata = Boston[-train,])
mean((yhat.rf - boston.test)^2)
```

Question 7: Compare the *MSE* of the test data to the *MSE* of the bagging.

Test has a smaller MSE.

Question 8: Use the `importance()` function what are the two most important variables?

The two most important variables are rm & lstat.

```
importance(rf.boston)
varImpPlot(rf.boston)
```

Boosting

Run the following in R:

```
library(gbm)
```

```
## Loaded gbm 2.1.8.1
```

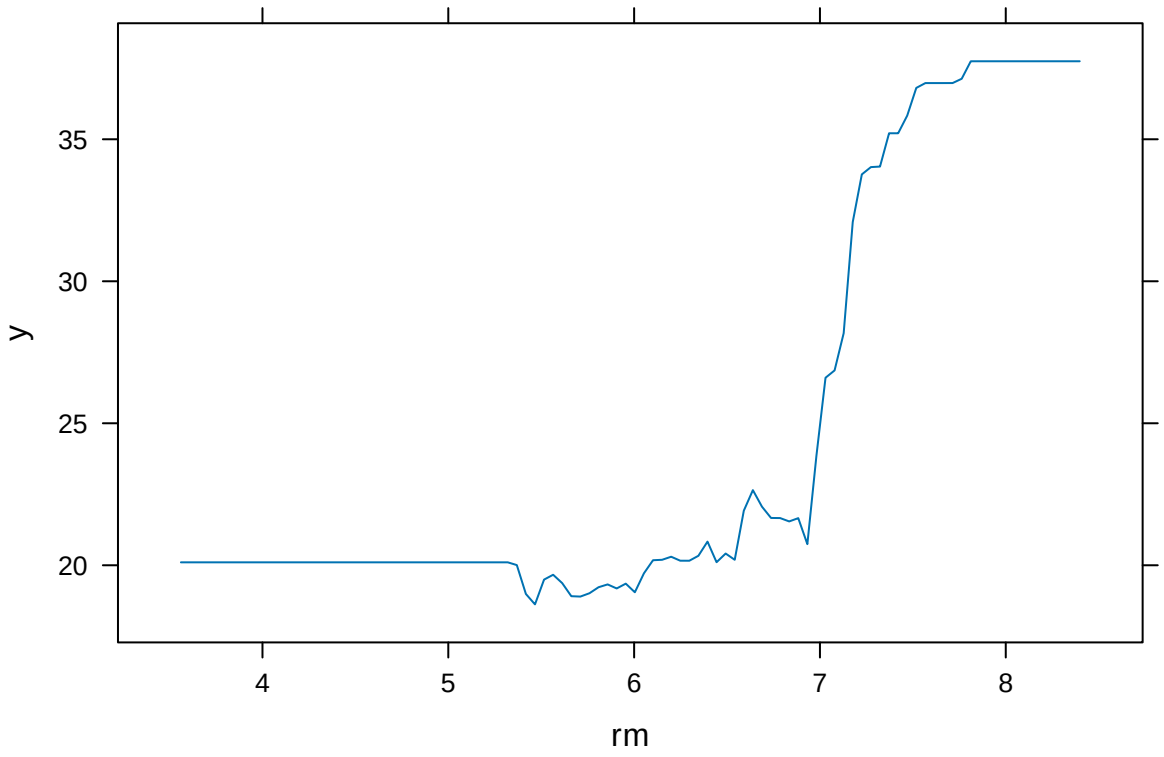
```
set.seed(1)
boost.boston = gbm(medv ~., data = Boston[train,],
                   distribution = "gaussian",
                   n.trees = 5000,
                   interaction.depth = 4)
summary(boost.boston)
```

Question 9: What are the two most important variables with the boosted trees?

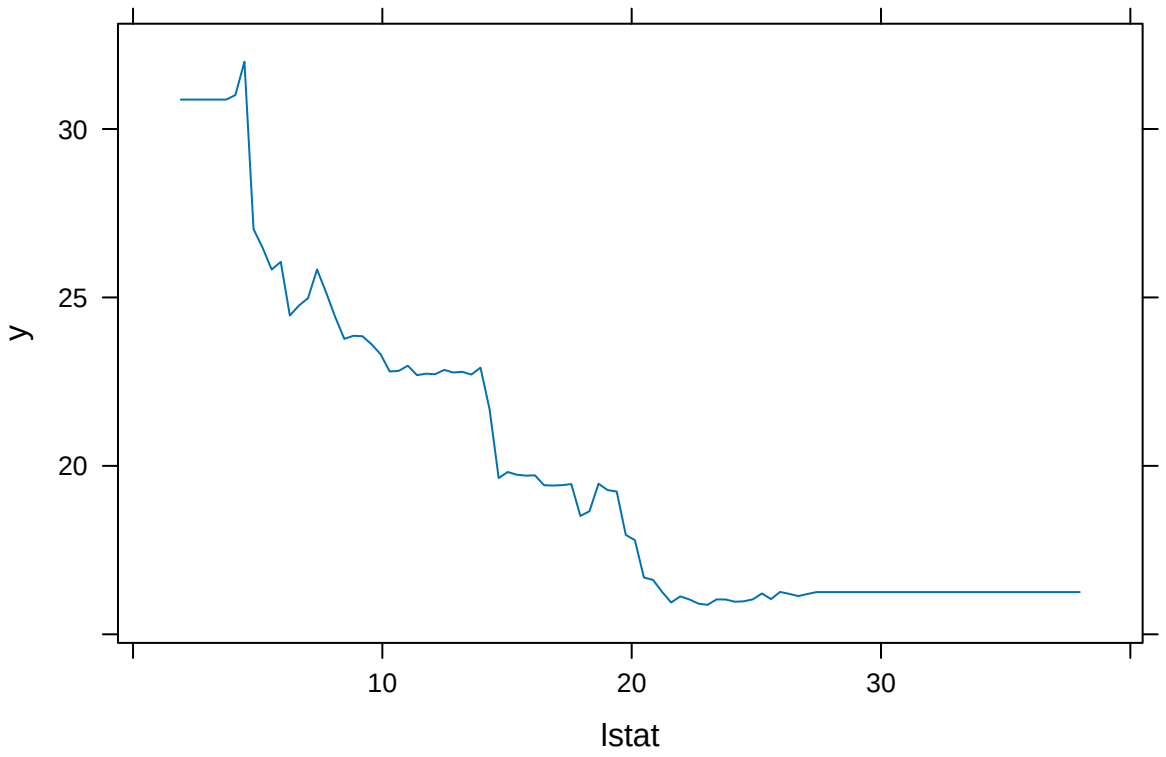
The two most important variables here are rm & lstat.

We can produce *partial dependence plots* for these two variables. The plots illustrate the marginal effect of the selected variables on the response after *integrating* out the other variables.

```
plot(boost.boston,i = "rm")
```



```
plot(boost.boston,i = "lstat")
```



Notice that the house prices are increasing with `rm` and decreasing with `lstat`.

We will use the boosted model to predict `medv` on the test set:

```
yhat.boost = predict(boost.boston,
                     newdata = Boston[-train,],
                     n.trees = 5000)
mean((yhat.boost - boston.test)^2)
```

Question 10: Compare this *MSE* to the *MSE* of the random forest and bagging models.

The MSE is smaller than both others but close to the test set's MSE.