Problem Set 3

Seth Harrison

2/17/2020

```
library(tidyverse)
library(gbm)
library(rsample)
library(randomForest)
library(stats)
```

Set Up

```
set.seed(1)

nes2008 <- read_csv("./data/nes2008.csv")

noBiden <- nes2008 %>%
            select(-nes2008$biden)

p <- ncol(noBiden)

lambda <- seq(from = 0.0001, to = 0.04, by = 0.001)
```

Create a training set consisting of 75% of the observations, and a test set with all remaining obs. Note: because you will be asked to loop over multiple λ values below, these training and test sets should only be integer values corresponding with row IDs in the data. This is a little tricky, but think about it carefully. If you try to set the training and testing sets as before, you will be unable to loop below.

```
set.seed(1)

train_ind <- sample(nrow(nes2008), size = nrow(nes2008)*.75)

train <- nes2008[train_ind,]
test <- nes2008[-train_ind,]
```

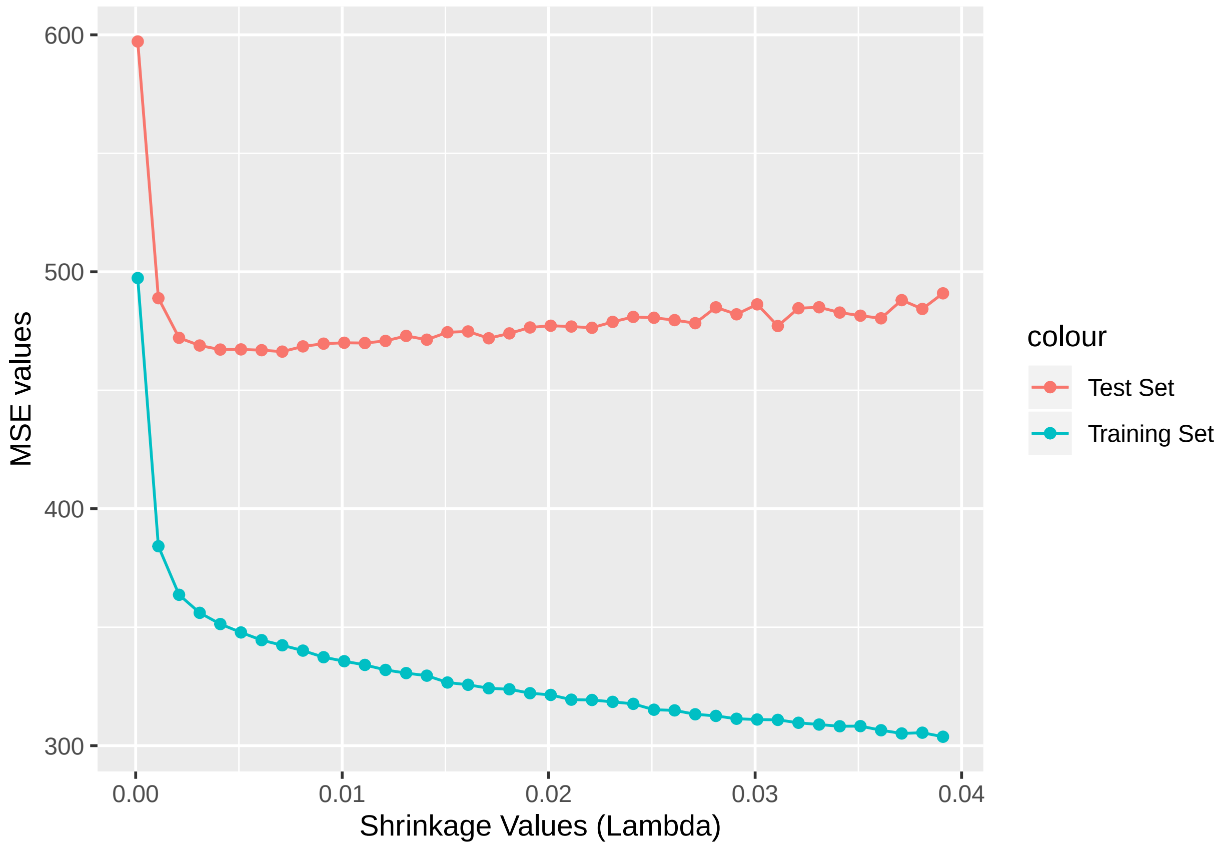
Create empty objects to store training and testing MSE, and then write a loop to perform boosting on the training set with 1,000 trees for the pre-defined range of values of the shrinkage parameter, λ. Then, plot the training set and test set MSE across shrinkage values.

```
TestMSE <- vector(mode = "numeric", length = length(lambda))

TrainingMSE <- vector(mode = "numeric", length = length(lambda))

for(i in seq_along(lambda)) {
```

```r
# boosting training set
  boost.train <- gbm(biden ~.,
                     data = train,
                     distribution = "gaussian",
                     n.trees = 1000,
                     shrinkage = lambda[i],
                     interaction.depth = 4
                     )
  training.pred <- predict(boost.train, newdata = train, n.trees = 1000)
  training.mse <- Metrics::mse(training.pred, train$biden)
# making prediction on the test set
  test.pred <- predict(boost.train, newdata = test, n.trees = 1000)
  test.mse <- Metrics::mse(test.pred, test$biden)
# extract MSE and lambda
  TrainingMSE[i] <- training.mse
  TestMSE[i] <- test.mse
  result <- cbind(lambda, TrainingMSE, TestMSE)
  result <- result %>%
    as_tibble()
}
#Plot
result %>%
  ggplot(aes(x = lambda)) +
  geom_point(aes(y = TrainingMSE, color = "Training Set")) +
  geom_point(aes(y = TestMSE, color = "Test Set")) +
  geom_line(aes(y = TrainingMSE, color = "Training Set")) +
  geom_line(aes(y = TestMSE, color = "Test Set")) +
  labs(x = "Shrinkage Values (Lambda)", y = "MSE values")
```



The test MSE values are insensitive to some precise value of λ as long as its small enough. Update the boosting procedure by setting λ equal to 0.01 (but still over 1000 trees). Report the test MSE and discuss the results. How do they compare?

```
boost.train2 <- gbm(biden ~.,

                   data = train,

                   distribution = "gaussian",

                   n.trees = 1000,

                   shrinkage = lambda[1]*100,

                   interaction.depth = 4

                   )


training.pred2 <- predict(boost.train2, newdata = train, n.trees = 1000)

training.mse2 <- Metrics::mse(training.pred2, train$biden)
```

```
# predict on the test set

test.pred2 <- predict(boost.train2, newdata = test, n.trees = 1000)

test.mse2 <- Metrics::mse(test.pred2, test$biden)

# extract MSE and lambda

TrainingMSE2 <- training.mse2

TestMSE2 <- test.mse2

result <- cbind(lambda, TrainingMSE2, TestMSE2)

result <- result %>%
            as_tibble()

TestMSE2
```

```
## [1] 470.9239
```

The MSE changes only marginally once lambda became greater than .002.

Now apply bagging to the training set. What is the test set MSE for this approach?

```
bag_biden <- randomForest(data = train,
                          x = train[,2:6],
                          y = train$biden,
                          mtry = p)

# predict on the test set

test.predbag <- predict(bag_biden, newdata = test)

test.msebag <- Metrics::mse(test.predbag, test$biden)

test.msebag
```

```
## [1] 550.5081
```

Now apply random forest to the training set. What is the test set MSE for this approach?

```
rf_biden <- randomForest(data = train,
                          x = train [,2:6],
                          y = train$biden)

# predict on the test set

test.predrf <- predict(rf_biden, newdata = test)
```

```r
test.mserf <- Metrics::mse(test.predrf, test$biden)

test.mserf
```

```
## [1] 475.1519
```