DS-6030 Homework Module 8

Tom Lever

07/16/2023

DS 6030 | Spring 2023 | University of Virginia

7. In the lab, we applied random forests to the Boston data using `mtry = 6` and using `ntree = 25` and `ntree = 500`.

   Create a plot displaying the test error resulting from random forests on this data set for a more comprehensive range of values for mtry and ntree. You can model your plot after Figure 8.10. Describe the results obtained.

```r
library(ISLR2)
library(randomForest)
```

```
# randomForest 4.7-1.1

# Type rfNews() to see new features/changes/bug fixes.
```

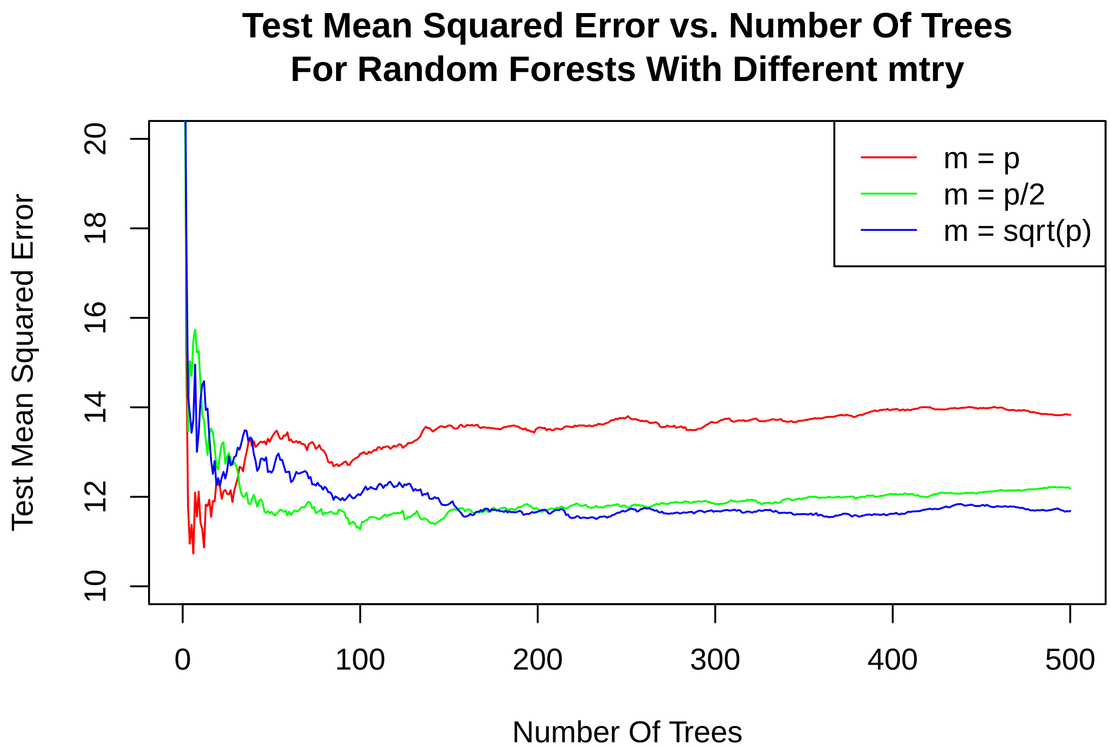
```r
library(TomLeversRPackage)

set.seed(1)
training_and_testing_data <- split_data_set_into_training_and_testing_data(
    Boston,
    proportion_of_training_data = 0.9
)
training_data <- training_and_testing_data$training_data
testing_data <- training_and_testing_data$testing_data
head(training_data, n = 3)
```

```
#        crim zn indus chas   nox    rm  age    dis rad tax ptratio lstat medv
# 505 0.10959  0 11.93    0 0.573 6.794 89.3 2.3889   1 273    21.0  6.48 22.0
# 324 0.28392  0  7.38    0 0.493 5.708 74.3 4.7211   5 287    19.6 11.74 18.5
# 167 2.01019  0 19.58    0 0.605 7.929 96.2 2.0459   5 403    14.7  3.70 50.0
```

```r
index_of_column_medv <- get_index_of_column_of_data_frame(training_data, "medv")
data_frame_of_training_predictors <- training_data[, -index_of_column_medv]
number_of_predictors <- ncol(data_frame_of_training_predictors)
data_frame_of_training_response_values <- training_data[, index_of_column_medv]
data_frame_of_testing_predictors <- testing_data[, -index_of_column_medv]
data_frame_of_testing_response_values <- testing_data[, index_of_column_medv]
randomForest_for_mtry_equal_to_number_of_predictors <- randomForest(
    x = data_frame_of_training_predictors,
```

```r
    y = data_frame_of_training_response_values,
    xtest = data_frame_of_testing_predictors,
    ytest = data_frame_of_testing_response_values,
    mtry = number_of_predictors,
    ntree = 500
)
randomForest_for_mtry_equal_to_half_number_of_predictors <- randomForest(
    x = data_frame_of_training_predictors,
    y = data_frame_of_training_response_values,
    xtest = data_frame_of_testing_predictors,
    ytest = data_frame_of_testing_response_values,
    mtry = number_of_predictors / 2,
    ntree = 500
)
randomForest_for_mtry_equal_to_square_root_of_number_of_predictors <- randomForest(
    x = data_frame_of_training_predictors,
    y = data_frame_of_training_response_values,
    xtest = data_frame_of_testing_predictors,
    ytest = data_frame_of_testing_response_values,
    mtry = sqrt(number_of_predictors),
    ntree = 500
)
plot(
    x = 1:500,
    y = randomForest_for_mtry_equal_to_number_of_predictors$test$mse,
    ylim = c(10, 20),
    col = "red",
    type = "l",
    xlab = "Number Of Trees",
    ylab = "Test Mean Squared Error",
    main = "Test Mean Squared Error vs. Number Of Trees\nFor Random Forests With Different mtry"
)
lines(
    x = 1:500,
    y = randomForest_for_mtry_equal_to_half_number_of_predictors$test$mse,
    col = "green"
)
lines(
    x = 1:500,
    y = randomForest_for_mtry_equal_to_square_root_of_number_of_predictors$test$mse,
    col = "blue"
)
legend(
    x = "topright",
    legend = c("m = p", "m = p/2", "m = sqrt(p)"),
    col = c("red", "green", "blue"),
    lty = 1
)
```



Above is a plot of Test Mean Squared Error for random forests predicting median value of owner-occupied homes in thousands of dollars based on the other variables of data set `ISLR2::Boston`. Variable *mtry* represents the number of variables considered at each split. Red, green, and blue curves correspond to random forests with *mtry* equal to the number of predictors, half the number of predictors, and the square root of the number of predictors, respectively. For each curve, Test Mean Squared Error decreases exponentially with number of trees. A random forest with *mtry* equal to the number of predictors and a number of trees less than 25 has the lowest Test Mean Squared Error and performs best.

8. This question uses the `Caravan` data set.

   (a) Create a training set consisting of the first 1,000 observations, and a test set consisting of the remaining observations.

```r
set.seed(1)
training_data <- Caravan[1:1000, ]
testing_data <- Caravan[-c(1:1000), ]
```

(b) Fit a boosting model to the training set with `Purchase` as the response and the other variables as predictors. Use 1,000 trees, and a shrinkage value of 0.01. Which predictors appear to be the most important?

(c) Use the boosting model to predict the response on the test data. Predict that a person will make a purchase if the estimated probability of purchase is greater than 20 %. Form a confusion matrix. What fraction of the people predicted to make a purchase do in fact make one? How does this compare with the results obtained from applying KNN or logistic regression to this data set?