NYC DATA SCIENCE ACADEMY

NYC Data Science Bootcamp

Trees, Bagging, Random Forests, & Boosting

Question #1: Trees

Load the `OJ` dataset from the `ISLR` library into your workspace. The data contains 1,070 purchases where the customer either purchased Citrus Hill or Minute Maid orange juice. A number of characteristics of the customer and product are recorded.

1. Split the data into a training and test set with an 80% - 20% split, respectively. (**NB:** Use `set.seed(0)` so your results will be reproducible.)
2. Construct an initial decision tree predicting `Purchase` from all other variables in the training dataset defining splits based upon the Gini coefficient.
3. How many terminal nodes are there in your initial tree? What is the accuracy of your initial tree?
4. Predict the `Purchase` variable for observations that are within your test set using this initial tree. Report the accuracy of your predictions.
5. Implement cross-validation and, thus, cost-complexity pruning to determine how far back to prune your tree. (**NB:** Use `set.seed(0)` so your results will be reproducible.)
6. Visualize your results from part 5 across various numbers of terminal nodes/values for alpha.
7. Prune your tree based on the results of part 6.
8. How many terminal nodes are there in your pruned tree? What is the accuracy of your pruned tree?
9. Visualize your pruned tree.
10. Predict the `Purchase` variable for observations that are within your test set using this pruned tree. Report the accuracy of your predictions.
11. Why are the test set predictions more accurate for the pruned tree than those for the initial tree?

Question #2: Bagging & Random Forests

Continue using the `OJ` dataset and the training/test sets you already loaded into your workspace.

1. Construct an initial random forest predicting `Purchase` from all other variables in the training dataset using the default settings; this will create 500 trees. (**NB:** Use `set.seed(0)` so your results will be reproducible.)
2. What is the accuracy of this initial random forest on:
    a. The training set?
    b. The test set?
3. Which variable is aiding the most in classifying the orange juice purchases?
4. Vary the number of variables considered as candidates at each node split in the random forest procedure (from one to all predictors). Record the out-of-bag error rates for each of these random forests on the training set. (**NB:** Use `set.seed(0)` so your results will be reproducible.) (**Hint:** You will want to record the error rate instead of the MSE since this is a classification problem. If you are modifying class code, try using the code snippet `fit$err.rate[500, 1]`.)
5. Visualize the out-of-bag error rates as they change with the number of variables considered at each node split.
6. What is the maximum accuracy among your random forests on the training set? How many variables were considered at each split in this best random forest?
7. What is the accuracy of the bagged model on the training set? How many variables were considered at each split in this bagged model?
8. What is the accuracy of the best random forest from part 6 on the test set? (**NB:** Use `set.seed(0)` so your results will be reproducible.)
9. What is the accuracy of the bagged model on the test set? (**NB:** Use `set.seed(0)` so your results will be reproducible.)

Question #3: Boosting

Continue using the `OJ` dataset and the training/test sets you already loaded into your workspace.

1. In order to boost with classification trees, we need to do a bit of data munging to transform the response variable. You may use the following lines of code to produce the copies of your dataset `OJ.train.indicator` and `OJ.test.indicator` that have a transformed response variable. (**NB:** You must replace `OJ.train` and `OJ.test` with whatever names you used in your own code.)

```
OJ.train.indicator = OJ.train

OJ.test.indicator = OJ.test

OJ.train.indicator$Purchase = as.vector(OJ.train$Purchase, mode = "numeric") - 1

OJ.test.indicator$Purchase = as.vector(OJ.test$Purchase, mode = "numeric") - 1
```

2. Construct an initial boosted model on the training set that uses all of the following settings at once: (**NB:** Use `set.seed(0)` so your results will be reproducible.)
    a. The Bernoulli distribution.
    b. 10,000 trees.
    c. An interaction depth of 4.
    d. A shrinkage parameter of 0.001.
3. Predict your test set observations using the initial boosted model across up to 10,000 trees, considering groups of 100 trees at a time. (**Hint:** Use `type = "response")` and round your ultimate predictions.)
4. Calculate and store the accuracy for each of the 100 models considered in part 3. What is the minimum number of trees required to reach the maximum accuracy?
5. Plot the accuracies found in part 4 against the number of trees. Add to the plot:
    a. A horizontal line marking the best boosted accuracy on the test set.
    b. A horizontal line marking the best random forest accuracy on the test set.

c. A horizontal line marking the best pruned decision tree accuracy on the test set.