Machine Learning - Problem Set 4

PPHA 30545 - Professor Clapp
Winter 2021

This assignment must be handed in via Gradescope on Canvas by **11:45pm Central Time on Thursday, March 11th.** You are welcome (and encouraged!) to form study groups (of no more than 3 students) to work on the problem sets and mini-projects together. But you must write your own code and your own solutions. Please be sure to include the names of those in your group on your submission.

You should submit your code as a single Python (*.py) file and the write up of your solutions as a single PDF. For the former, please also be sure to practice the good coding practices you learned in PPHA 30535/6 and comment your code, cite any sources you consult, etc. For the latter, you may type your answers or write them out by hand and scan them (as long as they are legible).

You are allowed to consult the textbook authors' websites, Python documentation, and websites like StackOverflow for general coding questions. You are not allowed to consult material from other classes (e.g., old problem sets, exams, answer keys) or websites that post solutions to the textbook questions.

1. Do the following questions from Chapter 6 of the *Introduction to Statistical Learning* textbook:

   (a) Question 9, parts (a), (b), (e)-(g)

      i. Python does not have a PCR function, so you should use scikit-learn's PCA function (aptly called *PCA*), then run an OLS regression using the resulting principal components.

      ii. Scikit-learn does have a PLS regression function (*PLSRegression*).

2. Do the following questions from Chapter 8 of the *Introduction to Statistical Learning* textbook:

   (a) Question 4

   (b) This question is a modified version of Question 9. It involves the OJ data set which is available on Canvas.

      i. Create a training set containing a random sample of 800 observations, and a test set containing the remaining observations.

      ii. Fit a tree to the training data, with *Purchase* as the response and the other variables as predictors. What is the training error rate? (Set the *max_depth* parameter to 3 in order to get an interpretable plot)

      iii. Create a plot of the tree. How many terminal nodes does the tree have? Pick one of the terminal nodes, and interpret the information displayed.

      iv. Predict the response on the test data, and produce a confusion matrix comparing the test labels to the predicted test labels. What is the test error rate?

v. Determine the optimal tree size by tuning the *ccp_alpha* argument in scikit-learn's *DecisionTreeClassifier*.[1] You can use *GridSearchCV* for this purpose.

vi. Produce a plot with tree size on the x-axis and cross-validated classification error rate on the y-axis calculated using the method in the previous question. Which tree size corresponds to the lowest cross-validated classification error rate?

vii. Produce a pruned tree corresponding to the optimal tree size obtained using cross-validation. If cross-validation does not lead to selection of a pruned tree, then create a pruned tree with five terminal nodes.

viii. Compare the training error rates between the pruned and unpruned trees. Which is higher?

ix. Compare the test error rates between the pruned and unpruned trees. Which is higher?

[1] Note that by the nature of how pruning works, tree size is a function of the α hyperparameter.