Machine Learning for Public Policy - Problem Set 4

The University of Chicago - Harris School of Public Policy
PPHA 30545 - Professors Clapp and Levy
Winter 2025

This assignment must be handed in via Gradescope on Canvas by **11:45pm Central Time on Thursday, March 6th.** There will be separate Gradescope assignments for R and Python students. Please be sure to submit to the version that matches the coding language of the lab section you are enrolled in.

You are welcome (and encouraged!) to form study groups (of no more than 2 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 format your submission in one of two ways:

1. As a single PDF containing BOTH a write-up of your solutions that directly integrates any relevant supporting output from your code (e.g., estimates, tables, figures) AND your code appended to the end of your write up. You may type your answers or write them out by hand and scan them (as long as they are legible). Your original code may be a R (*.rmd) or Python (*.py) file converted to PDF format. OR

2. As a single PDF of an R Markdown (*.rmd), Jupyter Notebook (*.ipynb), or Quarto (*.qmd) document with your your solutions and explanations written in Markdown.[1]

Regardless of how you format your answers, be sure to make it clear what question you are answering by labeling the sections of your write up well and assigning your answers to the appropriate question in Gradescope. Also, be sure that it is immediately obvious what supporting output from your code (e.g., estimates, tables, figures) you are referring to in your answers. In addition, your answers should be direct and concise. Points will be taken off for including extraneous information, even if part of your answer is correct. You may use bullet points if they are beneficial. Finally, for your code, please also be sure to practice the good coding practices you learned in Data and Programming and comment your code, cite any sources you consult, etc.

You are allowed to consult the textbook authors' website, R/Python documentation, and websites like StackOverflow for general coding questions. If you get help from a large language model (LLM) or other AI tool (e.g., ChatGPT), you must provide in the query string you used and an explanation of how you used the AI tool's response as part of your answer. 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] Converting a Jupyter Notebook to PDF is not always straightforward (e.g., some methods don't wrap text properly). Please ensure that your PDF is legible! We will deduct points if we cannot read your PDF (even if you have the correct answers in your Notebook).

1. (ISL: Chapter 6, Question 9) In this exercise, we will predict the number of applications received using the other variables in the `College` data set.

   (a) Split the data set into a training set and a test set. Please use a 50/50 training/test split. To avoid confusion among partners and facilitate grading, Python students should set `random_state=37`, and R students should set `set.seed(37)` when splitting the data. Be sure to scale your predictors (for models that require it).[2]

   (b) Fit a linear model using least squares on the training set, and report the test error obtained.

   (c) Fit a principal components regression (PCR) model on the training set, with M chosen by cross-validation.[3] Use 10-fold cross-validation (10FCV) on the training set, shuffle the data randomly for splitting, and set `random_state=1` (Python) or set `set.seed(37)` (R). Report the test error obtained, along with the value of M selected by cross-validation, both by minimizing the appropriate cross-validated error and using the "elbow method."

   (d) Fit a partial least squares (PLS) model on the training set, with M chosen by cross-validation, using the same cross-validation settings as given in the previous question. Report the test error obtained, along with the value of M selected by cross-validation, both by minimizing the appropriate cross-validated error and using the "elbow method."

   (e) Comment on the results obtained. How accurately can we predict the number of college applications received? Is there much difference among the test errors resulting from these approaches?

2. (ISL: Chapter 8, Question 4) This question relates to the plots in the figure that follows (ISL Figure 8.14).

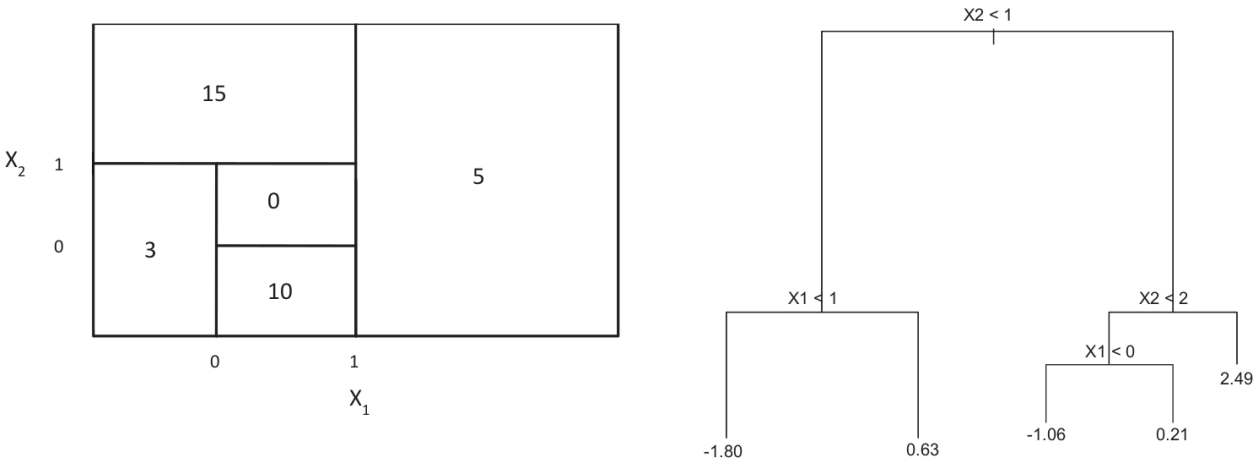

FIGURE 8.14. Left: A partition of the predictor space corresponding to Exercise 4a. Right: A tree corresponding to Exercise 4b.

Source: James, Witten, Hastie & Tibshirani (2023)

[2]There are several ways to scale or normalize variables. Scaling here refers to transforming each predictor so that the mean is 0 and the variance is 1. Please scale *after* splitting your data.

[3]R has a PCR command, but Python does not. Python users should perform principal components analysis (PCA) via `scikit-learn`'s `PCA` command, then run an OLS regression using the resulting principal components.

(a) Sketch the tree corresponding to the partition of the predictor space illustrated in the left-hand panel of the figure. The numbers inside the boxes indicate the mean of Y within each region.

(b) Create a diagram similar to the left-hand panel of the figure, using the tree illustrated in the right-hand panel of the same figure. You should divide up the predictor space into the correct regions, and indicate the mean for each region.

3. (ISL: Chapter 8, Question 9) This question involves the `OJ` data set which is available on Canvas.[4]

(a) Create a training set and a test set. Please use a 70/30 training/test split. Python students should set `random_state=3`, and R students should set `set.seed(3)` when splitting the data.[5]

(b) Fit a full, unpruned tree to the training data, with `Purchase` as the response and the other variables as predictors. When calling the `DecisionTreeClassifier()` function in Python, please set `random_state=2`.[6] There is not an analogous argument in the `rpart()` function in (R). What is the training error rate?

(c) Create a plot of the full, unpruned tree from the previous question. The plot is a mess, isn't it? For the purposes of this question, fit another tree with a maximum depth of 3 in order to get an interpretable plot. How many terminal nodes does the tree have? Interpret and explain the information displayed in the first (when reading from left to right) of the terminal nodes on your plot.

(d) Use your fit of the full, unpruned tree to 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?

(e) Use cost complexity pruning to determine the optimal subtree for prediction by tuning the the α hyperparameter.Use 5-fold cross-validation (5FCV) to choose the optimal value of the hyperparameter. Use the training dataset for 5FCV, shuffle the data randomly for splitting, and set `random_state=13` (Python) or set `set.seed(13)` (R). Produce a plot with the values of α on the x-axis and the cross-validated classification error rate on the y-axis. Which α corresponds to the lowest cross-validated classification error rate?

(f) Now produce a second plot showing the tree size on the x-axis and the cross-validated classification error rate (that you calculated in the method in the previous question) on the y-axis.[7] Which tree size corresponds to the lowest cross-validated classification error rate? Briefly explain why the value of α affects the tree size and the classification error rate.

[4] For variable definitions, see https://rdrr.io/cran/ISLR/man/OJ.html.

[5] Note that there are some redundant predictors in the dataset. You can ignore this complication and use the full dataset for prediction.

[6] Ties between different splits that result in the same improvement in the loss function are broken randomly. This ensures that they will be broken in the same way.

[7] Note that tree size is the number of terminal nodes or leaves.

(g) Produce a plot of the optimal pruned subtree obtained using cross-validation.

(h) Compare the training error rates between the pruned and unpruned trees. Which is higher? Briefly explain.

(i) Compare the test error rates between the pruned and unpruned trees. Which is higher? Briefly explain.