Machine Learning for Public Policy - Problem Set 3

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, February 20th.** 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 5, Question 6) We continue to consider the use of a logistic regression model to predict the probability of `default` using `income` and `balance` on the `Default` data set. In particular, we will now compute estimates for the standard errors of the `income` and `balance` logistic regression coefficients in two different ways: (1) using the bootstrap, and (2) using the standard formula for computing the standard errors in the `glm()` function (R) or the `sm.GLM()` function (Python).[2] Do not forget to set a random seed (equal to 23) before beginning your analysis.

   (a) Using the `summary()` and `glm()` functions (R) or the `sm.GLM()` function and the `.summary()` method (Python), determine the estimated standard errors for the coefficients associated with `income` and `balance` in a multiple logistic regression model that uses both predictors.[3]

   (b) Write a function, `boot_fn()`, that takes as input the `Default` data set as well as an index of the observations, and that outputs the coefficient estimates for `income` and `balance` in the multiple logistic regression model.

   (c) Following the bootstrap example in the lab, use your `boot_fn()` function to estimate the standard errors of the logistic regression coefficients for `income` and `balance`. Please draw 1,000 bootstrap samples when bootstrapping your standard errors.

   (d) Comment on the estimated standard errors obtained using the `glm()` function (R) or the `sm.GLM()` function (Python) function and using the bootstrap.

2. (ISL: Chapter 5, Question 8) We will now perform cross-validation on a simulated data set.

   (a) Generate a simulated data set as follows:

   - R:

```
> set.seed(1)
> x <- rnorm(100)
> y <- x - 2 * x^2 + rnorm(100)
```

   - Python:

```
rng = np.random.default_rng(1)
x = rng.normal(size=100)
y = x - 2 * x**2 + rng.normal(size=100)
```

   In this data set, what is n and what is p? Write out the model used to generate the data in equation form.

   (b) Create a scatterplot of X against Y. Comment on what you find.

[2] Python users: the textbook problem references the `sm.GLM()` function. Whenever that function is referenced in this question, you can use the `sm.Logit()` function instead if you'd prefer. The `sm.GLM()` function is more general way to estimate different linear models, hence the General Linear Models (GLM) name, but it will estimate a logistic regression with the appropriate arguments.

[3] Python users: the textbook question references the `summarize()` function. This a custom function in the `ISLP` package. Alternatively, you can use the `.summary()` method as a way to view the results of a `sm.GLM()` or `sm.Logit()` fit.

(c) Using the simulated data you generated in Question (2a), set a random seed equal to 10, then compute the LOOCV errors that result from fitting the following four models using least squares:

i. $Y = \beta_0 + \beta_1 X + \varepsilon$
ii. $Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \varepsilon$
iii. $Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 + \varepsilon$
iv. $Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 + \beta_4 X^4 + \varepsilon$.

Note you may find it helpful to use the `data.frame()` function (both R and Python) to create a single data set containing both X and Y.

(d) Repeat Question (2c) using a random seed equal to 20, and report your results. Are your results the same as what you got in Question (2c)? Why?

(e) Which of the models in Question (2c) had the smallest LOOCV error? Is this what you expected? Explain your answer.

(f) Comment on the statistical significance of the coefficient estimates that results from fitting each of the models in Question (2c) using least squares. Do these results agree with the conclusions drawn based on the cross-validation results?

3. (ISL: Chapter 6, Question 11) We will now try to predict the per capita crime rate in the `Boston` data set.

(a) Use best subset, forward stepwise, and backward stepwise selection on this data set to select the predictors that produce the best fit.[4] Do so by using 5-Fold Cross-Validation (5FCV) to estimate the test error. Use the entire dataset for 5FCV, shuffle the data randomly when splitting, and set `random_state=24` (Python) or set `set.seed(42)` (R). Time how long it takes you to run each selection algorithm. Present and discuss results for the three selection approaches that you consider.

(b) Fit three models each using one of the preferred set of predictors picked by the three feature selection algorithms (best subset, forward stepwise, and backward stepwise) in Question (3a). Which of those models gives the best prediction? Evaluate model performance using both 5FCV and the Akaike Information Criterion (AIC). Explain your answer. In doing so, be sure to explain why the different selection methods you use in the previous quesiton may select different models, why using 5FCV or the AIC is preferred to directly using the training error, and why 5FCV and the AIC may give different rankings of the models.

(c) Does your chosen model involve all of the features in the data set? Why or why not?

[4]The textbook question asks you to try a different list of methods covered in the chapter. You only need to use the listed feature selection methods.