Machine Learning for Public Policy - Mini-Project 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 Friday, March 14th.** 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 Overview

Ongoing concerns about election fraud have led to a wave of new policies designed to ensure the integrity of the vote. Unfortunately, there are concerns that these policies may also cause voter suppression. This has led to calls to make Election Day a national holiday as a partial way to address those concerns.[2] The intuition behind this proposed reform is simple: making Election Day a holiday will improve voter turnout by giving people the time to vote without needing to take time off from work. But is there empirical evidence that suggests work constraints prevent people from voting?

In order to inform potential policy, you are tasked with answering the following related question: is having flexible work hours associated with being more likely to vote? At your disposal are two datasets, "vote.csv" and "work.csv," that we will call `df_vote` and `df_work`.

Both come from the U.S. Census Bureau and U.S. Bureau of Labor Statistics' Current Population Survey (CPS).[3] In addition to being used to calculate monthly labor force statistics, the CPS provides information about a variety of economic and social well-being topics via supplemental questions that are asked to a rotating subset of the CPS sample. Because of the rotating nature of these supplements, individuals who are asked questions for one supplement are usually not asked asked the questions in other supplements.

Table 1 contains the names of the variables available in each dataset and their definitions. The `df_vote` dataset has a binary indicator variable, which is appropriately called `vote`, that indicates the voting status of each individual in the data. Meanwhile, the `df_work` dataset has its own binary indicator variable called `work` that indicates whether individuals have flexible work schedules. The two datasets share a set of core predictors.

Table 1: Variable Names and Definitions

		Dataset	
Variable	**Definition**	**Vote**	**Work**
`vote`	Person voted in the last election	✓	
`work`	Person has a flexible work schedule		✓
`prtage`	Age	✓	✓
`pesex`	Sex	✓	✓
`ptdtrace`	Race	✓	✓
`pehspnon`	Hispanic origin	✓	✓
`prcitshp`	U.S. citizenship status	✓	✓
`peeduca`	Highest level of schooling	✓	✓

Individuals in one dataset are almost assuredly different than the individuals in the other dataset. As a result, for any individual in our data, we will either know their voting status or their work

[2] See, for instance, this Brookings Institution blog post.

[3] Specifically, the `df_vote` dataset is based on the CPS Voting and Registration Supplement. The `df_work` dataset is based on the CPS Work Schedules Supplement. Note that these semi-synthetic datasets were created for pedagogical reasons, so results should be viewed accordingly.

schedule, but we cannot know both simultaneously. Hence, we have a missing data problem. The plan to overcome this challenge is as follows:

- First, explore and clean the data.
- Train a Support Vector Machine (SVM) classifier on `df_work` that uses the demographic variables to forecast whether someone has flexible work hours.
- Apply the SVM classifier from the previous step to `df_vote` to predict whether the people in that dataset have flexible work hours.
- Regress voting status on the predictions obtained in the previous step.
- Adjust our regression estimate to account for measurement error in the imputed flexible work hours measure.

2 Data Analysis

1. Report the data type of each variable in each dataset.
2. Note that all the variables in our dataset except for `prtage` are categorical, meaning that they take discrete values as opposed continuous values. Python and R's SVM classifiers only accept numerical values for training.
   (a) First convert the target variables (`vote` and `work`) to binary forms by mapping them to 0s and 1s.
   (b) For each of the remaining categorical variables compare the categories in the version of the variable in the `df_vote` and `df_work` datasets. Are there any discrepancies?
   (c) Convert the categorical variables using a technique called "one-hot encoding" by creating multiple binary variables corresponding to each category.
   (d) We want each core variable between our two datasets to have the same structure for our prediction exercise, so whenever there's a discrepancy between the categories reported for a given variable, adjust the data to ensure that the core variables have the same structure between the `df_vote` and `df_work` datasets.

       For instance, the core variable `prcitship` in the `df_work` dataset has five categories while `prcitship` in the `df_vote` dataset has four. So add a column that's all zeros for the "`FOREIGN BORN, NOT A CITIZEN OF`" citizenship category after one-hot encoding `prcitshp` in the `df_vote` dataset.
   (e) Be sure to scale the predictors before fitting the model by transforming all continuous predictors to be in the range $[0,1]$. Briefly explain why you need to scale the data before fitting the SVM classifier.

3. Now that the datasets are set up, train a SVM classifier on the `work` data that fits the flexible work variable as a function of the core variables (and a constant). There are several choices to make when fitting a SVM, mainly the regularization or *cost hyperparameter* (C) that penalizes observations that violate the margin/hyperplane and the *kernel* that introduces non-linearity to the SVM. Consider four values of hyperparameter C: `0.1`, `1`, `5`, and `10` and three kernels: linear, radial, and sigmoid. Use 5-fold cross-validation (5FCV) to determine which cost and kernel to use. Shuffle the data randomly for splitting, and set `random_state=26` (Python) and set `set.seed(26)` (R). Report the cross-validation error rates of all 12 SVM models.[4]

4. Pick and report the value of C and kernel that minimize the 5FCV error rate. Use this model for the rest of the exercises.

5. What is the accuracy score of the model that you decided on when fitting to the `df_work` data in the previous question?

6. With the SVM model that you fit on `df_work`, predict the work schedules using the core variables from `df_vote`. The result is the imputed work flexibility measure needed for the main analysis. Compute and report descriptive (summary) statistics for the imputed measure.

7. Regress voting status on the imputed work schedule. Use age, age squared, and sex as predictors in addition to the imputed work schedule. Report, briefly interpret, and discuss the results.

8. Since we imputed the work schedules, there are likely to be some forecasts that are incorrect. To account for the bias this measurement error causes, we will need to divide the estimate of the parameter of interest by a scaling function

$$M(a,b) = \frac{1}{1-2b}\left(1 - \frac{(1-b)\,b}{a} - \frac{(1-b)\,b}{1-a}\right),$$

where $a = \Pr\left(\widehat{work} = \text{"}flexible\text{"}\right)$, $b = \Pr\left(\widehat{work}\ is\ incorrectly\ labeled\right)$, and $\widehat{work}$ is the imputed value of the work variable from the SVM.[5] Write a simple function to compute $M(a,b)$. For the value of a, find the proportion of imputed work schedules that are flexible. For the value of b, use the cross-validation error rate from Question 3 . Report a, b, and $M(a,b)$.

9. Correct for the attenuation bias in your results from Question 7 . Is the bias corrected version larger or smaller? Does the bias-correction change your previous result? Briefly explain.

[4] Note that it may take some time to run this code.

[5] Note that this scaling function $M(\cdot)$ is different from the margin defined in the SVM slides.