Programming Lab #3

Machine Learning 1

Programming Lab 3

The purpose of this project is to gerrymander Virginia.

The data include:

- `voting_VA.csv` and `countypres_2000-2020.csv`: Voting data for presidential elections for Virginia, and the same data for the whole country
- `nhgis_county_data`: A folder containing many county-level summary stats for every county in the U.S. This is the most complete county-level data I could find. If you go to the IPUMS NHGIS web site, you can see what else is available (there are hundreds of variables, and I chose a large number of obvious ones; perhaps some useful ones escaped my attention). For standard IPUMS microdata, the county is not available for privacy reasons.
- `VA_districts.txt`: I looked up which voting districts currently contain which counties, roughly.

There are 134 counties in Virginia that must be allocated into 11 voting districts. In reality, voting districts do not need to respect county boundaries, but this simplifying assumption makes the project much more manageable.

You can use whatever data you want to create a predictive algorithm for voting outcomes, based on the `voting_VA.csv` and `nhgis_county_data` data or other sources you think would be useful. You can focus on Virginia data, but in principle, you could use data from the entire country. Since you only have five observations for each county on its own in Virginia, you can, in principle, use the additional data about county composition or data from other states to build richer and more powerful predictive models than just the sample average for each county (e.g 3 observations of `D` winning and 2 of `R` winning implies a probability 3/5 of `D` winning). You could also gather and use data about past candidates to see if there are county-candidate interaction effects that improve your model's performance. Indeed, 2024 might be a Biden-Trump rematch, in which case past data might be extremely relevant.

Once you have your predictive model for how each county votes, you can simulate outcomes at the county level for an election. Now you need to maximize the vote count for the Red party (Republicans) and the Blue party (Democrats). You have 11 voting districts and 167 counties. The rules for this are fairly complex in the real world, but we will keep it simple:

1. Voting districts must be *contiguous*, so that you can travel from any point to any other point within a voting district without leaving the voting district. For us, the counties must be adjacent, so that their borders touch.
2. Voting districts must be *proportional*, so they are roughly of equal size in terms of population. We'll try to keep it simple: Your largest voting district cannot be more than 5% larger in terms of population than your smallest voting district. (Currently, the eighth district has a population of 770k and the first has a population of 805k, so 4.55% larger.)

This kind of problem is difficult to solve (it is similar to a "bin packing" problem, known to computationally difficult; i.e. NP-Hard). I would not use a direct approach (e.g. linear programming), but would instead consider starting with an "electoral map" similar to what currently exists based on `VA_districts.txt` and then writing a random algorithm that looks to flip counties in ways that don't violate the constraints and do improve the predicted outcomes for the `R` and `D` parties. Or, I might start by finding the 11 largest counties

that are most supportive of the party I'm optimizing for, and then expanding from those, adding additional adjacent counties to each district that maintain proportionality and contiguity but minimize the number of districts that the other party wins.

Here is cynical advice about approaching this problem: The trick with gerrymandering is to include just enough counties for the party whose vote score you're maximizing to win that voting district, and then adding counties that support the other party to eliminate their power; conversely, if a voting district is going to the other party with high probability, maximize the number of counties that support the other party that are packed into that district. Essentially, you want to dilute the power of the other party by packaging their supporters into voting districts that are already decisive for your party.

The purpose of this final project is to integrate the data science piece (predicting electoral outcomes at the county level) into a decision-making system (the gerrymandering). If the second part becomes impossible, that is OK: It's an ambitious problem, and you are wrapping up the semester. Do your best, explain what you tried and why, and if it doesn't work, explain why not. If the contiguity and proportionality constraints are too hard to satisfy, drop one or the other and see what you get. But the predictive models are typically not useful unless they're integrated into a larger project.

If you find the premise of the project objectionable, I understand. We're not asserting that gerrymandering is *good*, we're asking how self-interested parties would go about doing it. On the other hand, there are important questions that this kind of analysis can inform. "Does the real electoral map look more like the `R`-optimal or `D`-optimal outcome?" "Do parties gain power by isolating and disenfranchising disadvantaged groups in ways that would constitute a violation of the Civil Rights Act?" Being able to predict, optimize, and unpack this problem opens up all kinds of possibilities for understanding the electoral system better, and improving its design.

The biggest advice would be: Keep things simple and scale up as you iterate. Don't try to use 1000 variables right away, and solve the gerrymandering problem perfectly. Write your code so that it scales, and work your way from something simple and feasible to something complex and powerful. You will make more progress faster that way, it will take less time, and it will be more engaging and rewarding.

Paper format

The format of the paper should be:

- Summary: A one paragraph description of the question, methods, and results (about 350 words).
- Data: One to two pages discussing the data and key variables, and any challenges in reading, cleaning, and preparing them for analysis.
- Results: Two to five pages providing visualizations, statistics, tables, a discussion of your methodology, and a presentation of your main results. In particular, how are you approaching the prediction and optimization problems? How confident are you about your assessments that counties will support one party or the other? Which party seems to have an advantage in terms of drawing the electoral map?
- Conclusion: One to two pages summarizing the project, defending it from criticism, and suggesting additional work that was outside the scope of the project.
- Appendix: If you have a significant number of additional plots or table that you feel are essential to the project, you can put any amount of extra content at the end and reference it from the body of the paper.

Submission

Each student should upload a zip folder to the Assignments tab on Canvas, which includes

- .R or .Rmd files that clean the data and conduct the analysis
- The paper, written in .Rmd format and compiled to a .html or .pdf file

Each student can submit independent work, despite being in a group, or the group can collaborate on a single submission that all members submit.

Criteria

The project is graded based on four criteria:

- Project Concept: What is the strategy for building and testing the group's predictive models? How are the models embedded in the decision problem of gerrymandering Virginia?
- Wrangling, EDA, and Visualization: How are are missing values handled? For variables with large numbers of missing values, to what extent do the data and documentation provide an explanation for the missing data? If multiple data sources are used, how are the data merged? For the main variables in the analysis, are the relevant data summarized and visualized through a histogram or kernel density plot where appropriate? Are basic quantitative features of the data addressed and explained? How are outliers characterized and addressed?
- Analysis: What are the groups' main findings? Do the tables, plots, and statistics support the conclusions? If the gerrymandering strategy succeeds, what are the results and how extreme can the map be drawn for each side? If the gerrymandering strategy fails, is there a thoughtful discussion about the challenges and limitations?
- Replication/Documentation: Is the code appropriately commented? Can the main results be replicated from the code and original data files? Are significant choices noted and explained?

Each of the four criteria are equally weighted (25 points out of 100).