QTM 347: Final Project

How much will a donor donate to a campaign?

The Problem

- Fundraising is a key focus for candidates
    - Events, “Call Time”, etc
- Costs valuable time and effort to be able to raise money from donors
- We want to maximize the time and effort spent raising money to people who are likely to donate
- This model will help candidates raise more efficiently, making campaigning more accessible for more people

Question

Can we predict how much someone will donate to a Congressional campaign given a combination of individual and location data in Pennsylvania?

Preface

- This is a data scraping and cleaning problem
  - Rate limits on APIs, so when scraping this data, I had to incorporate retries, time outs, and more to continuously fetch from the databases
  - Unclean data (but at least the way it's unclean has a pattern), so need an algorithm to parse it in order to join datasets
- Data scraping / cleaning was 80% of the task
- I don't have any conclusive findings on the relationships between donation amount and the variables selected

The Approach

1. Scrape data
2. Write algorithms to clean data
3. Join across various databases (FEC, FBI, Census)
4. Transform data (e.g. adding numerical info to columns with text, dummy variables), split data
5. Train a Linear Regression, Random Forest Regression, and XGBoost Regression Model
6. Finetune Hyperparameters
7. Report insights and compare to a baseline

Hypothesis

When political consultants or campaign managers analyze FEC data, they use their general, high-level interpretations to guess the effectiveness.

As such, I hypothesize that between FEC data, FBI crime data, and Census city-specific economic data, we can predict with better-than-random accuracy how much someone will donate

Data

Datasets

fec.gov/data/individual-contributions

150,000,000 records just from 2023 to 2024

Contains:

- Name, Recipient, City, State, Occupation, Employer, and more

City Crime and Economic Data

Crime and economic statistics that can be used to inform likelihood of donating

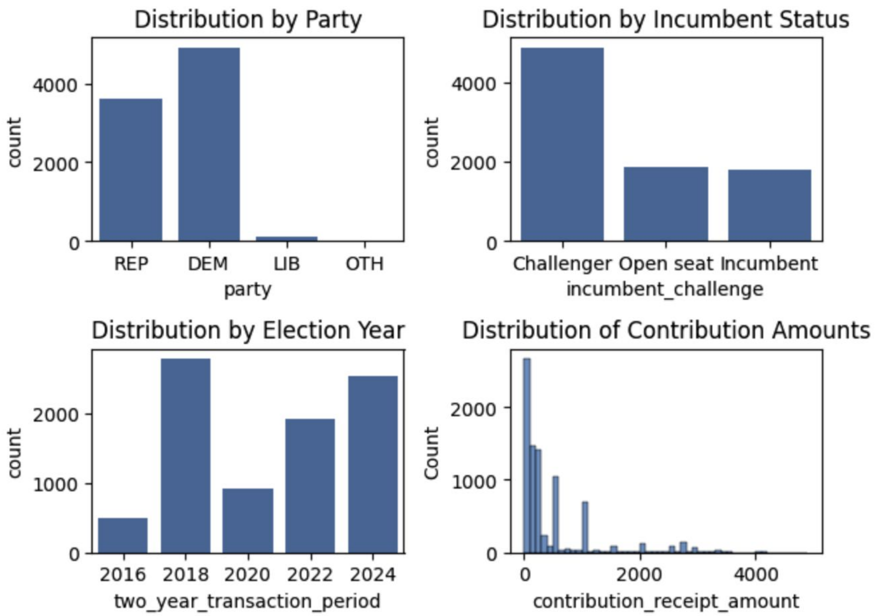
FEC API

Years: 2016, 2018, 2020, 2022

Columns: [

Party,
Incumbent v Challenger,
Contribution_Receipt_Amount,
Occupation

]



Census and FBI API

With some text matching, we can obtain the fips code for most cities in PA. This allows us to obtain row-level crime data. There was fips data up until 2022.

Years: 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022

We want to observe the % change in:

- unemployment
- property crime

For example, for 2016, we don't want to display "4% employment," we'd want to display -15% unemployment relative to 2015.

Cleaning

- FEC API contains donations from companies to candidates, individuals to companies, and many other entries that are not relevant to our analysis – I parsed these out by filtering for individual contributions to candidates running for Congress
- Handful of campaign refunds (almost negligibly few)
- Misspelled cities that needed to be dropped from the dataset (difficult to obtain fips for)
    - Could be handled with a manual mapping / using an LLM to intelligently re-spell
- There are a few massive donations – could be outliers or misentries. Unlikely that individuals are donating in multiple installations over $500k. These are difficult to

Dataset

Merge datasets together, the resulting dataframe will have the following properties:

contribution_amt	Party	Incumbent?	Unemploy_chng	Crime_chng	prev_cycle_contribution_avg	prev_cycle_contributions
1000	R	N	-0.05	-0.05	1000	10

7 Columns, ~40000 Rows

Limitations to the dataset

- Limited by the number of requests → we query the database and loop through the pages, pulling data sequentially.
- This means that rows aren't truly selected randomly, an improvement for our data may be:
  - generating a random set of of page numbers
  - navigating to those random pages
  - Pulling data from those random pages
- There may be bias in regards to which cities are spelled incorrectly (these rows are removed from our dataset)

Methodology

Data Splitting

Target Variable:

- Curr_cycle_contributions

Predictors:

- Prev_cycle_avg_contribution, crime_chng, occupation, prev_cycle_contributions, party, incumbent,

Split the data into the following chunks:

- 70% training
- 15% validation
- 15% testing

Before Analysis

1. Scale data
   a. Donations, both in the number of donations in the previous cycle and the average size of donations, is heavily skewed
2. Dummy Variables
   a. Incumbent (I, C) = 1, 0
   b. Party (R, D, I) = party_republican, party_democrat

Next Steps

1. Train 3 models: Linear Regression, Random Forest, XGBoost
    a. Linear: Interpretability – what's the specific impact of our training data on our target variable
    b. Random Forest: A quick, out-of-the-box, but powerful method to quickly obtain high quality predictions and determine if there is a relationship here
    c. XGBoost: Most robust and optimal outputs, but will require significant tuning and we may not be able to get the optimal output

Evaluation

"Baseline"

- Compare our models' performances (MSE) against using the average donation amount from the previous election cycle
- Account for the party to which individuals donate to

Limitations

Data Quality

- Data storage is highly varied across different government databases → FBI, FEC, Census contain data differently
    - There are many misnamed cities (~1/8th of the data) in the FEC database. This is very hard to deal with
    - There are many multiple matches (most of this has been cleaned away with an algorithm that identifies patterns within the text)
- We coerce poor quality entries by dropping them, but this may be removing rows discriminately

Model Issues

The data is stored in a way that treats each year independently:

- E.g. 2016 is an input for 2018, 2018 is an input for 2020, etc.

This assumption is made because the political and economic environment evolve rapidly, and applying a time series approach over 2 years may introduce noise. However, my assumption may be incorrect.

- In this sense, it may be better to approach this problem as a time series problem, which will introduce significantly higher dimensionality

Thank you