ORIE 4741 Final Project
Fatima Al-Sammak (fma29) and Alexander Ellis (ase49)

GitHub Link: https://github.com/fatimaalsammak/ORIE4741Project.git
Google Drive Link: https://drive.google.com/drive/folders/1A1GluYNeLaa7wq1qI11hmn1vDjpRVsWm?usp=sharing

Project Introduction

Question: Can we predict the voter turnout for a presidential election?

The primary dataset we will use to answer our question is the National General Election VEP (Voting Eligible Population) Turnout Rates for 1789 - Present. It provides the VEP turnout rate for each Presidential election since 1789. This dataset will be what we use for our y_i parameters, or the *actual* voter turnout. We will join this dataset by year with other datasets that measure economic and political factors to build our x vector:

1. Real Gross Domestic Product (GDP), which is the United States GDP adjusted for inflation, measures the value of the goods the United States has sold each year. This data is only available starting from 1947.
2. GINI Coefficient, which measures the economic inequality in the United States each year. This data is only available starting from 1963.
3. Census Bureau Economic Indicators, which is a compilation of various economic measures from the Census Bureau, recorded each month since August 2004. Each month's recorded value may have a different level of predictive strength since voters are more likely to be interested in voting based on the economic state of the country at different times of the year.
4. Incumbency, which is a binary variable indicating whether one of the candidates in the race is a sitting president at the time of the election. We will build this data by hand using common knowledge, which will make it easily available for all years since 1789.
5. Primary election turnout, which is the percent of the voting eligible population that participates in primary elections. This data is only available starting from 2000.

Each of the first four factors is something that can have an impact on individuals' political involvement and, as a result, can be determinative of election turnout. The last factor, primary election turnout, might not necessarily be determinative of turnout but can be predictive as primary voting is one of the first available indicators of political involvement in an election cycle. Our study will involve teaching our algorithm to understand the relationship between each of these factors and the subsequent general election voter turnout.

Much of the mainstream news coverage today centers around politics and, in the years leading up to a presidential election (which is almost always) coverage of the election cycle is heavy. For example, recent clips from hosts across MSNBC's lineup show their clear interest in analyzing election and campaign events and in forecasting the upcoming election.

CNN takes this a step further, as they currently include an "Election Center" on their website, where they attempt to break down everything relevant about the upcoming 2024 presidential election. This page includes the "Road to 270," which is their model-based prediction of how electoral college votes will be distributed between the presumptive party nominees, Donald Trump and Joe Biden, come election night. They also include a section that examines exit poll data they have collected from the primary elections that have been held so far and provides analysis of its significance towards the outcome of the general election in November.

It is evident that networks are interested in and collect primary data; we are recommending they take it a step further. The data available can be used to better understand and perhaps even predict election outcomes; forecasting election turnout would be a significant part of this endeavor.

Data Cleaning/Creating Our Dataset

Alexander & Fatima together

Once we acquired our data, we recognized immediately that our data was in very different formats, and would require a great deal of cleaning on our end. Our GDP data, for example, was in chronological order, but each economic quarter was in its own row. We required there to be a single column for every YEAR, and for each economic quarter to have its own column. It required us to build a new dataframe using years as our index, and individually attach the data for specific economic quarters using pandas.

Fixing our Primary election turnout data proved to be a difficult task as well. We had data for every fourth year, between 2000 and 2020, inclusive. But each year was placed in an individual dataset, and the information we were interested in(National primary turnout rate) had to be calculated by hand. For each spreadsheet, we need to find the sum of the ballots counted for every state's primary that year, and divide this number by the total number of eligible votes of this year. As a part of this process, we realized that even though they appeared as numbers to us, every number in these spreadsheets was a string object, sometimes with commas in between; after attempting to hard-code a solution, we decided the quickest way to deal with this was to slightly edit the format of each spreadsheet, and create a function that would allow us to remove commas, turn our strings into floats, and turn NAN values to 0, all in a single call.

Weapons of Math Destruction

“Weapons of math destruction” refers to potential consequences of mishandling our data; Should we be concerned if somebody with ill intent were to get their hands on it? The concerns of misuse should be saved for only the most extreme possible scenarios. At best, our model could show the value of certain economic or political factors for a successful presidential bid, and give a strong prediction as to who the next upcoming president will be. Many of those with ill intent will likely not have the influence to make large-scale economic changes in an effort to significantly alter the level of voter turnout. With that said, there is potential for an extreme case; if an enemy of the United States were able to get their hands on this data, a dependable prediction on who the next president might be could have disastrous consequences for the safety and security of the nation. Any number of economic sanctions, diplomatic moves, and counterintelligence initiatives could be vastly improved with that type of knowledge.

Linear Regression + Linear Regression with Train-Test Split

Primarily Alexander Ellis

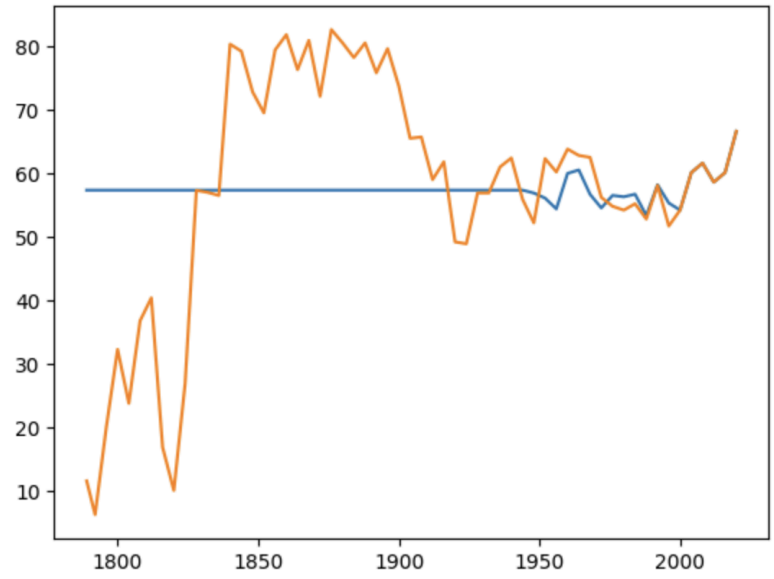
Linear Regression is a very common tool for predicting a trend in your data. We use a line to attempt to approximate a linear relationship between our dependent and independent variable(s). Our error is defined as the difference between our predictor (yhat) and that which is demonstrated by the data(y). The purpose of linear regression is to find the line that minimizes the sum of the square of the errors, or SSE.

For our purposes, our independent variables include the various forms of economic and political factors that make up our completed dataset. On the economic side, we have factors like GDP, which determines the difference between a country’s exports and imports, and the GINI coefficient, which measures a country’s level of economic inequality. On the political side, we’ve focused our research on the turnout of primary elections. Our dependent variable is voter turnout in presidential elections.

In our first attempt, we conduct a linear regression over the full dataset using statsmodels.api’s OLS regression, imputing all missing data values with 0. Below shows the actual turnout data compared to the predicted turnout data:

```
[64] predictions = model.predict(X_all)
     plt.plot(df_all["Year"].astype(float), predictions)
     plt.plot(df_all["Year"].astype(float), df_all["turnout"].astype(float))
```

```
[<matplotlib.lines.Line2D at 0x793173462ef0>]
```



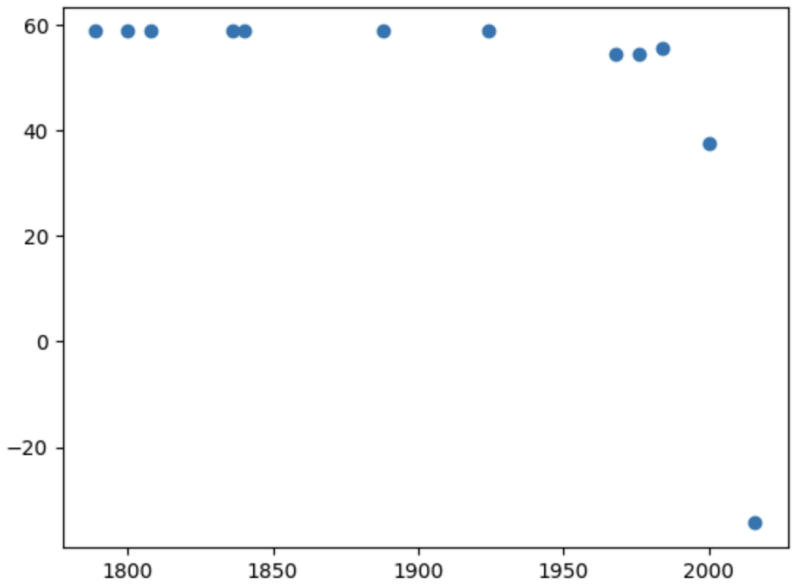
This graph is meant to depict our predictions for the percentage of voter turnout. On the x axis, we have years since 1789, and for our y we have the percent of the population that voted. There is a great deal of fluctuation early in our graph. This is because over time, more and more populations(White women, African Americans) were considered eligible to vote; this had a significant impact on what percentage voted. After 1960, we can see that our prediction begins to match a little more closely. We can see that it matches the rises and falls of the data, and predicts a steady incline in voter turnout for the future.

This model has an R^2 of 0.009, and it does not show statistical significance for any covariate. This indicates the model is of low quality, and that the covariates are not in fact predictive of turnout.

We also attempt to build a linear regression model with a 80-20 train test split. We do this to help build a more accurate model. The model is applied to the test data, and the output is shown below:

```
plt.scatter(X_all_test["Year"], predictions)
```

```
<matplotlib.collections.PathCollection at 0x7931723b93f0>
```



The R^2 of this model is 0.58, indicating it is a stronger model overall than the previous linear regression model. However, all covariates still show low statistical significance with the exception of the incumbent candidate indicator variable.

We are unable to perform cross-validation due to the significant amount of missing data.

Logistic Regression

Primarily Fatima Al-Sammak

Another way we can use the data available to understand voter turnout is by predicting whether turnout will be higher than average in a particular year. Given the changing voting laws over time, today's definition of "average turnout" is different from what it was in, for example, 1792. For this reason, we use the average over the period 1972 to 2020; this is because the electorate today is most comparable to the electorate during this period, since people aged 18-20 gained the right to vote in 1971 with the ratification of the 26th Amendment.[1] The average voter turnout since 1972 is 57.25%. We note that limiting the time period for this model reduces the number of datapoints available to us to 13, and we note that this is a weakness of our approach while emphasizing that we believe the policy justifications for this approach are important.

While we discussed logistic regression in class, we did not go over any programming demonstrations. To conduct this analysis and interpret the results, we reference content covered

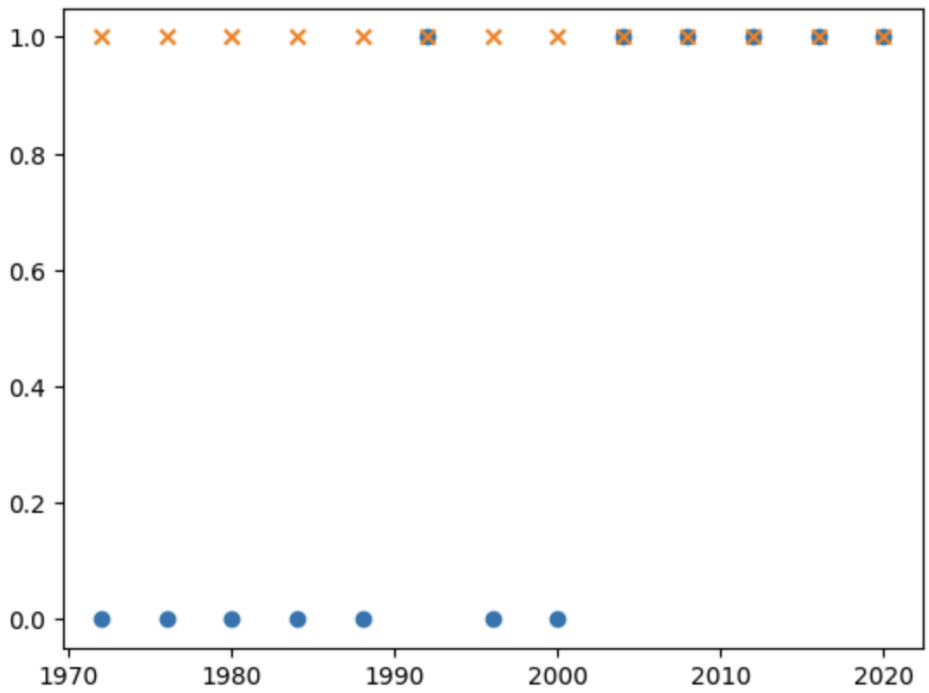
[1] https://www.reaganlibrary.gov/constitutional-amendments-amendment-26-voting-age-eighteen

in ORIE 3120. We construct a logistic regression model using statsmodels.api that examines each of the features of our dataset. Our first attempt at constructing this model drops all rows with missing values, so it is constructed using data from only 2012 to 2020. The model is then applied to all data since 1972 to check its accuracy, and at a threshold of p>0.5, it correctly predicts 6 out of 13 data points:

```
plt.scatter(predictions["Year"],predictions["actual"])
plt.scatter(predictions["Year"],predictions["classification"], marker = "x")

print(np.sum(predictions["classification"] == predictions["actual"]))
```

```
6
```



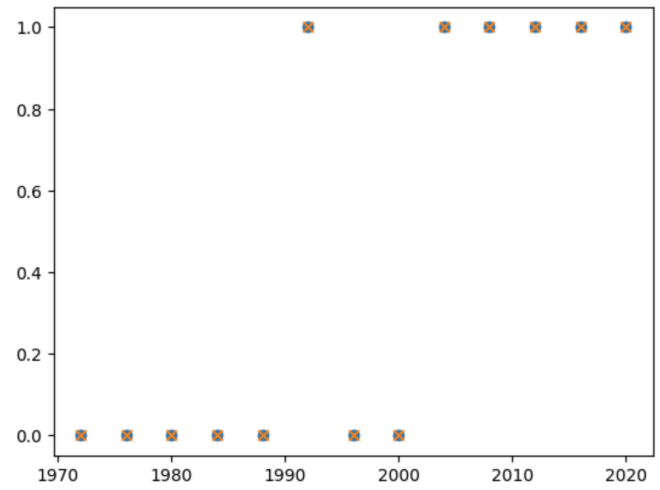
The pseudo-R^2 of this model is infinity, suggesting that the model is overfit. The p-values of the z-statistics of this model are all 1 and nan (infinity), indicating a low statistical significance. Overall, this is a poor model.

We try a new logistic regression model that inputs “0” for all missing values rather than dropping rows, so it is constructed using data from only 1972 to 2020. The model is then applied to our data to check its accuracy, and at a threshold of p>0.5, it correctly predicts 13 out of 13 data points:

```
plt.scatter(predictions["Year"],predictions["actual"])
plt.scatter(predictions["Year"],predictions["classification"], marker = "x")

print(np.sum(predictions["classification"] == predictions["actual"]))
```

```
13
```

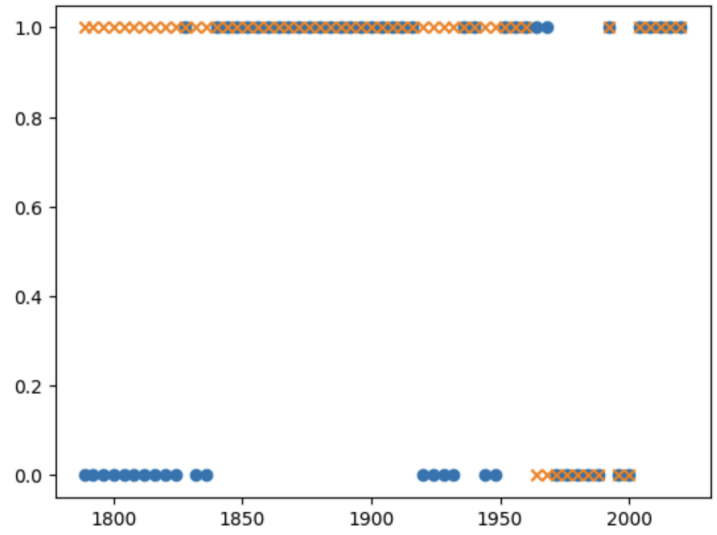


While the model more accurately predicts our data, the pseudo-R^2 of this model is also infinity, suggesting that this model is also overfit. The p-values of the z-statistics of this model are also all 1 and nan (infinity), indicating a low statistical significance. Overall, this is also a poor model, though it appears to perform better for the data from 1972 to 2020. Applying this model to the full dataset, we see that it correctly predicts 39 out of 59 data points:

```
plt.scatter(predictions["Year"],predictions["actual"])
plt.scatter(predictions["Year"],predictions["classification"], marker = "x")

print(np.sum(predictions["classification"] == predictions["actual"]))
```

```
39
```



There are two potential explanations for the low quality and predictiveness of these models:

1. *Massive gaps in the data available*: if this is the case, these models do have the potential to be informative once more data is collected.
2. *Suggested covariates are not predictive*: the data we have selected is not predictive and the models will be poor regardless of how much data we have available.

Regardless, as it stands, these models would not be particularly useful, and we would not recommend using them in industry.