CONGRESSIONAL VOTE CLUSTERING

Foundations of Data Science and Analytics (DSCI 608)

Greeshma Ganji
School of Information
Rochester Institute of Technology
gg1849@rit.edu
UID: 362007736

Le Nguyen
Department of Software Engineering
Rochester Institute of Technology
ln8378@g.rit.edu
UID: 383006341

Nandhini Lakshman
Department of Software Engineering
Rochester Institute of Technology
nl7222@g.rit.edu

Tolulope Olatunbosun
Department of Software Engineering
Rochester Institute of Technology
tao5634@g.rit.edu

Problem Statement

In this project, we wanted to see if the current political divide in American politics shows up in voting record data. Furthermore, we could build a model to classify which party a member of congress belonged to based on their voting record. Over the years, there have been many ideological differences between Republicans and Democrats. We suspected that these vast opinionated dissimilarities between the parties would be very apparent and show up as significant differences in voting records. We assume that the congressional data can be clustered into groups representing each party from this suspicion.

Research & Literature

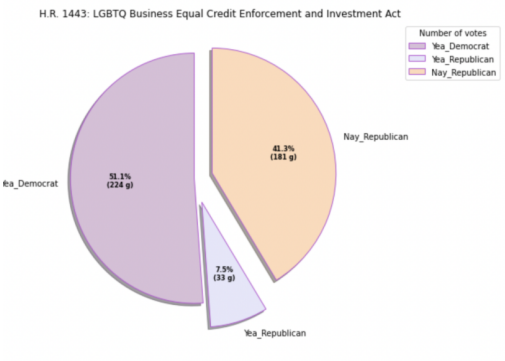
For this project, we looked at past analyses of this problem. Past work has primarily been done on clustering data from senators, while we clustered data from house members (GovTrack, 2021)(Swallow, 2017). We can see from historical data that the senate has been becoming more and more divided over time (Swallow, 2017). We suspect the house of representatives will follow the same pattern and be divided into distinct clusters along party lines. We can use some clustering method to do this.

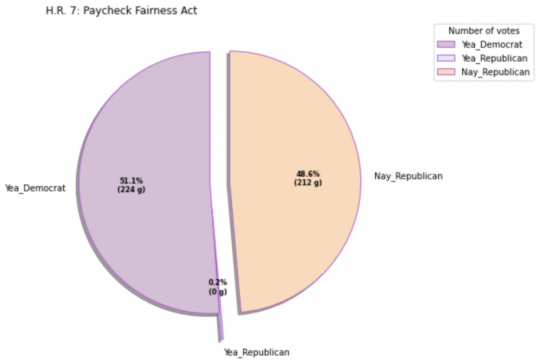
We had to find a proper model for this and further a distance metric to define distances between congresspeople and the cluster. We found the Jaccard distance to be the most suitable distance metric to use as it is the distance between two sets (the sets of voting records) (Glen, 2016). Also, through our research, we found that the most appropriate clustering method to use was hierarchical clustering (Sharma, 2019).

Analysis

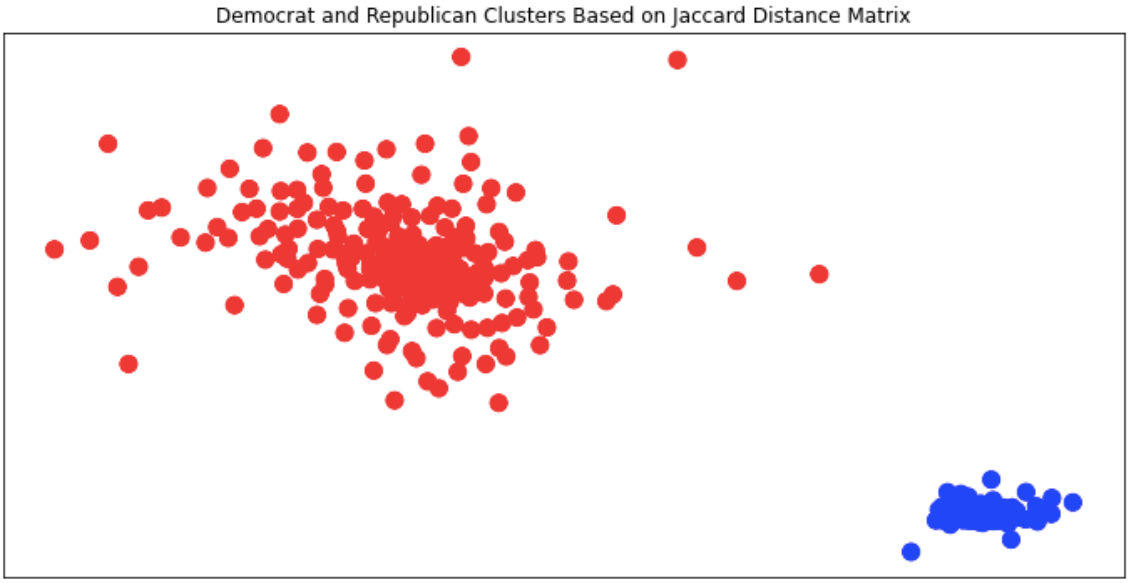
Two lists were created for the number of times a Democrat and a Republican voted against their party. The mean count was computed for each list to get a better picture of the average number of outliers. From the results, we observed a higher number of Republicans who voted against their party compared to the number of democrats. We wanted to visualize the outliers in pie charts for certain bills.

Upon visualization, we noticed a significantly higher number of Republicans deviating from typical Republican voting behavior for certain bills. The portions of the pie charts were much smaller for the Democrats who voted against their parties.
Generally, the parties seemed to be very divided in their voting behavior.





After computing the Jaccard distance, we discovered relational distances between each politician with respect to their voting records. Smaller distances demonstrated similarity between voting choices across various bills as well as similar beliefs, being members of similar parties, or perhaps collusion (Scipy, 2021). On the other hand, greater distance metrics represent distinct differences in voting preferences. The figure below illustrates each senator concerning other senators. We found that members of the Democratic party generally vote with their party, and often with little variance. Contrarily, Republican senators represented varied voting preferences, though distinct enough to stay within their respective clusters.



Working Plan

Le - Web scraping and cleaning the data for analysis and model development. Did some data analysis on the side as well.
Greeshma - Feature encoding of the clean data and transformed the categorical data into numerical ones for the model to understand and check whether any congressmen voted against their party.
Nandhini - Performed data visualization and data analysis on controversial bills to inform voting preferences and to detect anomalies in either party.
Tolu - Performed data analysis and implemented Jaccard distance metric on the data to map preferences between politicians' voting behaviors.

Teamwork & Collaboration

In terms of teamwork and collaboration, we did struggle a bit to have consistent meetings because we were all busy with our classes and had conflicting schedules. After the first few meetings to set up the project idea, we did not meet again for about a month. We had to start frequent meetings again when the project due date came near. This was a problem when giving weekly updates because we were not consistently meeting to generate new work and have everyone else informed. Everything did come together in the end, though, and worked out.

In hindsight, having more frequent meetings would have made the project a more comfortable process, but, understandably, we got caught up in other work and classes. Perhaps having online meetings would have served us better with our busy schedules.

Individual Contribution

The raw data is obtained from the GovTrack website, which keeps track of every bill the congressmen vote in the senate. It provides details of the bills such as name, party, state of each senator, and which senator voted 'Yea' or 'Nay' to which bill, according to the bill number. 'Yea' means that they voted in favor of the bill, and 'Nay' means they voted against it. Using Selenium WebDriver, we can web scrape the data and retrieve the required data. After the data was web scraped, we checked if there was any missing data or ragged data. Not every congressperson voted on every bill. Everyone doesn't need to vote on all bills, so we found a few missing data, and then it was filled in with relevant value depending on which party the congressman belonged to. We also had a challenge regarding keywords of states which had two words. This process was done by my teammate Le Nguyen.

As the data is filled in, the next step is feature encoding. The data is still in the text format of 'Yea' and 'Nay' divided into two categories. This is considered categorical data. To visualize and analyze the data, we build the machine learning model. For the model to understand the data, it should be in numerical format, so we use the process of feature encoding. Feature encoding is the process of converting categorical data into numerical data. There are many encoding processes such as label encoding, One Hot encoding, Frequency One Hot encoding, target mean encoding, binary encoding, count encoding, feature hashing, etc.; it is better to use Label Encoder as our data contains only two values 'Yea' and 'Nay'. Label encoder is used to normalize the data. The function LabelEncoder () is used for preprocessing the data, and then function fit () encodes the data and transform () function labels to normalized encoding. The 'Yea' and 'Nay' in the bills are transformed into '1's and '0's. We first thought of using one hot encoding and then used a label encoder because it was sufficient for our data. We also want to see from the data which party congressman went against their party and voted on bills; by using mode function, the frequency of people who voted against a bill is calculated. This is calculated for each bill and then appended in repAgainst () function for republicans who voted against republicans and appended in function demAgainst () for democrats who voted against democrats. After that mean is calculated which gives the average of the congressmen willing to go against their parties. It is observed that the republicans have a higher mean value than democrats. By analyzing this, we can tell that the democrats usually vote similarly and differ less in voting than republicans. The republicans tend to go against their party to vote for a certain bill. As the data is now in the format of senators' details and which bill they voted for, it would be easier if we transpose the data to be analyzed properly. For the visualization part, we don't require party and state columns. So, we are dropping the columns 'State' and 'Party'. Using the transpose function, the data is transposed. We have each bill and which senator voted 'Yea' and 'Nay' for that bill represented in '1's and '0's.

The Data visualization and analysis part is done by Nandhini and Tolu for each bill and checked for anomalies. Further, the Jaccard distance is calculated and appended in Jaccard Matrix. After that, the clusters are plotted based on the Jaccard distance matrix, which is done by Tolu. The model fitting and hierarchical clustering and dendrogram part are done by Le Nguyen.

Works Cited

1. GovTrack. (2021). *Voting Records*. Retrieved from govtrack.us: https://www.govtrack.us/congress/votes

2. Swallow, E. (2017, November 17). *U.S. Senate More Divided Than Ever Data Shows*. Retrieved from Forbes: https://www.forbes.com/sites/ericaswallow/2013/11/17/senate-voting-relationships-data/?sh=4768ba54031d

3. P. Sharma, "A Beginner's Guide to Hierarchical Clustering and how to Perform it in Python," 27 May 2019. [Online]. Available: https://www.analyticsvidhya.com/blog/2019/05/beginners-guide-hierarchical-clustering/

4. Scikit-Learn Machine Learning in Python, Pedregosa, *et al.*, JMLR 12, pp. 2825-2830, 2011. https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.html

5. pceccon. (2016, March 31). *Generating graph from distance matrix using networkx: inconsistency - Python*. Retrieved from StackOverFlow.com: https://stackoverflow.com/questions/36339865/generating-graph-from-distance-matrix-using-networkx-inconsistency-python

6. S. Glen, "Jaccard Index / Similarity Coefficient," 2 December 2016. [Online]. Available: https://www.statisticshowto.com/jaccard-index/

7. *Scipy.spatial.distance.jaccard*¶. scipy.spatial.distance.jaccard - SciPy v1.7.1 Manual. (n.d.). Retrieved December 2021, from https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.distance.jaccard.html