Introduction:

I started this project out hoping to give an in-depth analysis of internet chatter surrounding the 2018 Brazilian Presidential election. As a Brazilian citizen who is extremely interested in data surrounding online presence and usage, I could not help but wonder if there were particular elements that were predictive of election results. In 2018, the current Brazilian President Jair Bolsonaro was infamously elected after unprecedented level of pro-Bolsonaro political propaganda was spread WhatsApp. That same year Brazil had over 108 million users on WhatsApp[[1]](#footnote-1), the third largest amount out of any of other country in the world.

Given Brazil’s large online population, I decided to investigate if online search trends were at all predictive of the election winner.

Aim:

* To discover what are the predictive variables of winning a presidential election
* Particularly focusing on search terms on Google and their popularity

What is Covered in the Report:

Problem Statement and Background:

I intended on using online chatter in order to predict if a certain candidates’ ability to win based on the amount of chatter surrounding their candidacy. However, after submitting my proposal in which I outlined how I would pull data from Twitter and Google, I realized that I did not have the ability to pull on Twitter without getting through a paywall and so quickly dropped the twitter component of the project. After spending a fair amount of time on Google Trends, I came to the conclusion that I could not simply just analyze how much each candidate was searched for as a measure prediction given that I would have too little data. With this in mind, I decided I would shift my project’s gears into seeing if certain keywords were predictive of a certain candidate winning.

Having realized that I would focus my project on Google and Google Trends, I started compiling a variety of different terms I believe Brazilians would be searching for in preparation for a Presidential election. I then planned on using these search terms as a predictive factor in the Brazilian Presidential Election. After asking multiple friends and family, I came up with a list of top 10 terms:

* Economia – economy
* Corrupção – corruption
* Previdência – Retirement Benefits
* Plano Governo – government plan
* Responsability Social – social responsibility
* Desemprego – unemployment
* Bolsa Familia – Brazil’s top welfare program
* Currículo – curriculum
* Pobreza – poverty
* Inflação – inflation

I intended to use 10 items as I planned on seeing overall mentions of each item and comparing them. In doing so, however, I ran into my first major problems with the project. The first was the fact that Google does not give raw numbers of how many times a term was used in a search but gives you relative terms instead. The second is that Google also caps the number of terms you can pull a relative search for at five and as such I would not be able to use all the search terms I had planned to. Issue number one is a recurring one that has multiple implications on the results of this paper and shall be addressed as we go along. However, issue number two was easily solved by putting them all in the search alternating words until I got both a diverse group of words that served to pull insights on different topics that would influence voters as well as terms that would have enough data for the analysis to be significant. I finally ended up with my five search terms which are the basis for the research conducted in this project:

* Bolsa Família
* Desemprego
* Economia
* Plano Governo
* Previdência

After narrowing down my search terms, I decided that I would try to pull data from the two weeks prior to each election round, however in doing so the software I was using *pytrends* would pull each day as an individual query and since the searches are relative, the levels for each search term would no longer be comparable and the predicted model not only flawed but also likely of very low predictive value. As such, I pulled the trends my month in using the *‘all’* timeframe while building my payload in pytrends, and after pulling interest over time, filtered out for only the searches done in the month of the election (October, which was pulled on November 1st), which I would compare against the search done in the month after the election (November, which was pulled on December 1st).

Since Google Trends allows us to pull data starting in 2004, I pulled the relevant keywords searches in four election years: 2006, 2010, 2014 and 2018. After doing so, it was clear to me that my research would require another level to it and as such I decided to not only pull the keyword search on a national level, but also pull them for the most populous state of each of the five Brazilian Regions[[2]](#footnote-2):

* North: Amazonas
* Northeast: Bahia
* Midwest: Mato Grosso Do Sul
* Southeast: São Paulo
* South: Rio Grande do Sul

Finally, when it came to the predictive model for this project, I realized that it would be hard to a winner by party because Brazil has over 30 parties[[3]](#footnote-3) and there are not two leading parties as there are in the United States. Yet, in the elections that I was analyzing, the Labor Party’s candidate made it to the final round ever time and as such I decided to create a Dummy Variable using the winner being from the Labor Party in each state as 1 and the winner being in any other party as 0.

Data:

All the data used in this project, apart from the word-of-mouth research I did to get search terms Brazilians would search for in the eve of each election, came from pytrends, the unofficial API for Google Trends.[[4]](#footnote-4) This Application programming interface (API) allowed me to pull results tracked by Google Trends in python, without having to manually scrape the Google Trends website. While I initially did struggle with pytrends in order to standardize my results, I was able to ensure standardized data through the use of “timeframe=‘all’” when I built my payload.

The unit of observation of this project was the relative number of times that certain keywords were searched in the month before and the month after the election, in Brazil and each of the most populous states of each of the 5 Brazilian regions.

Since Brazil does not have a two-party system in order for me to get the outcome of interest i.e. who would win, I created a dummy variable which accounted for the Labor party winning in that particular region, given that it is the only party that was in the final round of every election since 2006.

#final

For the main part of my project, I used both the states that we are analyzing as well as the keywords as predictive variables in order to see if there was any predictive value in the results. After doing so I ran the model only using the keywords as the predictive variables.

There were a couple of issues in the data. While the missingness matrix did not show any data missing, there were a couple of search terms whose value in a certain state at a certain time was zero. Additionally, since the score we get for each search term is relative (i.e. a 50 doesn’t mean it was searched 50 times but rather has a score of 50 relative to a time in which there was the most searches and that would be 100), if this experiment was to become an actual model of use, there would need to be further manipulation of the data.

#miss.matrix(final)

Analysis:

As previously mentioned in the data section, all of my data came from the pytrends API, however I did perform a variety of changes to the data in order to results:

Step-by-step of what I did:

1. First, I built out my payload which was the same for all the results I ran except for the geo which I changed each time
2. After getting the data for all the months since the API starts (2004) I narrowed it down solely to the months of interest i.e. the month of the election October (by retrieving data from November 1st) and December (retrieving data from December 1st)
3. After doing so for each geo – i.e. all the regions of interest, I melted each of the result tables I had in order from go from wide to long, so that I could draw graphs for Keyword Popularity in each of the 6 regions.
4. Next, I drew line graphs to show the popularity of each search term in each region, which was the easiest way to visualize the trends. I expected that there would be a sharp decrease between the October searches to the November searches and that was the case for the most part.
5. I then started working on my prediction model. I visualized all the distributions for the variable for the training data in order to see how my data was distributed.
6. After doing so, I preprocessed my data into the pipeline since we are only using information from the training data.
7. Next, I started to run machine learning models in order to get the best model, score and parameters for my data. I gaged its performance and finally I interpreted the importance of each variable in the after starting a permutation.
8. After seeing the first model I just ran the predictions for a model without the States as variables and got a different predictive value

1. https://www.businessofapps.com/data/whatsapp-statistics/ [↑](#footnote-ref-1)
2. https://www.gov.br/pt-br/noticias/financas-impostos-e-gestao-publica/2021/08/populacao-brasileira-chega-a-213-3-milhoes-de-habitantes-estima-ibge [↑](#footnote-ref-2)
3. https://www.tse.jus.br/partidos/partidos-registrados-no-tse [↑](#footnote-ref-3)
4. https://pypi.org/project/pytrends/#interest-over-time [↑](#footnote-ref-4)