CPSC 324: Big Data Analytics

Final Report

Spring 2024

Project Description

- US Presidential Election Data Analysis
- Using data from X (formerly Twitter)
- Create a model to analyze the sentiment toward two candidates
- Applied on the most current data to see if there has been a shift in voters' sentiment for two candidates
- Hypothetically correlates with the possibility of winning the election

Background

- At the time this report is created, Joe Biden and Donald Trump are the two presumptive candidates for the 2024 presidential election.
- The twos were also the candidates back in 2020, when Biden defeated Trump.
- Between 2020 and 2024, there have been many events that might alter people’s opinions about the two candidates. With that, it should be interesting to see how things are affected.

Background (cont)

- X (formerly Twitter)
- Their API policies…much stricter
- Use Kaggle to find the alternative dataset
  - https://www.kaggle.com/datasets/manchunhui/us-election-2020-tweets

The Datasets

- Two smaller datasets
  - One for Trump, and one for Biden
- The Biden file
  - Around 775,000 instances
  - 20 attributes.
- The Trump file
  - Around 958,500 instances.
  - 20 attributes
- Considered to be a Big Data dataset

Dataset Challenges

- Inconsistent format, especially the newline characters
- Tweet contents vary a lot
  - Mixed use of hashtags and mentions
  - Unrecognize characters
  - Different languages

Approach/Services Used

- Preprocess locally
- Use BigQuery to perform some exploratory data analysis
- Use Google Translate API to recognize and translate non-English content
- Use Vertex AI's AutoML to train the data
- Use Cloud Composer to create a full pipeline after that
  - Receive incoming data
  - Perform cleaning and create TF-IDF data (later)
  - Pass it through the model

Google Translate

- Pretty easy to use
- Setting up takes a bit of time
- Expensive: runtime is not fast and is costly

TF-IDF

- A statistical measure used to evaluate how important a word is to a document
  - Term Frequency (TF): Measures how frequently a term occurs in a document. TF is the number of times a term appears in a document divided by the total number of terms in that document.
  - Inverse Document Frequency (IDF): Measures the importance of the term across a set of documents. It is calculated as the logarithm of the number of documents divided by the number of documents that contain the word.

Why TF-IDF?

- **Relevance**: Weigh the importance of a term
- **Specificity**: Filter out less important terms, focusing on words that distinguish documents from each other.
- **Feature Scaling**: By converting text data into numerical scores, TF-IDF facilitates the use of mathematical models and algorithms that work with quantitative data
- **Improved Model Performance**: Make it faster

Vertex AI

- A bit confusing at first but gets pretty straightforward after understanding what to do
- AutoML makes it very efficient to run these training
- Take quite a bit of time to run
  - Around 5 hours per model
- Take care of other housekeeping duties:
  - Training and testing set splitting

Model Evaluation

- Pretty impressive
- Labeled into three categories (aka doing Classification)

	The Biden Model	The Trump Model
Precision-Recall Curve	0.898	0.91
Receiver Operating Characteristic	0.942	0.947
Log Loss	0.461	0.448
Micro-Average F1	0.8570497	0.8521643
Macro-Average F1	0.78704923	0.8100085
Micro-average precision	87.6%	87.1%
Micro-average recall	83.9%	83.4%

What’s next…?

- Fetch Data (streaming ideally) from X into the model to generate the prediction
- Implement a sentiment score to weigh how much each category (positive, negative, and neutral) a post earn
- Implement a more complicated scoring method to weigh the result of the prediction

What can be done better if time/financial allows?

- Streaming Data to complete the entire pipeline
- Cloud Composer implementation to complete the pipeline fully
- Additional training using Vertex AI on top of the current models

What have I learned?

- Training takes more time than I thought, compared to 322
- With a big data set, any decision made is expensive
  - Time and money
- X is no longer friendly with developers like Twitter used to be
  - Consider moving to Meta?
  - Or higher budget to pay for API access