

Tony Nguyen

Dr. Shawn Bowers

CPSC 324 01

22 April 2021

Project Checkin

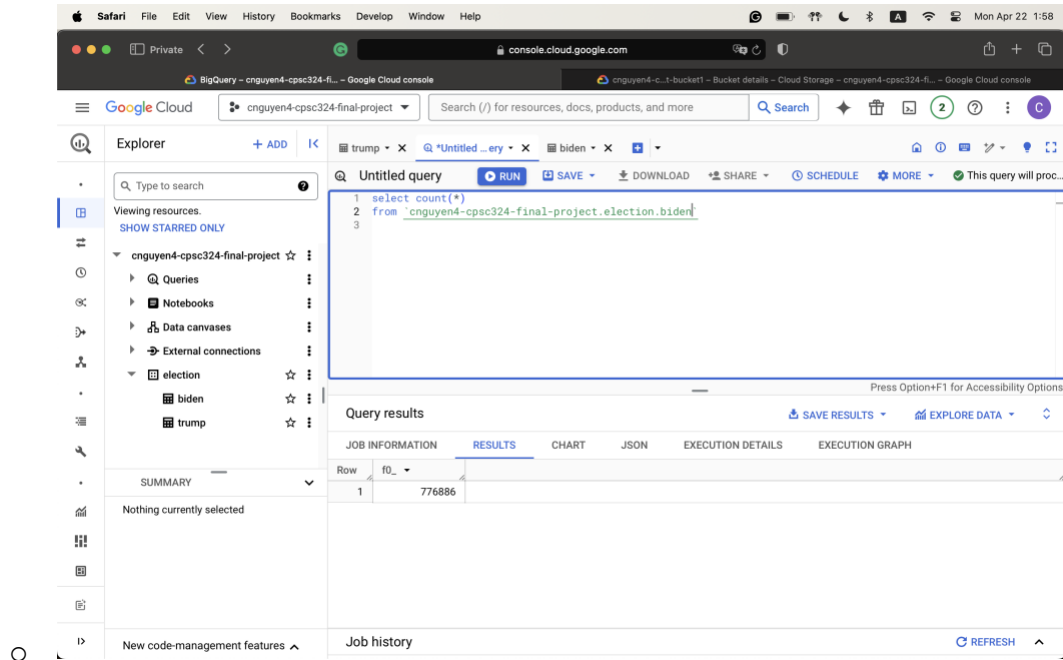
I am working on the Presidential Election X (formerly Twitter) Analysis, which aims to train a model that analyzes the sentiment towards the two presumptive candidates for the presidential election later this year (2024) – Trump and Biden. This model is trained based on two datasets that include hashtags #Trump, #DonaldTrump, #Biden, and #JoeBiden during the 2020 election. The trained model will then be used against two similar but more current datasets to see if there is a shift compared to the 2020 version, which will hopefully see if it can predict the 2024 winner.

So far, I have been able to clean the dataset and load it to Cloud Storage, which is then run in BigQuery. I initially thought that this process should be quick; however, I got some hiccups while doing so. The biggest problem I have, besides the size of the file, is the post contents. As the contents are user-created and usually not in any inherent format or language, I ended up with some weird characters, inconsistent quotation marks (that makes BigQuery think it is creating several fields instead of just one post), inconsistent newline notation, and more. This issue came up as soon as I attempted to load it into BigQuery when it rejected the job after going through around 450 rows (0.04%-ish of the file).

I ended up taking a much smaller subset to test it first with pandas and performing cleaning. From there, I ran the script with the two larger files. I was able to get the script to run locally without having to create a virtual machine instance on Google Cloud. During the cleaning

process, I applied filterings to replace the quoting inconsistencies and the inconsistent newline notation. Then, I exported it to a JSON format to prevent BigQuery from being misread.

- Biden's file



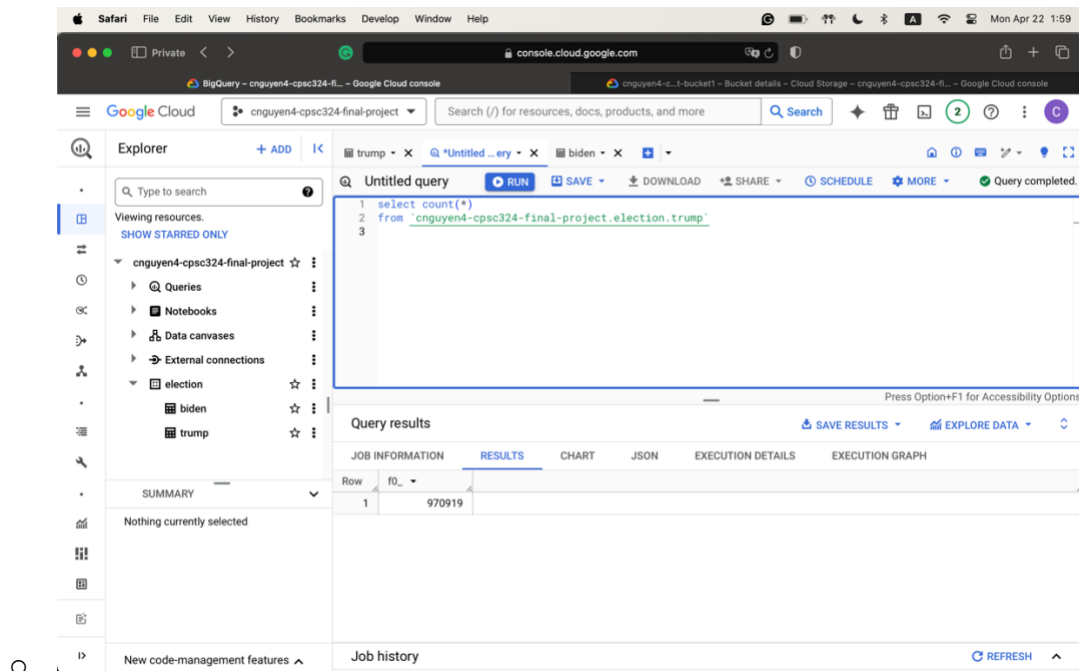
The screenshot shows the Google Cloud BigQuery console interface. The left sidebar displays the project hierarchy: 'cnguyen4-cpsc324-final-project' > 'election' > 'biden'. The main query editor contains the following SQL code:

```
1 select count(*)
2 from `cnguyen4-cpsc324-final-project.election.biden`
3
```

The 'Query results' section shows a single row with the count 776886.

Row	count(*)
1	776886

- Trump's file



The screenshot shows the Google Cloud BigQuery console interface. The left sidebar displays the project hierarchy: 'cnguyen4-cpsc324-final-project' > 'election' > 'trump'. The main query editor contains the following SQL code:

```
1 select count(*)
2 from `cnguyen4-cpsc324-final-project.election.trump`
3
```

The 'Query results' section shows a single row with the count 970919.

Row	count(*)
1	970919

From there, I did some basic exploratory data analysis to understand more about the data. Among those, here are some of the findings

- There are around 300,000 unique users in the Trump file, on top of the total 970,000 posts, meaning, on average, each person makes three posts.

```

1 select count(distinct user_id)
2 from `cnguyen4-cpsc324-final-project.election.trump`

```

Query results

JOB INFORMATION		RESULTS	CHART	JSON	EXECUTION
Row	f0_				
1		301358			

- The number is around the same for Biden.

```

1 select count(distinct user_id)
2 from `cnguyen4-cpsc324-final-project.election.biden`

```

Query results

JOB INFORMATION		RESULTS	CHART	JSON	EXECUTION
Row	f0_				
1		316053			

- There are around 450,000 posts that are similar to each other among the two files.

```

1 select count(*)
2 from `cnguyen4-cpsc324-final-project.election.trump` join `cnguyen4-cpsc324-final-project.election.biden`
   using (tweet_id)

```

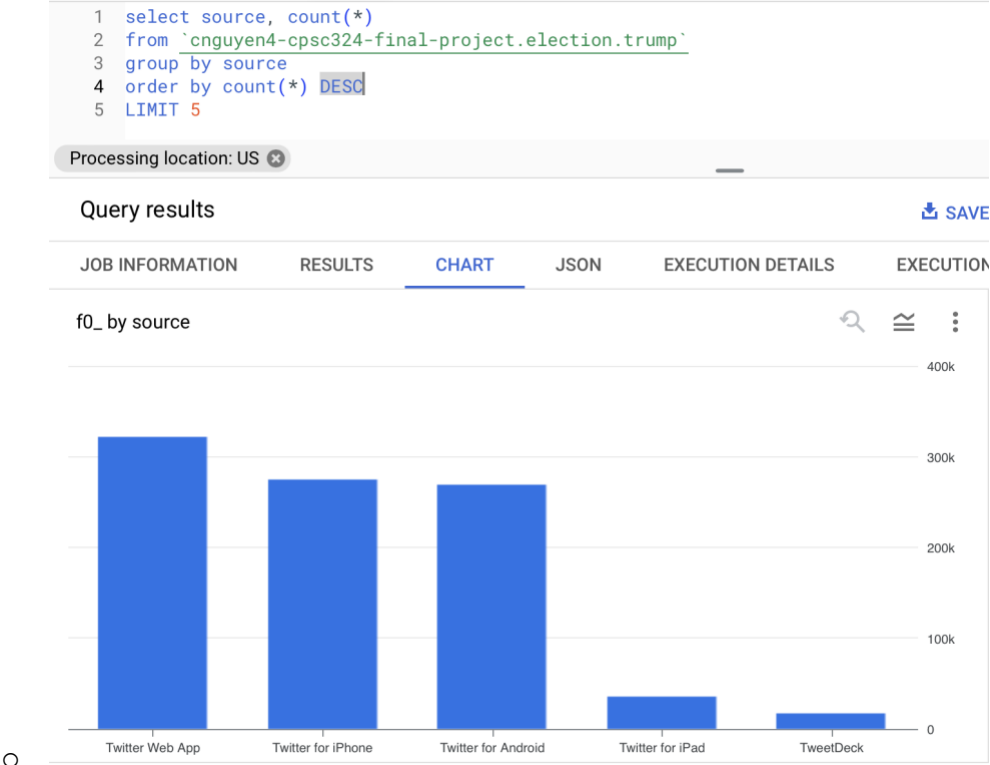
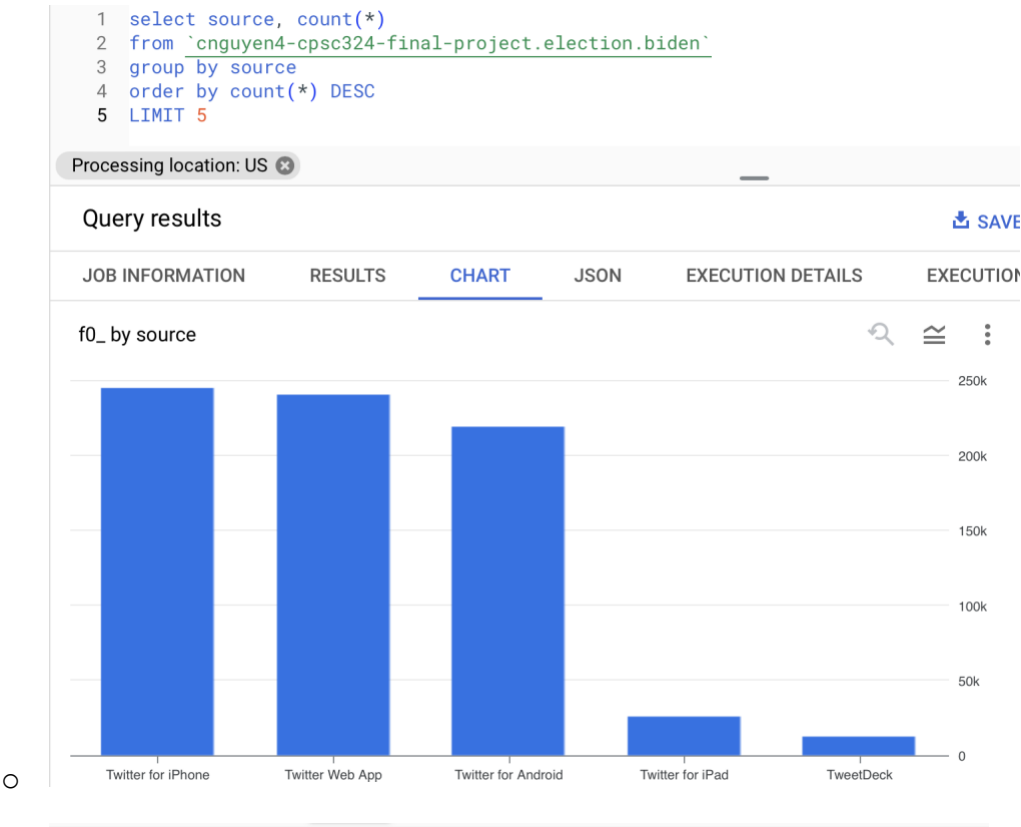
Query results

[SAVE RESULTS](#) [EXPLORE DATA](#)

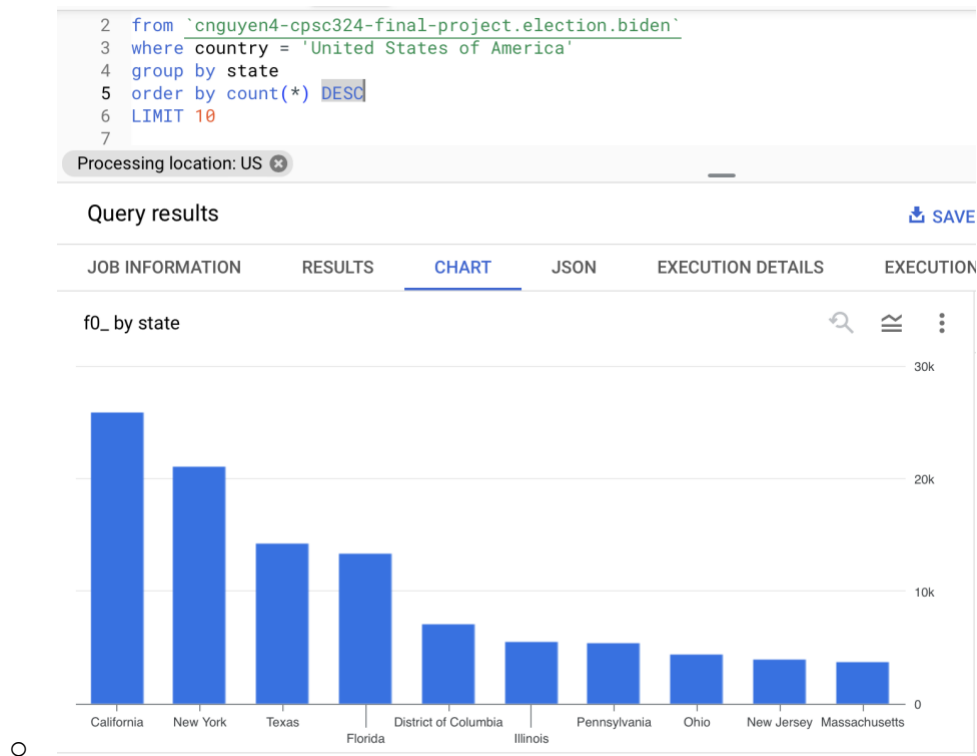
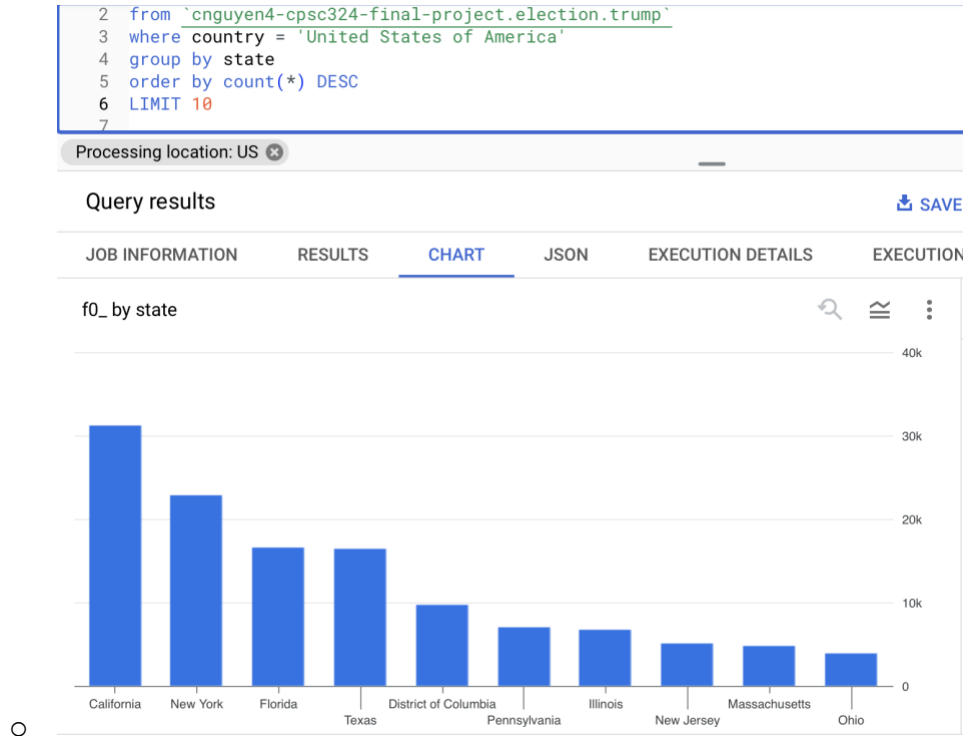
JOB INFORMATION		RESULTS	CHART	JSON	EXECUTION DETAILS	EXECUTION GRAPH
Row	f0_					
1		459011				

- Among those, around 223,000 posts originate from the same person.

- Percentage of Posts by Source



- Number of tweets per state



- The top states are similar. Pennsylvania, which is a swing state, is also on the list.

From here, I will use the Natural Language API to perform the sentiment analysis and then use Vertex AI to train the model based on that result. This will be on my to-do list in the coming time. However, I think the remaining tasks should be faster, as cleaning usually takes up the most time while completing a project. Besides, I have spent some time researching how to use Google's Natural Language API, which I think will be helpful in the future.