POTATO Take-Home Task

Overview:

POTATO (the Panel-based Open Term-level Aggregate Twitter Observatory) is a prototype website that uses data from the Lazer Lab's Twitter Panel. The Twitter Panel links over one million real U.S. voters to their Twitter accounts, and has each panelist's tweets from about 2016 until 2023 or so. POTATO will allow users to search for a term ("COVID") and get **aggregate** information about the people who tweeted about the term. We will threshold results such that any demographic bucket with fewer than ten users will not be shown; we're also considering using statistical processes to add noise without disrupting the overall distribution of the data. Right now the system uses Docker, Elasticsearch, Streamlit, and Python. Our biggest technical problems are a) ingesting the data from HDFS efficiently and b) returning results quickly. We also need to look into strengthening our privacy protections.

Task:

In this Google Drive folder, you'll find two TSV files of tweets about Britney Spears. One is ~50MB and the other is ~500MB. While I'd prefer you use the larger file, please feel free to use the smaller one if your computer can't handle it. The point of this exercise is not doing everything to the letter. I want to see how well you can do an open-ended task and how effectively you write and explain code. Please feel free to email me at which is assigned specifically for assessment. harsh.p@silverspaceinc.com if you have questions about the assignment, but understand that this is left as an open-ended exercise for a reason.

Part 1

Ingest the data: figure out a way to put the data in a structure so that you can query it as described in Part 2.

Part 2

Construct functionality that allows you to query the data. If I search for a term, like "music," I would like to know some subset of the following:

- How many tweets were posted containing the term on each day?
- How many unique users posted a tweet containing the term?
- How many likes did tweets containing the term get, on average?
- Where (in terms of place IDs) did the tweets come from?
- What times of day were the tweets posted at?
- Which user posted the most tweets containing the term?

Part 3

Explain to me how I can use your system. I should be able to run this system and query it on my own computer using your instructions. Please explain and justify any important design choices you make.

Part 4

Send me a link to the Github repo that contains your system; you can email me with a link or add me (itsmeblackops is my Github username).

Bells and Whistles

I would be really happy to see any of the following:

- Usage of Docker
- Usage of a NoSQL database
- Construction of an API, with proper documentation (consider using Flask for this)
- Queries that return results quickly (and/or optimizations to make queries return quickly on average)
- Thorough documentation
- Well-commented code
- Tests (consider using pytest/mock/similar packages)