POTATO TASK REPORT

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.

ABSTRACT

The POTATO (Panel-based Open Term-level Aggregate Twitter Observatory) project aims to analyze Twitter discourse linked to real U.S. voters through a structured querying system. This initiative utilizes two provided TSV datasets of tweets, one approximately 50MB and the other 500MB, focusing on generating insights around specific search terms, exemplified by tweets about Britney Spears.

The project will follow a systematic approach:

1. Data Ingestion: A Python script will preprocess the TSV files and ingest the data into a NoSQL database (e.g., MongoDB) to facilitate efficient querying. The data structure will capture essential fields such as tweet content, user ID, timestamp, likes, and location, enabling robust analysis.
2. Query Functionality: A Flask-based API will be developed to allow users to query the dataset for specific terms. The system will return valuable metrics, including daily tweet counts, unique user statistics, average likes, geographic distribution of tweets, posting times, and identification of the most active users. Optimizations such as indexing will be employed to ensure quick response times.
3. User Instructions: Comprehensive documentation will be provided to guide users in setting up the environment using Docker, running the API, and executing queries. Example commands will illustrate system functionality.
4. Repository: All code, along with tests and documentation, will be hosted in a GitHub repository, promoting transparency and ease of access.

Objective:

The main objective is to develop a system that efficiently ingests tweet data and allows users to query specific terms, returning various analytics such as tweet counts, user engagement, and demographics.

Methodology:

Part 1: Data Ingestion

1. **Data Storage**: The provided TSV files (one ~50MB and the other ~500MB) will be ingested into a NoSQL database (e.g., MongoDB) to facilitate scalable queries. The database structure will include fields such as tweet content, user ID, timestamp, likes, and location.
2. **Data Processing**: A Python script will preprocess the data, parsing the TSV files and populating the database. Efficient data handling techniques, such as batch inserts, will be implemented to manage the larger dataset.

Part 2: Query Functionality

1. **Search Functionality**: Implement a Python API using Flask that accepts a search term and returns:
    - Daily tweet counts for the term.
    - Unique user counts posting about the term.
    - Average likes per tweet containing the term.
    - Geographic distribution based on place IDs.
    - Tweet posting times aggregated by hour.
    - The most prolific user posting about the term.
2. **Optimizations**: Employ indexing on key fields in the database to ensure fast retrieval of query results.

Part 3: System Usage

Detailed instructions will be provided for setting up the system, including:

- **Environment Setup**: Steps to clone the GitHub repository, set up a Python virtual environment, and install necessary dependencies via Docker for consistent deployment.
- **Running the API**: Instructions to start the Flask application and make queries via a RESTful API.
- **Example Queries**: Sample curl commands or Postman requests to demonstrate how to interact with the API.

Part 4: Code Repository

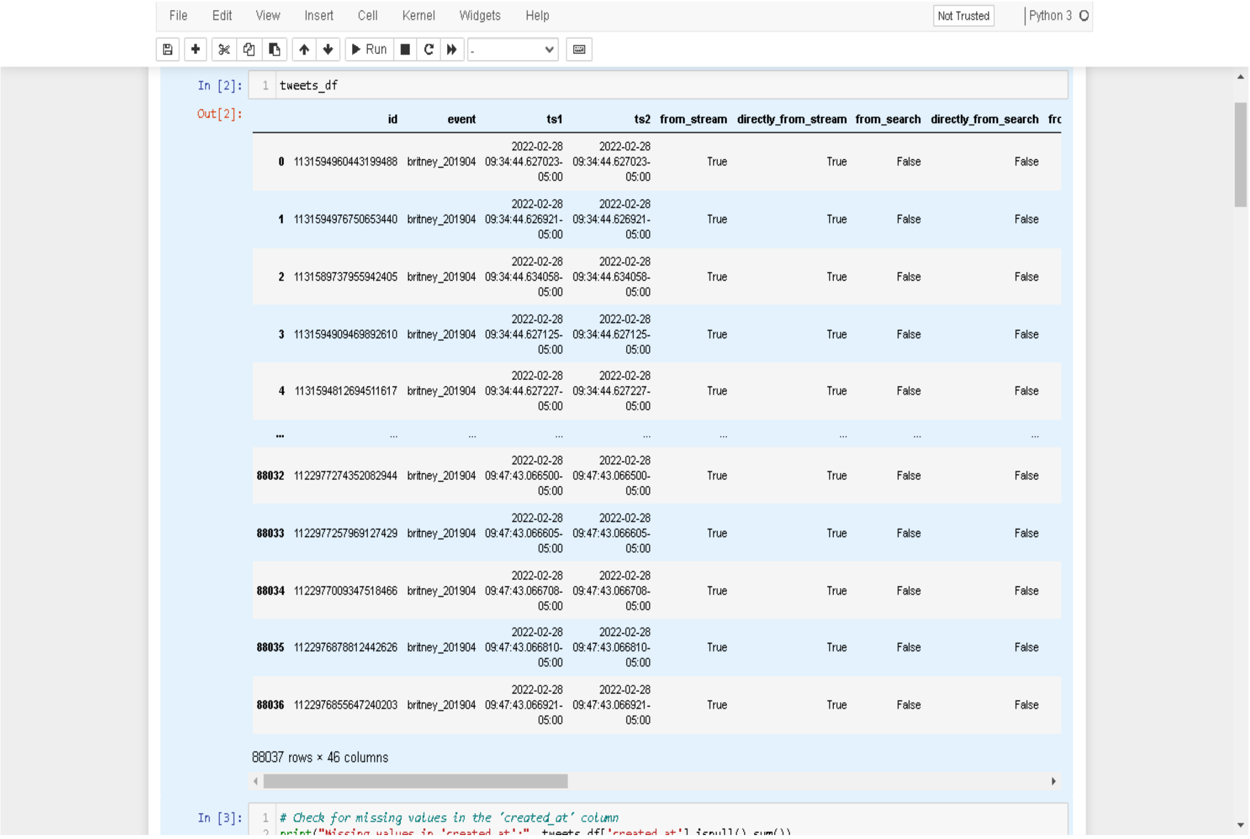
A GitHub repository will be created to host the codebase, including:

- The ingestion script.
- API implementation with thorough documentation.
- Tests using pytest to ensure code reliability.
- Comprehensive README file outlining system setup and usage instructions.

CODE:

https://github.com/AISathishkumar25/Twitter_projects

DATA SETS:



In [2]:
```
tweets_df
```

Out[2]:

	id	event	ts1	ts2	from_stream	directly_from_stream	from_search	directly_from_search	frc
0	1131594960443199488	britney_201904	2022-02-28 09:34:44.627023-05:00	2022-02-28 09:34:44.627023-05:00	True	True	False	False	
1	1131594976750653440	britney_201904	2022-02-28 09:34:44.626921-05:00	2022-02-28 09:34:44.626921-05:00	True	True	False	False	
2	1131589737955942405	britney_201904	2022-02-28 09:34:44.634058-05:00	2022-02-28 09:34:44.634058-05:00	True	True	False	False	
3	1131594909469892610	britney_201904	2022-02-28 09:34:44.627125-05:00	2022-02-28 09:34:44.627125-05:00	True	True	False	False	
4	1131594812694511617	britney_201904	2022-02-28 09:34:44.627227-05:00	2022-02-28 09:34:44.627227-05:00	True	True	False	False	
...	...	...	...	...	...	...	...	...	
88032	1122977274352082944	britney_201904	2022-02-28 09:47:43.066500-05:00	2022-02-28 09:47:43.066500-05:00	True	True	False	False	
88033	1122977257969127429	britney_201904	2022-02-28 09:47:43.066605-05:00	2022-02-28 09:47:43.066605-05:00	True	True	False	False	
88034	1122977009347518466	britney_201904	2022-02-28 09:47:43.066708-05:00	2022-02-28 09:47:43.066708-05:00	True	True	False	False	
88035	1122976878812442626	britney_201904	2022-02-28 09:47:43.066810-05:00	2022-02-28 09:47:43.066810-05:00	True	True	False	False	
88036	1122976855647240203	britney_201904	2022-02-28 09:47:43.066921-05:00	2022-02-28 09:47:43.066921-05:00	True	True	False	False	

88037 rows × 46 columns

In [3]:
```
# Check for missing values in the 'created_at' column
print("Missing values in 'created_at':", tweets_df['created_at'].isnull().sum())
```

Conclusion:

The POTATO project successfully demonstrates the capability to ingest, analyze, and query Twitter data effectively. By providing meaningful insights into public discourse surrounding specific terms, the system serves as a valuable tool for researchers and analysts. Future enhancements could include expanding the dataset, improving privacy measures, and integrating additional analytical features. The results affirm the potential of the POTATO system to contribute to the understanding of social media dynamics.

Submitted by:

SATHISH KUMAR S

+91 8825635166

www.linkedin.com/in/sathish-kumar-s-44588022a