POTATO Take-Home Task Documentation

This Task is for the Entry Level Data Analyst Job at SilverSpace Technologies Inc.

POTATO (Panel-based Open Term-level Aggregate Twitter Observatory) is a system that has been created to let users request tweets containing certain terms (e.g., “COVID”) and obtain panelled responses about the tweets. This work shows how to consume large datasets from Twitter, perform effective queries on the data, and provide the answer to these queries in the form of a web application.

This specific implementation uses:

- **MongoDB** as the database for storing tweet data.
- **Flask** to build a web application where users can query the data.
- **Python** for data processing and cleaning, and for managing MongoDB interactions.

Project Breakdown

Part 1: Ingesting the Data

Objective: Load and stage Twitter data from the TSV files linked above in a manner that can be queried in an efficient manner.

Steps:

- **Data Loading:** The loaded data was the provided tweet data (that was about 500mb), and after loading the data we cleaned some of the data using the Pandas library.
- **Data Cleaning:** During data cleaning if there were any null values or invalid records, those were deleted in this step. The columns like timestamps were converted to respective datetime fields, and the numeric columns like the like_count were also changed into proper data types (here int64).
- **MongoDB Insertion:** The cleaned data was then directly inputted into a MongoDB by applying pymongo. Every record was saved as a document in a collection called tweets_cleaned_large.

Part 2: Querying the Data

Objective:

Develop capability that would allow the users to search for specific words in the tweets and obtain useful information from the findings.

Queries implemented:

- Total Tweets by Day: The search term is employed to search for all the tweets within a specific day, and the resulting number is recorded.

- Unique Users: This feature defines the exact number of users in your Twitter sample who tweeted at least one tweet with the search term.
- Average Likes: This is the average number of likes which was accrued by the search tweets containing the search term.
- Place IDs: The origin (place ID) of the tweets. The place IDs of the locations from where the tweets originated.
- Tweet Times: Classification of tweets according to the time of the tweet.
- Top User: The individual who tweeted most frequently with relation to the search term.

Part 3: How to Run the System

Prerequisites:

- MongoDB installed or a connection to MongoDB Atlas (cloud-hosted MongoDB).
- Python installed.
- Required Python libraries (pymongo, Flask, pandas, etc.).

Steps

- Step1: Clone the repository in VS Code Run this command in Terminal:

git clone https://github.com/Niranjan8185/POTATO-Take-Home-Task.git

- Step2: Install dependencies using requirements.txt file

Run this command: **pip install -r requirement.txt**

- Step 3: Use the large_tweet.pynb file to clean and store the data in mongoDB *(Important, change the name of collection in **large_tweet.ipynb** if you want to clean and store the data again , or else the data will be duplicated twice in the same collection).*

- Step 4: Run the Flask app by this command: **python app.py** *(Important , if you have changed the name of collection in **large_tweet.pynb** , then make sure to do the same in **app.py**)*

This will run the web app on **http://localhost:5000/**. You can open this URL in your browser.

- Step 5: Enter a search term (e.g., "COVID", "music") in the web interface to query the data. The app will return statistics based on the tweets containing the search term.

Conclusion:

The project shows how big data sets can be consumed into MongoDB, build a web application using Flask to enable querying data from Twitter, and how the real-time insights on search terms can be created. This system is the most suitable due to its scalability, efficiency, and its ability to support the analysis of tweet data.