POTATO Take-Home Task Documentatio

1. Introduction

This document provides instructions on how to ingest and query a dataset of tweets based on specific terms (such as “music”). The system is designed to allow users to easily search for a term in a dataset and receive detailed insights regarding the tweets associated with that term.

The key queries include:
- Daily counts of tweets containing the term.
- Unique users who posted tweets containing the term.
- Average likes for tweets containing the term.
- Place IDs where the tweets originated.
- Times of day when the tweets were posted.
- The user who posted the most tweets containing the term.

This document explains the system’s setup, usage, and the design choices made during development.

---

2. System Overview
The system is built using Python, and it handles TSV (Tab-Separated Values) files containing tweet data. These files can be large, so efficiency in processing is important. The steps involved include:
1. Data Ingestion: Loading and parsing large tweet datasets.
2. Data Cleaning: Ensuring the correct formatting of time columns and handling missing data.
3. Querying the Data: Implementing filters and grouping mechanisms to answer key questions.
4. Output: Displaying insights based on the search term input.

The system allows the user to run these queries with minimal setup on their own computer, provided they have Python and the necessary libraries installed.

---

3. Data Ingestion
The tweet data is provided in TSV format, which is parsed into a Pandas DataFrame for easy querying and manipulation. The main columns of interest include:
- id: Tweet ID
- text: Tweet content
- author_id: User ID of the tweet author
- created_at: Date and time when the tweet was posted
- like_count: Number of likes the tweet received
- place_id: Geographical place ID associated with the tweet

Justification: Using a Pandas DataFrame allows efficient data manipulation and supports large datasets, making it ideal for this type of analysis.

4. Query Functionality
Once the dataset is loaded, the following queries are supported:

1. How many tweets were posted containing the term on each day?

  - The system filters the dataset based on the search term and groups tweets by date to return daily tweet counts.

2. How many unique users posted a tweet containing the term?

  - The system calculates the number of distinct users who posted tweets containing the term.

3. How many likes did tweets containing the term get, on average?

  - The system calculates the average number of likes for all tweets containing the search term.

4. Where (in terms of place IDs) did the tweets come from?
  - The system groups tweets by `place_id` to show the geographical locations of the tweets.

5. What times of day were the tweets posted at?
  - The system extracts the hour from the `created_at` timestamp and groups tweets by hour to return tweet activity by time of day.

6. Which user posted the most tweets containing the term?
  - The system identifies the user with the highest tweet count for the given term.

Justification: Each of these queries provides valuable insights into the behavior and engagement patterns of users, which aligns with the goals of POTATO. The choice to focus on these queries is based on the need to give users a detailed understanding of trends in the data.

---

5. How to Run the System

Prerequisites:
- Python 3.x installed on your system.
- Required Python libraries: `pandas`, `numpy`, etc.

Setup Instructions:
1. Clone the repository or download the code.
2. Install the required libraries using `pip`:
  ```bash
  pip install pandas numpy
  ```
3. Place the TSV file (e.g., tweets about Britney Spears) in the project directory.
4. Run the Python script by executing:
  ```bash
  python analyze_tweets.py
  ```
5. When prompted, input the search term you wish to analyze (e.g., "music").

Output:
The system will display the results for each of the queries mentioned above, such as daily tweet

counts, unique users, average likes, tweet origins by place ID, tweet activity by time, and the most active user.

---

6. Design Choices

- Use of Pandas: Pandas is used for data manipulation due to its efficiency in handling large datasets and its built-in support for grouping and filtering, which is essential for querying this dataset.
  - Use of Datetime Functions: Converting timestamps to datetime objects allows for efficient extraction of date and time components, which is necessary for answering time-related queries.

- TSV File Format: The data is stored in TSV format, which is easy to parse and suitable for large datasets with many columns.

- Handling of Large Datasets: The system uses efficient filtering and grouping to handle potentially large datasets (~500MB), ensuring quick response times for user queries.

Justification:The design choices ensure that the system is scalable and can efficiently process large datasets without performance bottlenecks.

---

7. Conclusion

This system provides a powerful tool for analyzing large datasets of tweets based on user-defined search terms. By following the setup instructions and running the queries, users can quickly gain insights into the tweeting behavior of users and the engagement levels of specific terms.

For any further questions or support, feel free to reach out to the provided contact details.

Appendix: Example Output

Example query results for the term “music” are as follows:

- Tweets containing the term on each day:
  ```
  date  tweet_count
  2022-01-04         66
  2022-02-28       2935
  ```
- Number of unique users: 2109
- Average likes: 161.41
- Tweets by place ID: ...
- Tweets by hour: ...
- User with the most tweets:ID 118301422, 90 tweets