Twitter Geospatial Data Analytics Pipeline

Contents

1 Project Overview

This project implements a **batch data processing and analytics pipeline** for large-scale **Twitter geospatial data** (approximately 14.2 million tweets). The goal is to perform **data cleansing, transformation, feature engineering, temporal and geospatial aggregation**, and **export structured insights** to support downstream analytics and dashboarding. The pipeline is built using **AWS Glue (PySpark)** and leverages **Amazon S3** for data storage.

2 Implemented Tasks

2.1 Data Ingestion & Extraction

- Uploaded raw Twitter dataset (ZIP format) into **S3 (raw-data bucket)**.
- Created an AWS Glue job to:
  - Extract the CSV file from the ZIP archive.
  - Store parsed data into an **S3 output bucket**.
- *Outcome*: Extracted CSV available in S3 for processing.

2.2 Data Storage & Format Standardization

- Converted raw CSV files into **Parquet format** for optimized storage and querying.
- Stored data into **S3 in partitioned format**.
- Verified schema and structure using **AWS Athena**.
- *Outcome*: Compact, query-optimized data format (Parquet).

2.3 Feature Engineering

- Normalized **timestamps** to IST/UTC.
- Extracted new features:
  - `hour_of_day`
  - `day_of_week`
  - `is_weekend`
  - **Geospatial bins** (latitude/longitude buckets).
- Implemented reusable utility functions for transformations.
- *Outcome*: Enriched dataset with engineered features.

2.4 Timezone Mapping

- Implemented a **PySpark Glue job** to map **tweet geocoordinates to US timezones** using bounding box logic.
- Assigned a `timezone` field to each tweet.
- *Outcome*: Each tweet tagged with a timezone (Eastern, Central, Mountain, Pacific, Other).

2.5 Aggregation by Timezone

- Aggregated **tweet counts by timezone**.
- Wrote results into **S3 in Parquet format**, partitioned by `timezone`.
- *Outcome*: Ready-to-query aggregated data by timezone.

2.6 Temporal Activity Analysis

- Performed analysis of **temporal tweet activity** across US timezones.
- Calculated **hourly tweet flow per timezone**.
- Identified **peak tweet hours** for each timezone.
- Compared cross-timezone patterns to highlight behavioral differences.
- *Outcome*: Temporal trends and peak-hour metrics established.

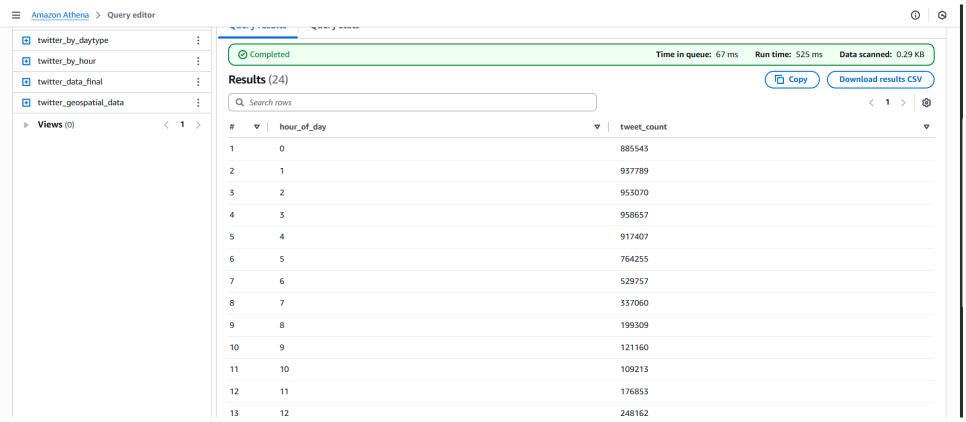
2.7 Structured Aggregation Exports

- Aggregated results into **structured tables** containing:
  - Top-hour metrics.
  - Tweet counts per timezone, per hour.
- Exported data to **S3 in CSV and Parquet formats**.
- Organized folder structure for **dashboard consumption**.
- *Outcome*: Clean, pre-aggregated exports available for BI dashboards.

3 Visual Results

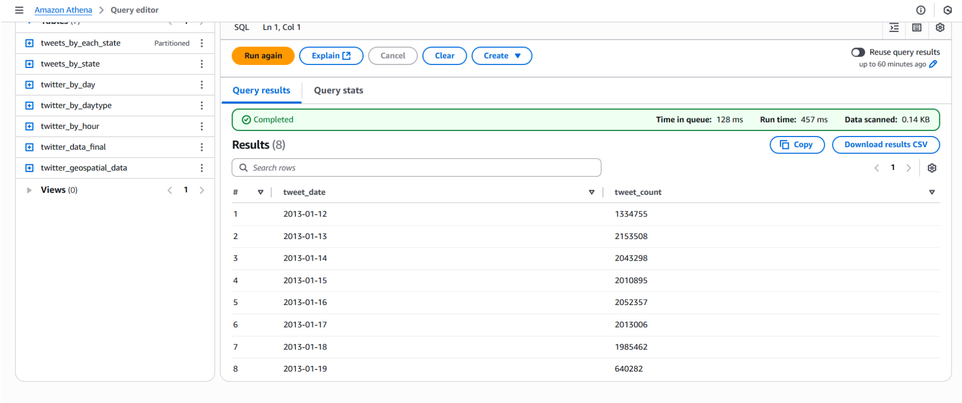
The following visualizations were generated based on Athena query results (replace the placeholder image paths with the actual file names or paths where you save the images):

- **Time-Series Aggregation – Hour**: Visualization of hourly tweet activity.

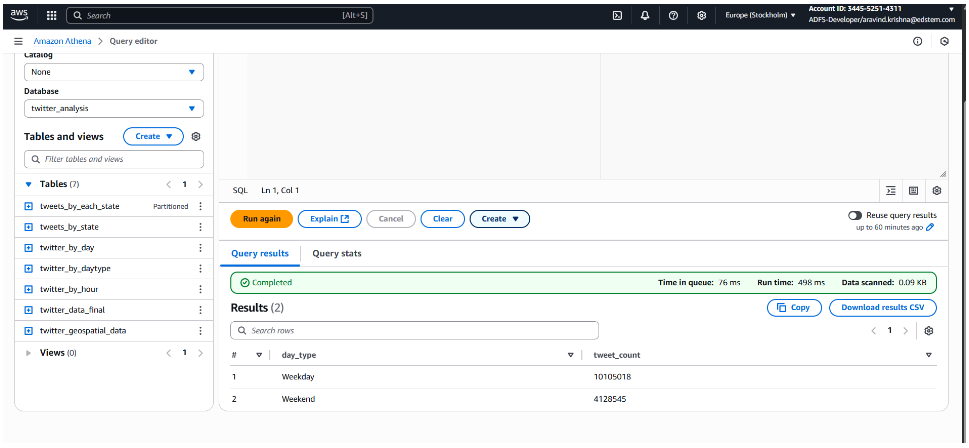


#	hour_of_day	tweet_count
1	0	885543
2	1	937789
3	2	953070
4	3	958657
5	4	917407
6	5	764255
7	6	529757
8	7	337060
9	8	199309
10	9	121160
11	10	109213
12	11	176853
13	12	248162

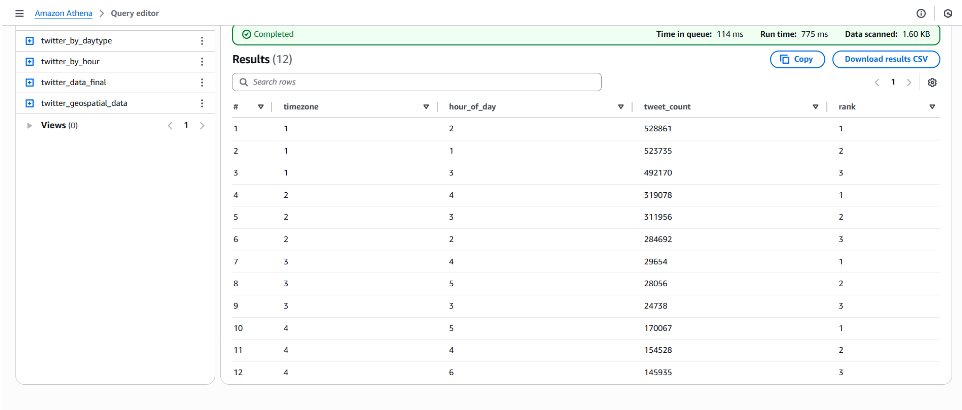
- **Time-Series Aggregation – Day**: Visualization of daily tweet activity.



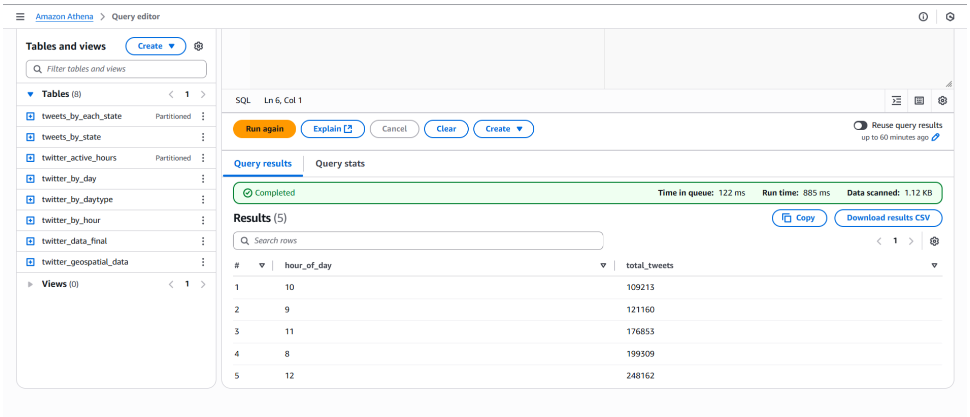
- **Time-Series Aggregation – Day Type**: Visualization of tweet activity by day type (weekday/weekend).



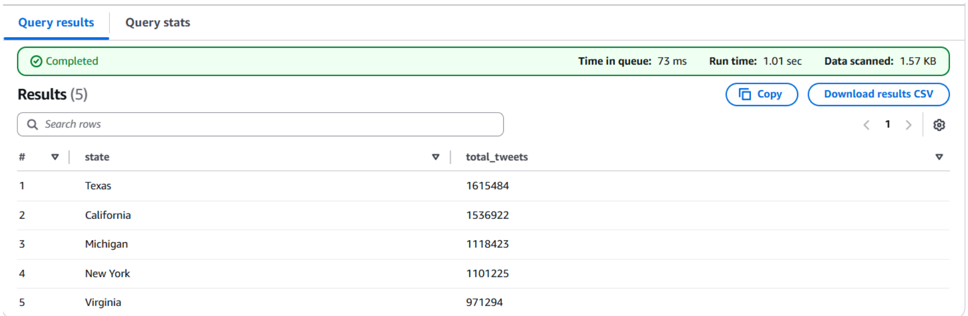
- **Timezone Metrics – Peak Hours**: Visualization of peak tweeting hours by timezone.



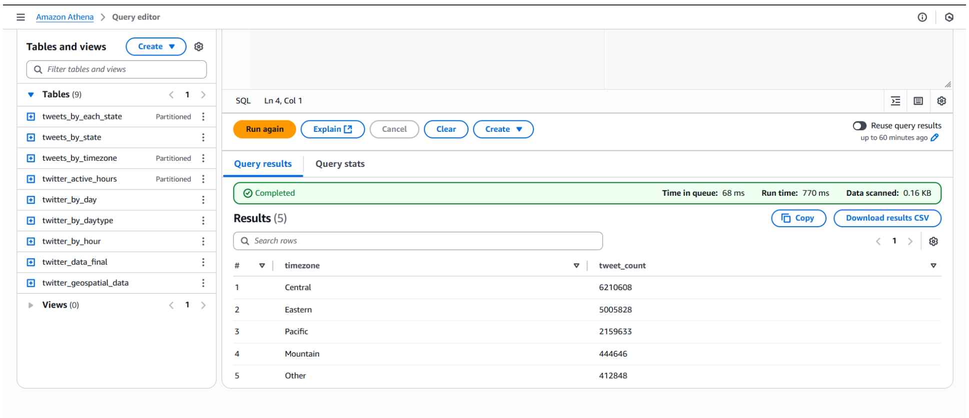
- **Timezone Metrics – Idle Hours**: Visualization of idle tweeting hours by timezone.



- **Geospatial Aggregation – Tweets by US States**: Visualization of tweet distribution across US states.



- **Timezone Classification & Metrics**: Visualization of tweet counts by timezone.



4 Glue Job Profiling & Performance Analysis

- Completed **profiling of all Glue jobs** with multiple runs.
- Generated a consolidated **CSV performance report** containing:
  - Job Name
  - Total Runs
  - Success/Failure Count
  - Average Execution Time
  - Total DPU Hours
  - DPU Hours (Success Runs Only)
- *Highlights*:
  - Profiled **9 Glue jobs**.
  - Identified variability in execution times across jobs.
  - Established baseline for benchmarking and optimization.
- *Outcome*: Execution profiling report for cost and performance insights.

5 Performance Optimization

- Identified inefficiency in the **twitter-feature-engineering job** due to redundant `.count()` operations.
- Fix applied:
  - Removed intermediate `.count()` calls.
  - Replaced with a **single `.count()`** at the end.
  - Used `.limit(10)` sampling for debugging.
- Verified correctness and **improved runtime**.
- All other jobs were already efficient.
- *Outcome*: Optimized job execution without loss of accuracy.

6 Final Deliverables

- **Processed Data in S3**:
  - Partitioned **Parquet and CSV outputs** for states, hours, and timezones.
- **Dashboard-Ready Outputs**:
  - Clean, structured aggregations for direct ingestion.
- **Performance Profiling Report**:
  - Consolidated Glue job execution metrics in CSV format.

7 Key Insights

- Tweets mapped to **US timezones** using bounding box geolocation logic.
- Clear **temporal trends** identified:
  - Peak tweeting hours vary across Eastern, Central, Mountain, and Pacific timezones.
- Optimized pipeline ensures **faster execution and reduced costs**.

8 Conclusion

This project successfully:

- Implemented an **end-to-end PySpark Glue pipeline** for Twitter geospatial data.
- Delivered **clean, enriched, and aggregated datasets** for downstream analytics.
- Performed **job profiling and optimization** to ensure scalability and efficiency.

The pipeline is production-ready and supports future extensions such as **real-time streaming ingestion** and **advanced geospatial clustering**.