

Twitter Geospatial Data Analytics Pipeline

Contents

1	Project Overview	2
2	Implemented Tasks	2
2.1	Data Ingestion & Extraction	2
2.2	Data Storage & Format Standardization	2
2.3	Feature Engineering	2
2.4	Timezone Mapping	2
2.5	Aggregation by Timezone	3
2.6	Temporal Activity Analysis	3
2.7	Structured Aggregation Exports	3
3	Visual Results	3
4	Glue Job Profiling & Performance Analysis	6
5	Performance Optimization	6
6	Final Deliverables	6
7	Key Insights	7
8	Conclusion	7

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.

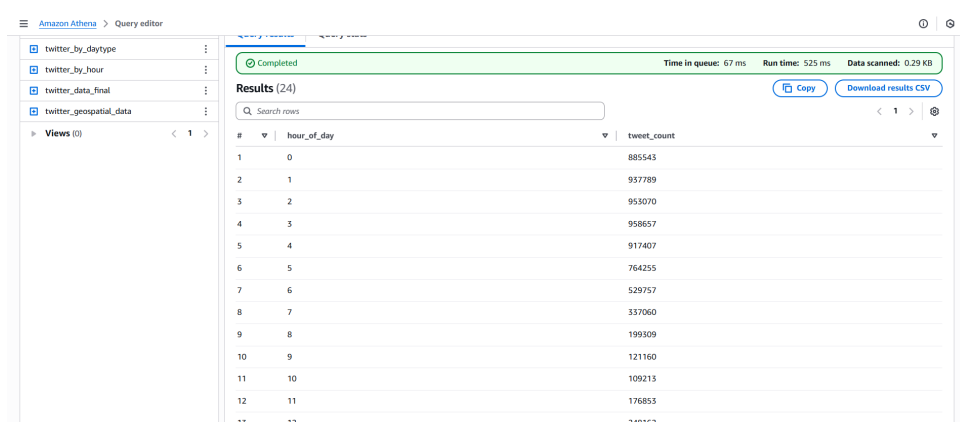
2.7 Structured Aggregation Exports

- 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.



The screenshot shows the Amazon Athena Query Editor interface. On the left, a sidebar lists four tables: `twitter_by_daytype`, `twitter_by_hour`, `twitter_data_final`, and `twitter_geospatial_data`. The main panel displays the results of a query, showing a table with 24 rows. The table has three columns: `#`, `hour_of_day`, and `tweet_count`. The results show tweet counts for each hour of the day, with the highest count at hour 11 (109,213) and the lowest at hour 0 (88,543).

#	hour_of_day	tweet_count
1	0	885543
2	1	957789
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.

The screenshot shows the Amazon Athena Query Editor interface. On the left, a sidebar lists tables and views: tweets_by_each_state, tweets_by_state, twitter_by_day, twitter_by_daytype, twitter_by_hour, twitter_data_final, and twitter_geospatial_data. The main panel displays the query results for a completed query. The query status bar indicates 'Completed' with a time in queue of 128 ms, a run time of 457 ms, and data scanned of 0.14 KB. The results table has 8 rows and 3 columns: #, tweet_date, and tweet_count.

#	tweet_date	tweet_count
1	2013-01-12	1334755
2	2013-01-13	2155508
3	2013-01-14	2043298
4	2013-01-15	2010895
5	2013-01-16	2052357
6	2013-01-17	2013006
7	2013-01-18	1985462
8	2013-01-19	640282

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

The screenshot shows the Amazon Athena Query Editor interface. On the left, a sidebar lists tables and views: tweets_by_each_state, tweets_by_state, twitter_by_day, twitter_by_daytype, twitter_by_hour, twitter_data_final, and twitter_geospatial_data. The main panel displays the query results for a completed query. The query status bar indicates 'Completed' with a time in queue of 76 ms, a run time of 498 ms, and data scanned of 0.09 KB. The results table has 2 rows and 3 columns: #, day_type, and tweet_count.

#	day_type	tweet_count
1	Weekday	10105018
2	Weekend	4128545

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

The screenshot shows the Amazon Athena Query Editor interface. On the left, a sidebar lists tables and views: twitter_by_daytype, twitter_by_hour, twitter_data_final, and twitter_geospatial_data. The main panel displays the query results for a completed query. The query status bar indicates 'Completed' with a time in queue of 114 ms, a run time of 775 ms, and data scanned of 1.60 KB. The results table has 12 rows and 6 columns: #, timezone, hour_of_day, tweet_count, rank, and an additional column.

#	timezone	hour_of_day	tweet_count	rank	
1	1	2	528861	1	
2	1	1	523735	2	
3	1	3	492170	3	
4	2	4	319078	1	
5	2	3	311956	2	
6	2	2	284692	3	
7	3	4	29654	1	
8	3	5	28056	2	
9	3	3	24738	3	
10	4	5	170067	1	
11	4	4	154528	2	
12	4	6	145935	3	

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

The screenshot shows the Amazon Athena Query Editor interface. On the left, a sidebar lists tables and views, including 'tweets_by_each_state', 'tweets_by_state', 'twitter_active_hours', 'twitter_by_day', 'twitter_by_daytype', 'twitter_by_hour', 'twitter_data_final', and 'twitter_geospatial_data'. The main panel displays a query result for a query named 'Ln 6, Col 1'. The query is completed, with a time in queue of 122 ms, a run time of 885 ms, and data scanned of 1.12 KB. The results are shown in a table with 5 rows and 3 columns: '#', 'hour_of_day', and 'total_tweets'.

#	hour_of_day	total_tweets
1	10	109213
2	9	121160
3	11	176853
4	8	199509
5	12	248162

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

The screenshot shows the Amazon Athena Query Editor interface. The main panel displays a query result for a query named 'Ln 4, Col 1'. The query is completed, with a time in queue of 73 ms, a run time of 1.01 sec, and data scanned of 1.57 KB. The results are shown in a table with 5 rows and 3 columns: '#', 'state', and 'total_tweets'.

#	state	total_tweets
1	Texas	1615484
2	California	1536922
3	Michigan	1118423
4	New York	1101225
5	Virginia	971294

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

The screenshot shows the Amazon Athena Query Editor interface. The main panel displays a query result for a query named 'Ln 4, Col 1'. The query is completed, with a time in queue of 68 ms, a run time of 770 ms, and data scanned of 0.16 KB. The results are shown in a table with 5 rows and 3 columns: '#', 'timezone', and 'tweet_count'.

#	timezone	tweet_count
1	Central	6210608
2	Eastern	5005828
3	Pacific	2159633
4	Mountain	444646
5	Other	412848

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**.