NYC Taxi Mobility Dashboard – Technical Report (Summary)

1. Executive Summary & Problem Framing

This report details the architecture and insights of a rebuilt NYC Taxi Mobility Dashboard, designed to analyze 1.4 million taxi trip records from 2016. The initial system was unstable, suffering from memory leaks, inefficient queries, and data processing failures, leading to frequent crashes. This revised version represents a ground-up rebuild focused on robustness, performance, and scalability.

Dataset & Key Challenges

- Dataset: Contains trip details including timestamps, passenger counts, GPS coordinates, and trip duration.
- Data Quality Issues: The primary challenge was significant data anomalies, including:
    - Missing values and invalid coordinates outside NYC.
    - Critical Discovery: Approximately 15% of calculated trip speeds were unrealistic (exceeding 100 mph), revealing major data integrity issues stemming from short durations or GPS errors.
- Cleaning Strategy: Implemented geographic filtering (NYC boundaries), temporal validation (trip durations between 0–2 hours), and custom anomaly detection to ensure data reliability.

2. System Architecture & Algorithmic Core

Architecture & Design

- Frontend: HTML/CSS/JavaScript with Chart.js for a lightweight, fast-loading interface.
- Backend: Flask (Python) for rapid development and integration with data science libraries.
- Database: SQLite with a normalized schema (separate `trips`, `locations`, and `derived_metrics` tables) to reduce redundancy and improve query performance. Strategic indexing was applied to key fields.

Core Custom Algorithms

1. **Z-Score Anomaly Detection**
    - Problem: Identify and filter unrealistic data points (e.g., extreme speeds) without relying on external libraries.
    - Logic: Manually calculates mean and standard deviation, then flags records where the z-score exceeds a threshold (e.g., three standard deviations).
    - Complexity: *O(n)* time, *O(k)* space.

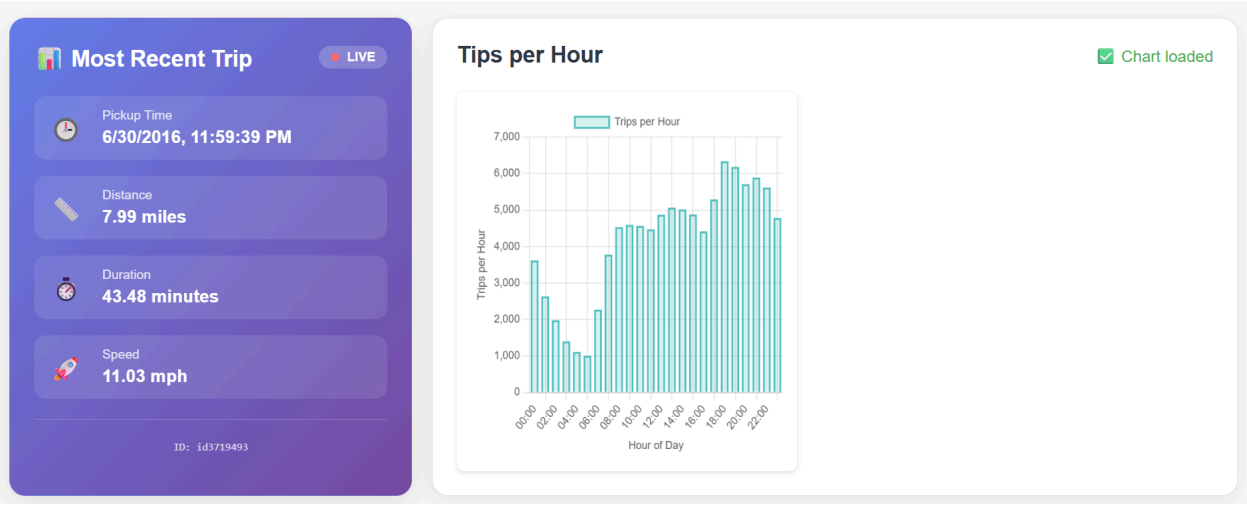
2. **Haversine Distance Calculation**
    - Problem: Accurately calculate the geographic distance between pickup and dropoff coordinates.
    - Logic: Implements the Haversine formula to compute great-circle distances on the Earth's surface.
    - Complexity: *O(1)* time and space.

3. **Efficient Data Sampling**
    - Problem: Enable real-time chart rendering without processing the entire 1.4M record dataset.
    - Logic: Uses SQL to randomly sample a large, filtered subset of records (e.g., 100,000) for visualization, maintaining statistical significance while drastically improving performance.
    - Complexity: *O(k)* time and space, where *k* is the sample size.

3. Key Insights & Interpretation

Insight 1: Peak Hour Congestion Patterns

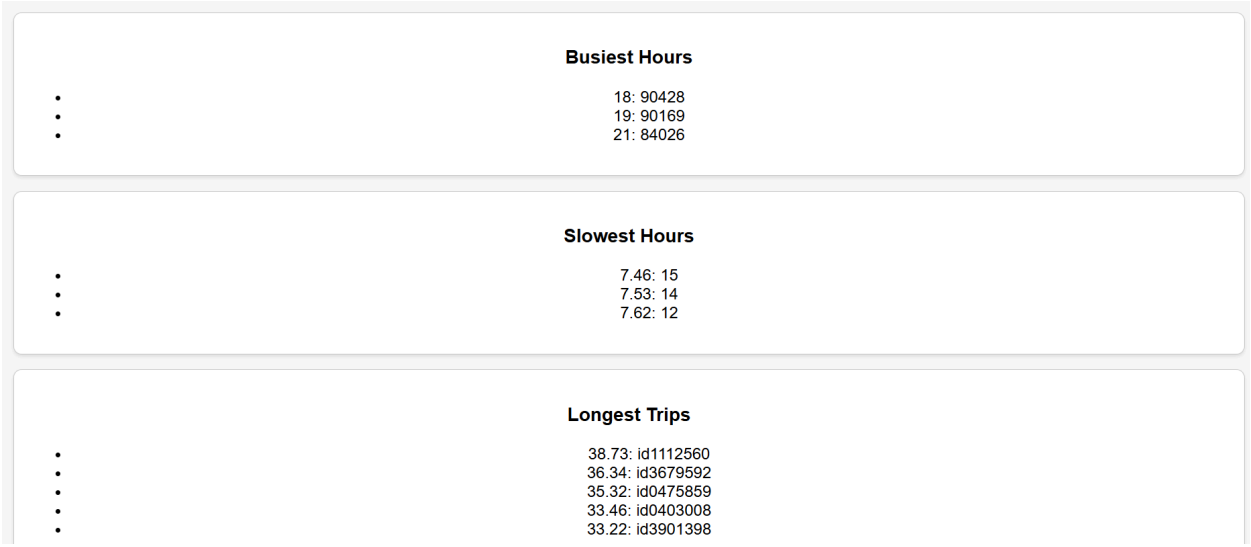
- Derivation: Aggregated trip counts by hour of the day.
- Finding: Clear peak hours were identified: 6–8 PM (evening rush) and 8–10 AM (morning rush). A significant trough occurs between 2–4 AM.
- Interpretation: This reflects classic urban commuting behavior and highlights opportunities for dynamic pricing and resource allocation during high-demand periods.



Insight 2: Geographic Distribution of Long Trips

- Derivation: Identified the longest trips by calculated distance using the Haversine algorithm.
- Finding: The longest trips (over 15 miles) predominantly occur during off-peak hours and connect outer boroughs to Manhattan.

- Interpretation: These are likely airport transfers or long-distance commutes, providing valuable insight for optimizing ride-sharing services and airport shuttle logistics.



4. Reflection & Future Roadmap

Technical Challenges Overcome

- Data Quality: Addressed through custom anomaly detection and filtering rules.
- Performance: Solved via database indexing, intelligent sampling, and pagination, reducing query times from >30 seconds to under 2 seconds.

Future Improvements

Technical Enhancements:

- Migrate to a more scalable database (e.g., PostgreSQL).
- Implement real-time data streaming (e.g., Apache Kafka).
- Integrate machine learning for predictive demand forecasting and advanced anomaly detection.

Feature Additions:

- Add interactive geographic maps for trip density visualization.
- Include comparative analysis (year-over-year, seasonal).
- Develop user authentication and data export functionality.