Project Task Analysis

Project Task for Citi Bike Trip Histories

Group: 11

A. Domain Situation: User Stories

Here are five user stories speculating on the tasks of typical users for the Citi Bike dataset. The overarching theme is **"Understanding and optimizing Citi Bike usage patterns and operational efficiency in NYC."**

1. **Persona:** Sarah, a marketing analyst for Citi Bike.

   - **Task/Question:** Sarah is planning a new campaign to convert "casual" riders into "member" riders. To target this campaign effectively, she needs to know: **"How do member and casual users differ in their riding activity across hours of the day and days of the week?"**

2. **Persona:** Chen, an urban planner for the NYC Department of Transportation (DOT).

   - **Task/Question:** Chen is evaluating the impact of existing bike lanes and assessing the need for new ones. He needs to understand traffic flow and requires an answer to: **"What are the most common origin-destination routes based on geographic coordinates? Which routes are most popular during peak commuter hours?"**

3. **Persona:** Alex, an operations manager for Citi Bike (similar to the "Max" example).

   - **Task/Question:** Alex is responsible for "rebalancing" stations—moving bikes from full stations to empty ones. He needs to optimize his truck routes and schedules. His daily question is: **"Which stations are the most popular start stations and end stations? What is the net inflow/outflow of bikes for the top 20 busiest stations during morning (7-10 AM) and evening (5-7 PM) rush hours?"** (Uses `start_station_id` and `end_station_id`).

4. **Persona:** Priya, a data quality analyst for the Citi Bike tech team.

   - **Task/Question:** Priya must monitor the system for data errors or anomalous usage that could indicate software bugs or bike malfunctions. She needs to address the known issue of outliers. Her task is to ask: **"What is the distribution of trip durations? Are there significant numbers of trips that are too short (< 60 seconds) or too long (> 24 hours), and how are these outliers distributed across different `rideable_type`s?"**

5. **Persona:** David, a fleet maintenance supervisor for Citi Bike.

   - **Task/Question:** David manages the maintenance schedules for different bike models. He knows that electric bikes (`rideable_type`) have different usage patterns and can be parked anywhere, leading to missing end station data. He needs to know: **"How does the usage of electric bikes differ from classic bikes in terms of average trip duration and common parking zones (using `end_lat`, `end_lng`), especially for trips that do not end at a designated station?"**

B. Task Abstraction

Here is a task abstraction for two of the user stories.

User Story 1: Marketing Analysis

> **Story:** "Sarah (Marketing) needs to know: 'How do member and casual users differ in their riding activity across hours of the day and days of the week?'"

Abstraction:

- **High level: Discover**. (The user is asking "how" things differ, which is exploratory, similar to Scenario 6: "How do... distributions differ?").
- **Mid/Search: Lookup**. (She knows the two groups she wants to compare: 'member' and 'casual'. This matches Scenario 6).
- **Low/Query: Compare**. (The keyword "differ" maps directly to this action).
- **Target: Distributions** (by group).
- **Final: Compare distributions**.

User Story 2: Urban Planning

> **Story:** "Chen (Urban Planner) needs to know: 'What are the most common origin-destination routes based on geographic coordinates? Which routes are most popular during peak commuter hours?'"

Abstraction:

- **High level: Discover**. (The user is asking "what" or "which" are the "most", which is an exploratory discovery task, similar to Scenario 4: "...which directories account for the most total size?").
- **Mid/Search: Explore**. (The user does not know the target in advance; they are exploring the entire set to find the top items).
- **Low/Query: Summarize (Rank)**. (The solutions file uses "Rank" for "most" or "highest". The goal is to aggregate all trips and rank the resulting routes).
- **Target: Paths** (or Routes).
- **Final: Rank routes**. (Following the pattern of Scenario 4: "Rank directories").