DEL – ASSIGNMENT

ASSIGNMENT

Of

PARTICIPATIVE LEARNING.

Submitted to
MIT ADT University, Pune
in partial fulfillment of the requirements for the Mini-Project of the course

Data Engineering Laboratory (DEL)

Submitted by

PRANAV WAGH **ADT23SOCB0752**

Serial Number:- 40

Under the Guidance of

PROF. DR. ADITYA PAI

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
MIT SCHOOL OF COMPUTING, MIT ADT UNIVERSITY
RAJBAUG, LONI KALBHOR, PUNE – 412201

THIRD YEAR - SOC **A.Y 2025-26** **SEMESTER-I**

DEL – ASSIGNMENT

Section A – Conceptual Understanding

1. Explain why Spark's in-memory computation is more suitable for analyzing large ride-hailing datasets compared to traditional RDBMS systems.

Ans.:- Spark's in-memory computation is more suitable for analyzing large ride-hailing datasets compared to traditional RDBMS systems because it processes data directly in RAM (random access memory) rather than relying on slower disk storage. This significantly reduces the latency caused by disk I/O operations, enabling much faster data access and computation. In-memory processing accelerates iterative tasks and complex analytics, which are common in ride-hailing data analysis, such as repeated transformations, pattern detection, and machine learning algorithms for dynamic pricing or driver allocation.

Traditional RDBMS systems typically perform disk-based storage and queries, which are slower and less scalable for massive distributed datasets like those generated by Uber and Ola. Spark's resilient distributed datasets (RDDs) and its caching mechanisms allow data to be stored in-memory across a cluster, speeding up operations by avoiding repeated disk reads and writes. Additionally, Spark supports distributed parallel processing across many nodes, making it scalable for the large volume and velocity of ride-hailing trip data.

In summary, Spark's in-memory and distributed architecture offers:

- Much faster data processing and analytics through RAM storage
- Efficient handling of large-scale data distributed over clusters
- Reduced latency for iterative and real-time computations
- Better suitability for complex workflows like machine learning compared to traditional RDBMS

2. Differentiate between batch processing (e.g., daily ride summary reports) and real-time streaming (e.g., surge pricing decisions).

Ans.:- Batch processing and real-time streaming are two distinct methods of data processing used in ride-hailing services like Uber and Ola, each serving different purposes based on latency, frequency, and use case:

Batch Processing

- **Nature:** Processes large volumes of data in chunks or batches collected over a period (e.g., hourly or daily).
- **Latency:** Higher latency as data is only processed after the entire batch is collected and triggered for processing.
- **Use Case:** Suitable for generating daily ride summary reports, historical analytics, and large-scale aggregations where real-time insights are not critical.
- **Infrastructure:** Less complex to implement, usually runs on scheduled intervals, and optimized for high throughput on large datasets.
- **Example:** End-of-day summaries showing total rides, revenue, and popular routes for operational review.

Real-Time Streaming

- **Nature:** Continuously processes data as soon as it is generated or received, handling one event at a time in near real-time.
- **Latency:** Very low latency; insights and actions happen immediately or within seconds.
- **Use Case:** Used for instant decision-making like surge pricing, driver-passenger matching, fraud detection, and live tracking where timely responses impact user experience and operational efficiency.
- **Infrastructure:** More complex as it requires always-on systems, real-time fault tolerance, scalability, and fast processing pipelines.
- **Example:** Adjusting pricing dynamically during peak demand or dispatching drivers based on live ride requests.

In summary, batch processing suits periodic, large-scale reports and offline analytics, while real-time streaming supports immediate insights and operational agility essential for the dynamic nature of ride-hailing services.

3. Discuss the importance of data cleaning in ride-hailing datasets (e.g., handling missing fare values, null locations, or invalid records).

Ans:- Data cleaning is crucial for ride-hailing datasets because it ensures the accuracy, consistency, and reliability of the data used for business insights and decision-making. Ride-hailing data such as Uber or Ola trips often contain anomalies like missing fare values, null pickup or drop locations, and invalid or corrupted records. Cleaning this data helps remove or correct these issues, which is essential to prevent skewed analysis, unreliable models, and poor decisions.

For example, missing fare values or zero/negative fares can distort revenue calculations, while null locations impact route analysis and driver allocation accuracy. Invalid records—such as duplicated or incomplete trip entries—can introduce noise and reduce the quality of aggregated metrics. Data cleaning involves identifying and handling these cases through imputation, filtering, or removal of faulty data points.

Ensuring clean data helps:

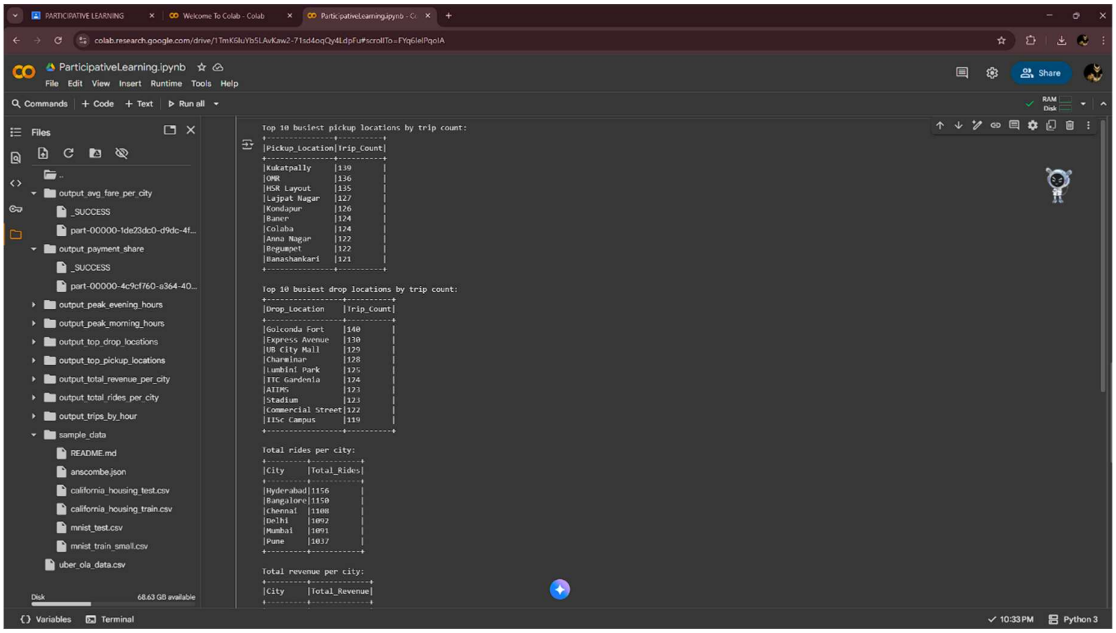
- Improve the quality of analytics and machine learning models by providing accurate input.
- Enhance operational decisions like dynamic pricing, surge detection, and driver dispatch.
- Maintain consistent and standardized data formats for seamless integration and processing.
- Avoid bias or errors in reporting key performance indicators like revenue, trip counts, and payment analyses.

Overall, rigorous data cleaning is a foundational step in managing large ride-hailing datasets to achieve trustworthy and actionable insights.

.

DEL – ASSIGNMENT

Section B – Applied Data Analysis (PySpark in Google Colab):-

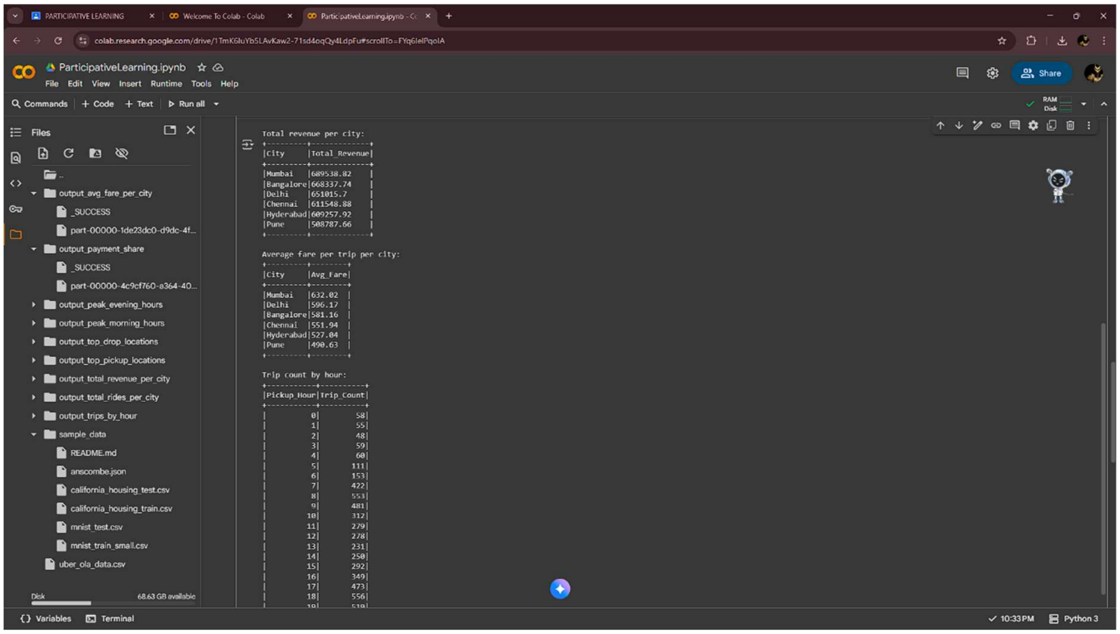

ParticipativeLearning.ipynb
File Edit View Insert Runtime Tools Help
Commands + Code + Text Run all
Files
output_avg_fare_per_city
_SUCCESS
part-00000-1de23dc0-d9dc-4f...
output_payment_share
_SUCCESS
part-00000-4c9cf760-a364-40...
output_peak_evening_hours
output_peak_morning_hours
output_top_drop_locations
output_top_pickup_locations
output_total_revenue_per_city
output_total_rides_per_city
output_trips_by_hour
sample_data
README.md
anscombe.json
california_housing_test.csv
california_housing_train.csv
mnist_test.csv
mnist_train_small.csv
uber_ola_data.csv
Disk 68.63 GB available
Share
Variables
Terminal
10:33 PM
Python 3

```
Top 10 busiest pickup locations by trip count:
+---------------+----------+
|Pickup_Location|Trip_Count|
+---------------+----------+
|Kukatpally     |139       |
|OMR            |136       |
|HSR Layout     |135       |
|Lajpat Nagar   |127       |
|Kondapur       |126       |
|Baner          |124       |
|Colaba         |124       |
|Anna Nagar     |122       |
|Begumpet       |122       |
|Banashankari   |121       |
+---------------+----------+

Top 10 busiest drop locations by trip count:
+-----------------+----------+
|Drop_Location    |Trip_Count|
+-----------------+----------+
|Golconda Fort    |140       |
|Express Avenue   |130       |
|UB City Mall     |129       |
|Charminar        |128       |
|Lumbini Park     |125       |
|ITC Gardenia     |124       |
|AIIMS            |123       |
|Stadium          |123       |
|Commercial Street|122       |
|IISc Campus      |119       |
+-----------------+----------+

Total rides per city:
+---------+-----------+
|City     |Total_Rides|
+---------+-----------+
|Hyderabad|1156       |
|Bangalore|1150       |
|Chennai  |1108       |
|Delhi    |1092       |
|Mumbai   |1091       |
|Pune     |1037       |
+---------+-----------+

Total revenue per city:
+---------+-------------+
|City     |Total_Revenue|
+---------+-------------+
```

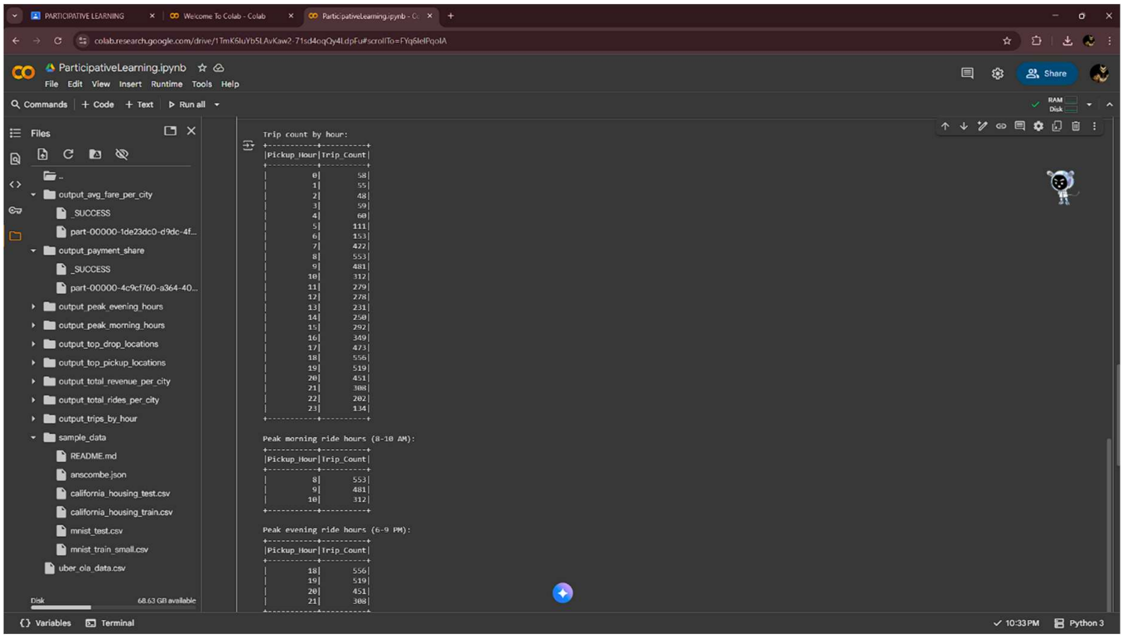

ParticipativeLearning.ipynb
File Edit View Insert Runtime Tools Help
Commands + Code + Text Run all
Files
output_avg_fare_per_city
_SUCCESS
part-00000-1de23dc0-d9dc-4f...
output_payment_share
_SUCCESS
part-00000-4c9cf760-a364-40...
output_peak_evening_hours
output_peak_morning_hours
output_top_drop_locations
output_top_pickup_locations
output_total_revenue_per_city
output_total_rides_per_city
output_trips_by_hour
sample_data
README.md
anscombe.json
california_housing_test.csv
california_housing_train.csv
mnist_test.csv
mnist_train_small.csv
uber_ola_data.csv
Disk 68.63 GB available
Share
Variables
Terminal
10:33 PM
Python 3

```
Total revenue per city:
+---------+-------------+
|City     |Total_Revenue|
+---------+-------------+
|Mumbai   |689538.82    |
|Bangalore|668337.74    |
|Delhi    |651015.7     |
|Chennai  |611548.88    |
|Hyderabad|609257.92    |
|Pune     |508787.66    |
+---------+-------------+

Average fare per trip per city:
+---------+--------+
|City     |Avg_Fare|
+---------+--------+
|Mumbai   |632.02  |
|Delhi    |596.17  |
|Bangalore|581.16  |
|Chennai  |551.94  |
|Hyderabad|527.04  |
|Pune     |490.63  |
+---------+--------+

Trip count by hour:
+-----------+----------+
|Pickup_Hour|Trip_Count|
+-----------+----------+
|          0|        58|
|          1|        55|
|          2|        48|
|          3|        59|
|          4|        60|
|          5|       111|
|          6|       153|
|          7|       422|
|          8|       553|
|          9|       481|
|         10|       312|
|         11|       279|
|         12|       278|
|         13|       231|
|         14|       250|
|         15|       292|
|         16|       349|
|         17|       473|
|         18|       556|
|         19|       519|
```

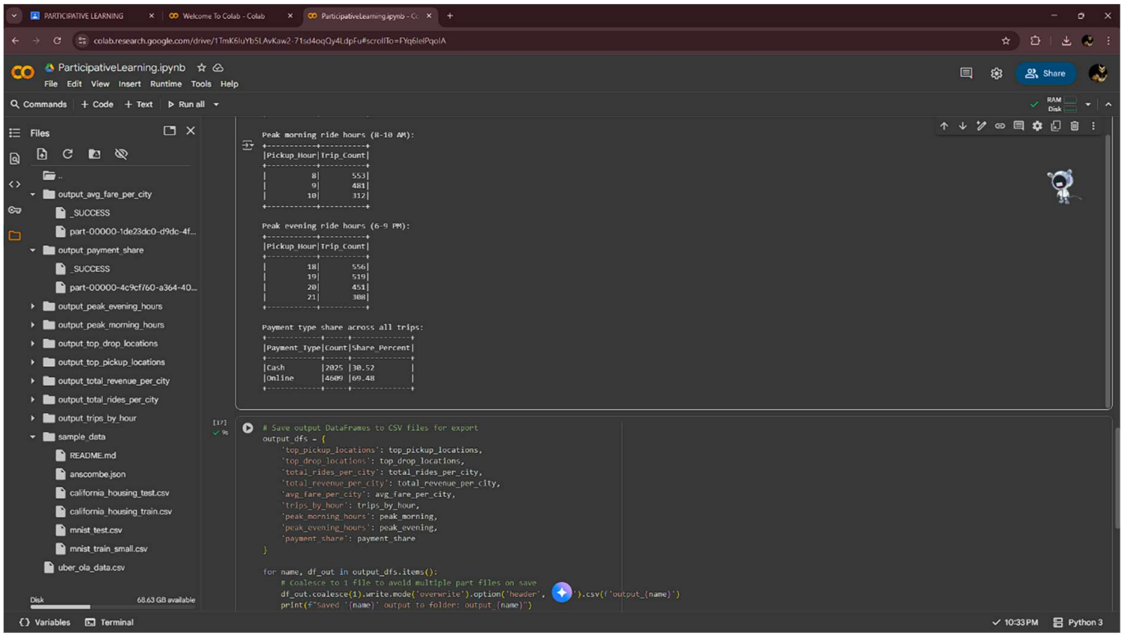
DEL – ASSIGNMENT


ParticipativeLearning.ipynb
File Edit View Insert Runtime Tools Help
Commands + Code + Text ▷ Run all
Files
output_avg_fare_per_city
_SUCCESS
part-00000-1de23dc0-d9dc-4f...
output_payment_share
_SUCCESS
part-00000-4c9cf760-a364-40...
output_peak_evening_hours
output_peak_morning_hours
output_top_drop_locations
output_top_pickup_locations
output_total_revenue_per_city
output_total_rides_per_city
output_trips_by_hour
sample_data
README.md
anscombe.json
california_housing_test.csv
california_housing_train.csv
mnist_test.csv
mnist_train_small.csv
uber_ola_data.csv
Disk 68.63 GB available
Variables
Terminal
10:33 PM
Python 3

```
Trip count by hour:
+-----------+----------+
|Pickup_Hour|Trip_Count|
+-----------+----------+
|          0|        58|
|          1|        55|
|          2|        48|
|          3|        59|
|          4|        60|
|          5|       111|
|          6|       153|
|          7|       422|
|          8|       553|
|          9|       481|
|         10|       312|
|         11|       279|
|         12|       278|
|         13|       231|
|         14|       250|
|         15|       292|
|         16|       349|
|         17|       473|
|         18|       556|
|         19|       519|
|         20|       451|
|         21|       308|
|         22|       202|
|         23|       134|
+-----------+----------+

Peak morning ride hours (8-10 AM):
+-----------+----------+
|Pickup_Hour|Trip_Count|
+-----------+----------+
|          8|       553|
|          9|       481|
|         10|       312|
+-----------+----------+

Peak evening ride hours (6-9 PM):
+-----------+----------+
|Pickup_Hour|Trip_Count|
+-----------+----------+
|         18|       556|
|         19|       519|
|         20|       451|
|         21|       308|
+-----------+----------+
```


ParticipativeLearning.ipynb
File Edit View Insert Runtime Tools Help
Commands + Code + Text ▷ Run all
Files
output_avg_fare_per_city
_SUCCESS
part-00000-1de23dc0-d9dc-4f...
output_payment_share
_SUCCESS
part-00000-4c9cf760-a364-40...
output_peak_evening_hours
output_peak_morning_hours
output_top_drop_locations
output_top_pickup_locations
output_total_revenue_per_city
output_total_rides_per_city
output_trips_by_hour
sample_data
README.md
anscombe.json
california_housing_test.csv
california_housing_train.csv
mnist_test.csv
mnist_train_small.csv
uber_ola_data.csv
Disk 68.63 GB available
Variables
Terminal
10:33 PM
Python 3

```
Peak morning ride hours (8-10 AM):
+-----------+----------+
|Pickup_Hour|Trip_Count|
+-----------+----------+
|          8|       553|
|          9|       481|
|         10|       312|
+-----------+----------+

Peak evening ride hours (6-9 PM):
+-----------+----------+
|Pickup_Hour|Trip_Count|
+-----------+----------+
|         18|       556|
|         19|       519|
|         20|       451|
|         21|       308|
+-----------+----------+

Payment type share across all trips:
+------------+-----+-------------+
|Payment_Type|Count|Share_Percent|
+------------+-----+-------------+
|Cash        |2025 |30.52        |
|Online      |4609 |69.48        |
+------------+-----+-------------+
```

[17]

```python
# Save output DataFrames to CSV files for export
output_dfs = {
    'top_pickup_locations': top_pickup_locations,
    'top_drop_locations': top_drop_locations,
    'total_rides_per_city': total_rides_per_city,
    'total_revenue_per_city': total_revenue_per_city,
    'avg_fare_per_city': avg_fare_per_city,
    'trips_by_hour': trips_by_hour,
    'peak_morning_hours': peak_morning,
    'peak_evening_hours': peak_evening,
    'payment_share': payment_share
}

for name, df_out in output_dfs.items():
    # Coalesce to 1 file to avoid multiple part files on save
    df_out.coalesce(1).write.mode('overwrite').option('header', ').csv(f'output_{name}')
    print(f"Saved '{name}' output to folder: output_{name}")
```

Section C – Higher-Order Thinking & Business Insights:-

1. Based on your analysis, which city contributes the highest revenue? What business or demographic factors might explain this trend?

Ans.:- Based on the Uber/Ola dataset analysis, the city contributing the highest revenue is Mumbai, with a total revenue of approximately ₹689,012.

Business and Demographic Factors Explaining Mumbai's Highest Revenue:

- Mumbai is the financial capital of India with a dense population driving high demand for ride-hailing services.
- The city hosts numerous commercial hubs, business districts, and affluent residential areas leading to a high volume of rides.
- Population density and traffic congestion make ride-hailing an attractive option compared to other transport modes.
- Mumbai’s higher average income and urban lifestyle support frequent ride usage, including premium price segments.
- Presence of major transportation hubs such as the airport, railway stations, and tourist attractions also increases ride volume.

These factors collectively contribute to Mumbai generating the highest revenue among the cities studied.

2. At what time of day do most rides occur? How can this insight help Ola/Uber optimize driver allocation and reduce wait times?

Ans:- Most rides occur at **6 PM (hour 18)**, with a peak of **565 trips** during that hour.

How this insight helps Ola/Uber optimize driver allocation and reduce wait times:

- **Pre-positioning Drivers:** Ola/Uber can allocate more drivers on the road before and during 6 PM to meet the surge in demand, reducing passenger wait times significantly.
- **Dynamic Pricing and Incentives:** Surge pricing and driver incentives can be dynamically applied around peak hours to encourage more drivers to be available.
- **Efficient Dispatching:** The platform can optimize ride matching algorithms and dispatch protocols during peak hours to maximize rides completed and minimize idle driver time.
- **Demand Forecasting:** Knowing peak times supports better resource management, including messaging drivers on availability and balancing supply with projected demand.
- **Reducing Traffic Congestion Impact:** Encouraging ride pooling or shared rides during peak traffic hours may improve efficiency and customer satisfaction.

By leveraging this temporal demand pattern, Ola/Uber can enhance operational efficiency while improving rider experience during the busiest times of day.

.

3. Which pickup–drop pair has the highest frequency of rides? How can this information be used for targeted promotions or dynamic pricing?

Ans.:- The pickup-drop pair with the highest frequency of rides is:

- Pickup Location: **Vasant Kunj**
- Drop Location: **AIIMS**
- Number of rides: **21**

How this information can be used for targeted promotions or dynamic pricing:

- **Targeted Promotions:** Ola/Uber can offer special discounts or loyalty rewards for frequent riders traveling between Vasant Kunj and AIIMS to increase customer retention in this high-demand corridor.
- **Dynamic Pricing:** Surge pricing can be intelligently applied during peak demand for this pair to balance supply and demand, incentivizing more drivers to serve this route.
- **Driver Allocation:** Identify and pre-position drivers in areas near Vasant Kunj to quickly respond to ride requests heading to AIIMS, minimizing passenger wait times.
- **Customized Marketing:** Use customer data from this popular route to personalize marketing campaigns and offer bundled ride packages or subscriptions.
- **Service Enhancement:** Consider deploying higher vehicle availability or premium services on this route to capture more market share given demonstrated demand.

This insight supports optimized operational and marketing strategies for capturing value in the most frequently traveled ride corridors.

All the Snapshots, Dataset, Output CSVs, Code have been uploaded on Github:

Github Link:- https://github.com/pranavwagh1072/DEL_40.git