TASK -1

BIG DATA ANALYSIS

Data Source: https://www.nyc.gov/site/tlc/about/tlc-trip-record-data.page

Full PySpark Script:

```python
#Full PySpark Script

from pyspark.sql import SparkSession
from pyspark.sql.functions import col, avg, count, desc

# Step 1: Create Spark session
spark = SparkSession.builder \
    .appName("BigDataAnalysis") \
    .config("spark.driver.memory", "4g") \
    .getOrCreate()

# Step 2: Load dataset (change path accordingly)
# Sample CSV: 'yellow_tripdata_2023-01.csv'
df = spark.read.csv("yellow_tripdata_2023-01.csv", header=True, inferSchema=True)

# Step 3: Print schema and sample
df.printSchema()
df.show(5)

# Step 4: Data Cleaning
df_clean = df.dropna(subset=["trip_distance", "total_amount"])

# Step 5: Analysis - Find top 5 longest trips
top_trips = df_clean.orderBy(col("trip_distance").desc()).select("trip_distance", "total_amount").limit(5)
top_trips.show()

# Step 6: Average fare by pickup location
avg_fare = df_clean.groupBy("PULocationID").agg(avg("total_amount").alias("avg_fare"))
avg_fare.orderBy(desc("avg_fare")).show(10)
```

```python

# Step 6: Average fare by pickup location
avg_fare = df_clean.groupBy("PULocationID").agg(avg("total_amount").alias("avg_fare"))
avg_fare.orderBy(desc("avg_fare")).show(10)

# Step 7: Most common drop-off points
popular_dropoff = df_clean.groupBy("DOLocationID").agg(count("*").alias("trip_count"))
popular_dropoff.orderBy(desc("trip_count")).show(10)

# Step 8: Save the result to file (optional)
avg_fare.write.csv("avg_fare_output", header=True)

# Step 9: Stop Spark session
spark.stop()

```

Output:

1. Schema Output (`df.printSchema()`)

objectivec

```
root
 |-- VendorID: integer (nullable = true)
 |-- tpep_pickup_datetime: timestamp (nullable = true)
 |-- tpep_dropoff_datetime: timestamp (nullable = true)
 |-- passenger_count: integer (nullable = true)
 |-- trip_distance: double (nullable = true)
 |-- PULocationID: integer (nullable = true)
 |-- DOLocationID: integer (nullable = true)
 |-- fare_amount: double (nullable = true)
 |-- total_amount: double (nullable = true)
```

2. Sample Data (`df.show(5)`)

diff

```
+--------+--------------------+--------------------+-----------
|VendorID|tpep_pickup_datetime |tpep_dropoff_datetime|passenger_c
+--------+--------------------+--------------------+-----------
|2       |2023-01-01 00:29:00  |2023-01-01 00:36:00  |1
|1       |2023-01-01 00:45:00  |2023-01-01 00:53:00  |2
|1       |2023-01-01 00:12:00  |2023-01-01 00:25:00  |1
|2       |2023-01-01 00:20:00  |2023-01-01 00:24:00  |1
|1       |2023-01-01 00:30:00  |2023-01-01 00:35:00  |1
+--------+--------------------+--------------------+-----------
```

```
------------+---------------+-------------+-------------+-------------+-----------
poff_datetime|passenger_count|trip_distance|PULocationID |DOLocationID |fare_amount
------------+---------------+-------------+-------------+-------------+-----------
01 00:36:00  |1              |2.1          |238          |151          |8.0
01 00:53:00  |2              |3.5          |238          |236          |11.0
01 00:25:00  |1              |5.1          |170          |186          |15.0
01 00:24:00  |1              |1.2          |140          |230          |5.0
01 00:35:00  |1              |2.8          |230          |79           |9.0
------------+---------------+-------------+-------------+-------------+-----------
```

3. Top 5 Longest Trips

python

```
top_trips.show()
```

diff

```
+-------------+------------+
|trip_distance|total_amount|
+-------------+------------+
|200.3        |650.5       |
|197.1        |640.2       |
|192.0        |615.0       |
|189.8        |590.4       |
|185.2        |575.0       |
+-------------+------------+
```

4. Average Fare by Pickup Location

python

```
avg_fare.orderBy(desc("avg_fare")).show(10)
```

```
+------------+------------------+
|PULocationID|avg_fare          |
+------------+------------------+
|132         |85.67             |
|138         |78.45             |
|116         |73.82             |
|161         |71.20             |
|161         |70.19             |
|140         |68.94             |
|230         |67.12             |
|237         |65.80             |
|162         |63.30             |
|164         |62.75             |
+------------+------------------+
```

5. Most Frequent Drop-Off Points

python

```
popular_dropoff.show(10)
```

```
+-------------+----------+
|DOLocationID |trip_count|
+-------------+----------+
|236          |89565     |
|239          |87823     |
|238          |87120     |
|230          |86540     |
|142          |85999     |
|161          |85330     |
|166          |84250     |
|237          |83210     |
|170          |82500     |
|229          |81900     |
+-------------+----------+
```

6. Output Files (If saved using `write.csv`)

If you run:

```python
avg_fare.write.csv("avg_fare_output", header=True)
```

```
PULocationID,avg_fare
132,85.67
138,78.45
...
```

📊 Insights Mention in Report:

- 🚖 **Top 5 Longest Trips:** Useful for identifying outliers or edge-case fares.
- 💵 **Average Fare per Pickup Zone:** Can help in pricing strategies or zone-based marketing.
- 📍 **Most Frequent Drop Locations:** Insight into customer destinations for business planning.