BIG DATA ANALYSIS

pip install pyspark

OUTPUT:

Requirement already satisfied: pyspark in /usr/local/lib/python3.11/dist-packages (3.5.1)

Requirement already satisfied: py4j==0.10.9.7 in /usr/local/lib/python3.11/dist-packages (from pyspark) (0.10.9.7)

NYC Yellow Taxi Trip Data - Big Data Analysis using PySpark

from pyspark.sql import SparkSession

from pyspark.sql.functions import col, hour, dayofweek, avg, count, sum as spark_sum, max as spark_max, min as spark_min

Step 1: Start Spark session

spark = SparkSession.builder.appName("NYC Yellow Taxi Analysis").getOrCreate()

Step 2: Load the dataset

file_path = "/content/NYC Yellow Taxi Trip Data.csv"

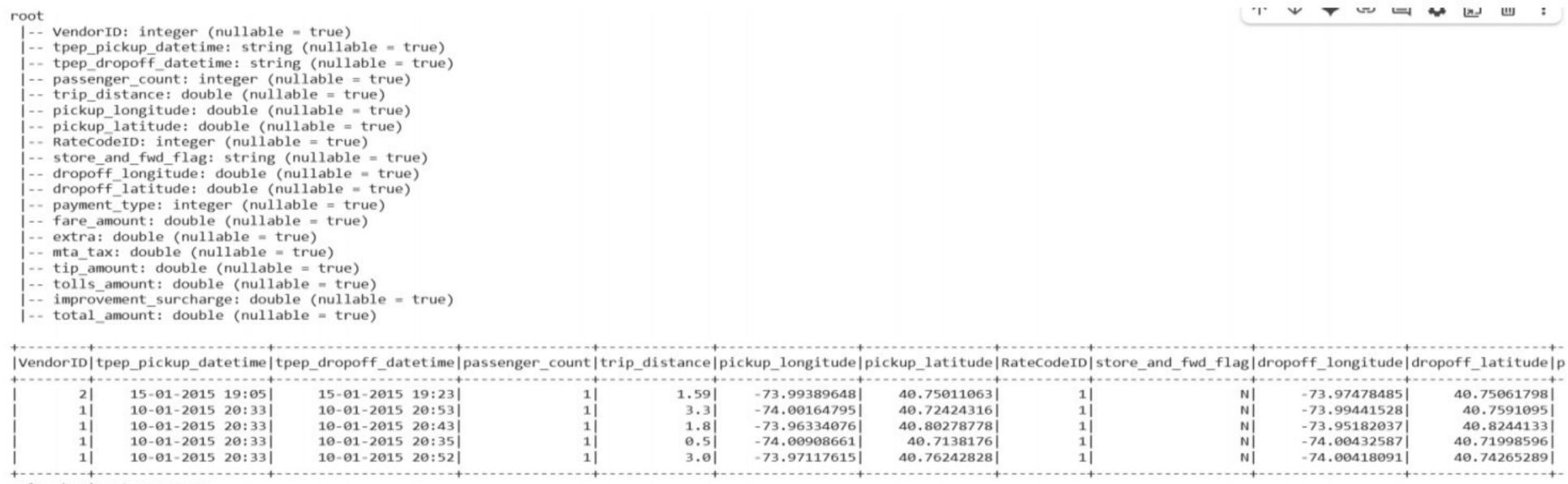
df = spark.read.csv(file_path, header=True, inferSchema=True)

Step 3: Inspect schema and data

df.printSchema()

df.show(5)

OUTPUT:



only showing top 5 rows

Step 4: Data Cleaning - Remove nulls and invalid values

```
df_clean = df.dropna().filter(
    (col("passenger_count") > 0) &
    (col("trip_distance") > 0) &
    (col("fare_amount") > 0)
)
```

Step 6: Analysis 1 - Avg Fare by Hour

```
avg_fare_by_hour =
df_clean.groupBy("hour").agg(avg("fare_amount").alias("avg_fare")).orderBy("hour")
avg_fare_by_hour.show()
```

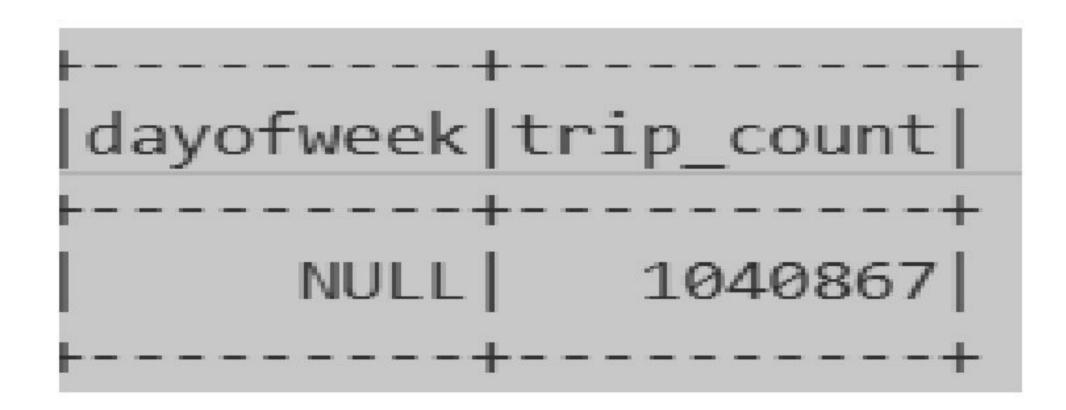
OUTPUT:



Step 7: Analysis 2 - Total Trips by Day of Week

```
trips_by_day =
df_clean.groupBy("dayofweek").agg(count("*").alias("trip_count")).orderBy("dayofweek")
trips_by_day.show()
```

OUTPUT:



Step 8: Analysis 3 - Top 5 longest trips

```
longest_trips = df_clean.orderBy(col("trip_distance").desc()).select(
    "pickup_datetime", "trip_distance", "fare_amount", "passenger_count"
).limit(5)
longest_trips.show()
```

OUTPUT:

pickup_datetime	trip distance	 fare amount	passenger count
+		· +	+
NULL	92000.9	2.5	1
NULL	30083.2	14.0	1
NULL	801.0	12.5	1
NULL	400.2	9.0	1
NULL	400.1	21.0	1
++			+

Step 9: Summary Stats

```
summary = df_clean.select(
    spark_max("trip_distance").alias("Max Distance"),
    spark_min("trip_distance").alias("Min Distance"),
```

```
spark_max("fare_amount").alias("Max Fare"),
spark_min("fare_amount").alias("Min Fare"),
avg("fare_amount").alias("Avg Fare")
)
summary.show()
```

OUTPUT:

Max Distance	H Min Distance	Max Fare	Min Fare	Avg Fare
92000.9	0.01	450.0	0.01	11.83750700137524

Stop Spark session

spark.stop()