**Yellow Taxi Data**

The schema for the yellow taxi types is shown below:

tpep\_pickup\_datetime: string (nullable = true)

tpep\_dropoff\_datetime: string (nullable = true)

passenger\_count: string (nullable = true)

trip\_distance: string (nullable = true)

PULocationID: string (nullable = true)

DOLocationID: string (nullable = true)

payment\_type: string (nullable = true)

fare\_amount: string (nullable = true)

extra: string (nullable = true)

mta\_tax: string (nullable = true)

tip\_amount: string (nullable = true)

tolls\_amount: string (nullable = true)

total\_amount: string (nullable = true)

congestion\_surcharge: string (nullable = true)

airport\_fee: string (nullable = true)

taxi\_type: string (nullable = true)

The description and sample of the yellow taxi data is shown below

tpep\_pickup\_datetime: The date and time when the meter was engaged

tpep\_dropoff\_datetime: The date and time when the meter was disengaged

passenger\_count: The number of passengers in the vehicle

trip\_distance: The elapsed trip distance in miles reported by the taximeter

PULocationID: Taxi Zone in which the taximeter was engaged

DOLocationID: Taxi Zone in which the taximeter was disengaged

payment\_type: A numeric code signifying how the passenger paid for the trip. 1= Credit card 2= Cash 3= No charge 4= Dispute 5= Unknown 6= Voided trip

fare\_amount: The time-and-distance fare calculated by the meter

extra: Miscellaneous extras and surcharges. Currently, this only includes the $0.50 and $1 rush hour and overnight charges.

mta\_tax: $0.50 MTA tax that is automatically triggered based on the metered rate in use

tip\_amount: $0.50 MTA tax that is automatically triggered based on the metered rate in use

tolls\_amount: $0.50 MTA tax that is automatically triggered based on the metered rate in use

total\_amount: $0.50 MTA tax that is automatically triggered based on the metered rate in use

congestion\_surcharge: $0.50 MTA tax that is automatically triggered based on the metered rate in use

airport\_fee: $0.50 MTA tax that is automatically triggered based on the metered rate in use

taxi\_type: yellow taxi

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

Green Taxi Data

The schema for the green taxi is shown below:

lpep\_pickup\_datetime: string (nullable = true)

lpep\_dropoff\_datetime: string (nullable = true)

PULocationID: string (nullable = true)

DOLocationID: string (nullable = true)

passenger\_count: string (nullable = true)

trip\_distance: string (nullable = true)

fare\_amount: string (nullable = true)

extra: string (nullable = true)

mta\_tax: string (nullable = true)

tip\_amount: string (nullable = true)

tolls\_amount: string (nullable = true)

ehail\_fee: string (nullable = true)

total\_amount: string (nullable = true)

payment\_type: string (nullable = true)

trip\_type: string (nullable = true)

congestion\_surcharge: string (nullable = true)

taxi\_type: string (nullable = true)

The description and sample of the green taxi data are shown below (Note there is an extra useless column named ehail\_fee in this dataset\*\*)

lpep\_pickup\_datetime: The date and time when the meter was engaged.

lpep\_dropoff\_datetime: The date and time when the meter was disengaged.

PULocationID: Taxi Zone in which the taximeter was engaged

DOLocationID: Taxi Zone in which the taximeter was engaged

passenger\_count: The number of passengers in the vehicl

trip\_distance: The elapsed trip distance in miles reported by the taximeter

fare\_amount: The time-and-distance fare calculated by the meter

extra: Miscellaneous extras and surcharges. Currently, this only includes the $0.50 and $1 rush hour and overnight charges

mta\_tax: $0.50 MTA tax that is automatically triggered based on the metered rate in use.

tip\_amount: Tip amount – This field is automatically populated for credit card tips. Cash tips are not included.

tolls\_amount: Total amount of all tolls paid in trip

ehail\_fee: Dont care column

total\_amount: The total amount charged to passengers. It does not include cash tips

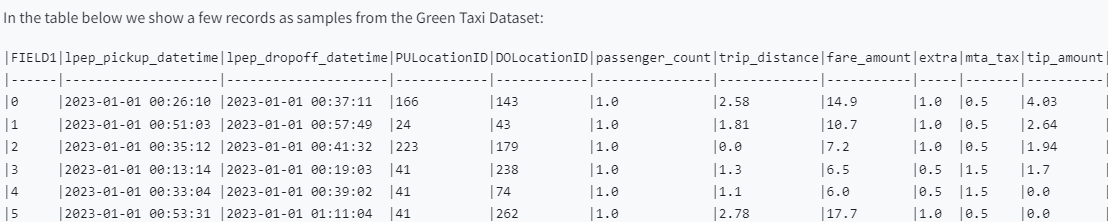
payment\_type: A numeric code signifying how the passenger paid for the trip. 1= Credit card 2= Cash 3= No charge 4= Dispute 5= Unknown 6= Voided trip

trip\_type: A code indicating whether the trip was a street hail or a dispatch that is automatically assigned based on the metered rate in use but can be altered by the driver

congestion\_surcharge: Total amount collected in trip for NYS congestion surcharge

taxi\_type: Green Taxi

In the table below we show a few records as samples from the Green Taxi Dataset:



A screenshot of a computer

Description automatically generated

**Taxi Zone Lookup Table**

The lookup table Taxi Zone Lookup Table has the details for each LocationID (PULocationID or DOLocationID) in the above tables. This table has the following schema:

LocationID: integer (nullable = true)

Borough: string (nullable = true)

Zone: string (nullable = true)

service\_zone: string (nullable = true)

You will use this table to extract pickup/dropoff (PULocationID/DOLocationID) location details for later tasks in this assignment. In the below table, we show a few records as samples from the Taxi Zone lookup table:

**A screenshot of a computer screen

Description automatically generated**

What I would like from you

Submit a zip file with well-commented and organized code for each task and also include scripts and CSV/TXT files used for compiling visualizations, such as graphs or charts, to represent your findings in tasks that require doing so

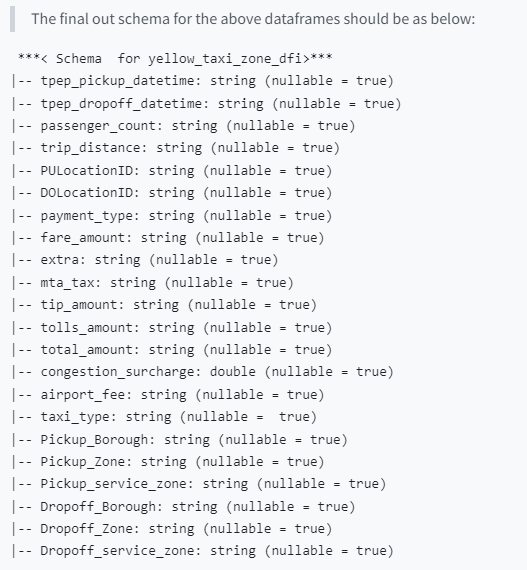
Submit a single PDF report detailing your approach to solving each task along with the visualisation of the results and discuss challenges faced and insights gained from each task.

Task 1

Create the data frame that combines all the months-wise csv files single data frame each for yellow and green trip data. Rename them as yellow\_tripdata\_df and green\_tripdata\_df

Using the above yellow\_tripdata\_df and green\_tripdata\_df data frames, apply the join method based on fields PULocation and DOLocationID into the taxi\_\_zone\_lookup.csv table to the extract pickup and dropoff locations information and rename the those columns ad Pickup\_Borough, Pickup\_Zone, Pickup\_service\_zone , Dropoff\_Borough, Dropoff\_Zone, Dropoff\_service\_zone

At the end, you will have two dataframes (as shown below) with six new columns added to the original dataset. 1. yellow\_taxi\_zone\_df 2. green\_taxi\_zone\_df



A screenshot of a computer code

Description automatically generated

After performing the above operation, write the number of rows and columns for both dataframes. Verify that their schemas match the above resulting schemas. You need to add in the report a screenshot of your command line of the resulting scheme using printschema API.

Task 2

Using the dataframe generated from task 1 to complete the below task.

Illustrate the pickup locations in each NYC borough for yellow taxis, with boroughs on the x-axis and pickup counts on the y-axis.

Display the number of pickup locations in each NYC borough for green taxis. Plot the boroughs on the x-axis and the pickup counts on the y-axis.

Depict the dropoff locations in each NYC borough for yellow taxis. The graph should have the boroughs on the x-axis and the dropoff counts on the y-axis.

Show the number of dropoff locations in each NYC borough for green taxis. On the graph, the boroughs should be on the x-axis with dropoff counts on the y-axis. (Note: There will be total of 4 plots for this task)

Task 3

For the yellow taxi dataset, filter out trips that have a fare greater than $50 but a distance of less than 1 mile for the first week of the year 2023 and plot the count of trips on each day of the week. The plot should have days along the x-axis and number of trips along the y-axis.

Hint: - you need to create a UDF that extracts and filters the required information and passes it to the map function. The UDF would extract the fare, distance, and pickup date fields and then apply the filter to exclude the records out of the requested time window (similar to clean\_lines function). - Note you could use the following functions to extract the timestamps from the string using timestamp = datetime.strptime(datefield, "%Y-%m-%d %H:%M:%S") and then you can extract the date from that timestamp using timestamp.strftime('%Y-%m-%d')

Task 4

Understand spatial patterns of taxi pickups and drop-offs in boroughs - For both the yellow and green taxi datasets Identify the top 5 most popular pickup and drop-off boroughs. Visualize on two plots (one for yellow and one for green) by plotting the number of trips on the y-axis against these boroughs on the x-axis.

Task 5

Analyze taxi trips over time to identify patterns. Identify any patterns or anomalies, such as specific days with unusually high or low trip counts. The plot can be scattered or line plot having dates along the x-axis and anomalies (number of trips) along the y-axis. For this task use only the Yellow Taxi dataset for the month of June 2023.

Hint: - Unusually high or low trip counts can be found by getting the mean\_trips which is the sum of total trips counts divided by the total number of total trips. - if the total trip counts for each day are greater than 80% can be considered as a high anomaly and below 20% a low anomaly

Task 6

Calculate the average fare per mile from trips. Identify outlier trips that have an unusually high fare per mile, using visualization. You will use a scatter plot to show the fare per mile and on the plot draw a line of the average fare per mile. Describe any outlier you see from the plot. For this task use only the Yellow Taxi dataset for the month of March 2023.

• Hint: Use the fare and distance columns to compute the fare per mile for each trip. Also, for scatter plot creating a sequence of length starting from 0 to len(fare\_per\_mile) for the x-axis and on the y-axis use the fare per mile

Task 7

Analyze the distribution of the number of passengers per trip. Are solo trips more common or do people often share taxis, give the answer in percentage for solo trips for both the yellow and green taxi datasets.

• Hint: Use the total trip count for the percentage

Task 8

Examine the correlation between the trip duration and distance travelled of the trips in the yellow taxi dataset. These could be indicative of traffic jams, route detours, or other anomalies. Create a plot having duration in hours along the x-axis and trip distance along the y-axis. For this task use only the Yellow Taxi dataset for the month of Jan 2023.

Task 9

Get the 10 most active boroughs (based on total pickups) where both green and yellow taxis operate and find which type of taxi (green vs yellow) for each of the top 10 boroughs had been used for the most trips over all the months. Visualize the output as colour-coded (yellow and green) bar plot for the boroughs with the y-axis representing the total pickups.

• Hint: you need to extract the count of the trips for the boroughs and then find out the top 10

Task 10

Identify the month where the most trips were recorded for both the yellow and the green taxi datasets. Plot the output as a bar plot (one bar for yellow and one for green) having the month name on the x-axis and the trip count along the y-axis.