# **NYC Rideshare Analysis**

In the Coursework, you will apply **Spark** techniques to the **NYC Rideshare dataset**, which focuses on analyzing the New York 'Uber/Lyft' data from January 1, 2023, to May 31, 2023. Source data pre-processed was provided by the NYC Taxi and Limousine Commission (TLC) Trip Record Data hosted by the state of New York. The dataset used in the Coursework is distributed under the MIT license. The source of the datasets is available on this [link](https://www.kaggle.com/datasets/aaronweymouth/nyc-rideshare-raw-data?resource=download)

Useful Resources: Lectures, Labs, and other materials are shared along with the following links:

<https://sparkbyexamples.com/pyspark>

<https://sparkbyexamples.com/pyspark-tutorial/>

<https://spark.apache.org/docs/3.1.2/api/python/getting_started/index.html>

<https://www.nyc.gov/site/tlc/about/tlc-trip-record-data.page>

You can see two CSV files under the path **//data-repository-bkt/ECS765/rideshare\_2023/**, including **rideshare\_data.csv** and **taxi\_zone\_lookup.csv**. Using the ccc method bucket ls command to check them.

## **Dataset Schema**

The information on the two CSV files is described below. Please read them carefully because it will help you understand the dataset and tasks.

### **rideshare\_data.csv**

The data information for the **rideshare\_data.csv** is shown below ( **s means second** ):

| Field | Type | Description |

|---------------------|-----------|----------------------------------------------------------------------------------------------------------|

| business | string | Uber and Lyft. |

| pickup\_location | string | Taxi Zone where the journey commenced. Refer to 'taxi\_zone\_lookup.csv' for details. |

| dropoff\_location | string | Taxi Zone where the journey concluded. Refer to 'taxi\_zone\_lookup.csv' for details. |

| trip\_length | string | The total distance of the trip in miles. |

| request\_to\_pickup | string (s)| The time taken from the ride request to the passenger pickup. |

| total\_ride\_time | string (s)| The duration between the passenger pickup and dropoff, indicating the total time spent in the car. |

| on\_scene\_to\_pickup | string (s)| The time duration between the driver's arrival on the scene and the passenger pickup, reflecting wait time. |

| on\_scene\_to\_dropoff | string (s)| Time from the driver's arrival on the scene to the passenger dropoff, indicating the driver's total time commitment. |

| time\_of\_day | string | Categorization of the time of day: morning (0600-1100), afternoon (1200-1600), evening (1700-1900), night (other times). |

| date | string (s)| The date when the ride was requested, expressed in UNIX timestamp. |

| passenger\_fare | string | The total fare paid by the passenger in USD, inclusive of all charges. |

| driver\_total\_pay | string | The complete payment received by the driver, including base pay and tips. |

| rideshare\_profit | string | The difference between the passenger fare and the driver's total pay, representing the platform's profit.|

| hourly\_rate | string | The calculated hourly rate based on 'on\_scene\_hours', including the duration from arrival to final drop-off. |

| dollars\_per\_mile | string | The driver's earnings per mile, calculated as total pay divided by trip length. |

The table below shows the samples from the rideshare\_data.csv:

| business | pickup\_location | dropoff\_location | trip\_length | request\_to\_pickup | total\_ride\_time | on\_scene\_to\_pickup | on\_scene\_to\_dropoff | time\_of\_day | date | passenger\_fare | driver\_total\_pay | rideshare\_profit | hourly\_rate | dollars\_per\_mile |

|:-----------|------------------:|-------------------:|--------------:|--------------------:|------------------:|---------------------:|----------------------:|:--------------|-----------:|-----------------:|-------------------:|-------------------:|--------------:|-------------------:|

| Uber | 102 | 157 | 2.08 | 119 | 648 | 79 | 727 | evening | 1684713600 | 12.61 | 9.1 | 3.51 | 45.06 | 4.38 |

| Uber | 235 | 244 | 1.7 | 59 | 1226 | 27 | 1253 | evening | 1684713600 | 13.57 | 13.75 | -0.18 | 39.51 | 8.09 |

| Uber | 79 | 170 | 2.02 | 438 | 835 | 31 | 866 | morning | 1684800000 | 26.26 | 21.14 | 5.12 | 87.88 | 10.47 |

| Uber | 256 | 196 | 6.27 | 180 | 1362 | 26 | 1388 | morning | 1684713600 | 31.21 | 25.04 | 10.17 | 64.95 | 3.99 |

| Uber | 230 | 43 | 1.63 | 72 | 637 | 65 | 702 | morning | 1684713600 | 24.44 | 13.13 | 16.31 | 67.33 | 8.06 |

### **taxi\_zone\_lookup.csv**

The [taxi\_zone\_lookup.csv](https://d37ci6vzurychx.cloudfront.net/misc/taxi+_zone_lookup.csv) has the details for each **pickup\_location/dropoff\_location** of the **rideshare\_data.csv**. **taxi\_zone\_lookup.csv** has the following schema:

LocationID: string (nullable = true)  
 Borough: string (nullable = true)  
 Zone: string (nullable = true)  
 service\_zone: string (nullable = true)

The table below shows the samples from the taxi\_zone\_lookup.csv:

|LocationID|Borough |Zone |service\_zone|

|----------|-------------------|-----------------------|------------|

|1 |EWR |Newark Airport |EWR |

|2 |Queens |Jamaica Bay |Boro Zone |

|3 |Bronx |Allerton/Pelham Gardens|Boro Zone |

|4 |Manhattan |Alphabet City |Yellow Zone |

|5 |Staten Island |Arden Heights |Boro Zone |

As you can see, **the pickup\_location/dropoff\_location** fields in **rideshare\_data.csv** are encoded with numbers that you can find the counterpart number (LocationID field) in **taxi\_zone\_lookup.csv**. You need to join the two datasets by using the mentioned fields. Before you go to the assignment part, there are three notes you need to understand about the taxi\_zone\_lookup.csv, (1): The LocationID 264 and 265 are \*\*Unknown\*\* in the Borough field, we see the 'Unknown' as one of the borough names; (2) In the Borough field, you can see the same Borough has different LocationIDs, it does not matter because you use LocationID (Unique Key) to apply 'Join'function; and (3) if you see any obscure description (like NV, NA, N/A, etc) in any fields, purely see it as the valid names.

## **Assignment**

**WARNING: DO NOT ATTEMPT TO COPY FROM YOUR CLASSMATES OR ANY OTHER MATERIALS. THE SUBMISSIONS WILL GO THROUGH PLAGIARISM CHECKS AND SUBMISSIONS FOUND IN VIOLATION WILL RESULT IN MISCONDUCT CASES (**[**https://arcs.qmul.ac.uk/students/student-appeals/academic-misconduct/**](https://arcs.qmul.ac.uk/students/student-appeals/academic-misconduct/)**). WE ALSO MAY ASK YOU FOR A LIVE DEMONSTRATION OF THE PROGRAM AND RESULTS OBTAINED**

**Note** 1. Write SPARK scripts to answer the questions, and you are allowed to use any Spark functions/API not limited to ones you learned in the module, any solutions without using SPARK scripts are invalid.  
 2. For drawing Graphs, you need to download your outputs, and then you can use any visualization toolkit; Python's matplotlib (<https://matplotlib.org/stable/users/index>), Gnuplot (<http://www.gnuplot.info>), or any plotting tool of your preference. Note that Matplotlib is readily available in your **local Jhub environment**, and you cannot run matplotlib or gnuplot on the **spark cluster**. The method of plotting should be clear. If no script is used, the plotting method should be reproducible and all steps to reproduce need to be described in detail in the report.  
 3. You are free to choose any combination of the below tasks (or any subtasks) that can sum up to 100% (the grade is capped at 100%). You are free to attempt all or as many, however, the score will be still capped at 100% even if your mark is more than 100%.  
 4. The grade of the coursework will account for 30% of the module's total grade.  
 5. You might need to convert when needed the fields from string type to another appropriate (e.g., integer or float) type if math operations are involved.

**Submission Guidelines**

There are two separate files you need to submit, (1) a PDF file, and (2) a Zip file.  
 1. Submit a single **PDF report**, including (1) detailed explanations for each step and used API to solve each task, (2) the visualization of your results (Graphs or Screenshots), (3) expressing challenges you encountered for each task, and how you overcome the challenges, and (4) what knowledge/insight attained from each task. **This should not be included in the zip file**.  
 2. Submit a **zip file**, including (1) a well-commented and organized Spark script for each task, (2) your output results (Screenshot or Data file containing data points only used for plotting), and (3) scripts used for visualizations.  
 3. Missing or obscuring the requested content for the report or Spark script files will result in deductions. Not submitting any of the above two files (Report PDF and Supplementary Zip) may result in your coursework not being marked or severe deductions.

**Evaluation Criteria:** 1. Quality, clarity, and details of the report.  
 2. Correctness of data analysis approach and results (e.g., the visualisations).  
 3. Correctness of your Spark implementation.

## **Tasks**

**Note** 1. We have attached the **starterkit.py** script which has the common configuration code in the Sprak job.  
 2. We have attached the **graph.py** script which helps the start of task 8.  
 3. We have attached a mini sample of rideshare datasets called **sample\_data.csv** that you can use for debugging and testing purposes, however, **your results of the assignment tasks must use original datasets under path //data-repository-bkt/ECS765/rideshare\_2023/.** 4. We have attached the **taxi\_zone\_lookup.csv**.  
 5. The tables we provided for your reference in some tasks aim to tell you the expected format. You can use the same name as we provided, or you can name the columns to your preference. However, the column's name should be clear and understandable.  
 6. The sample data possibly does not include all boroughs or dates because it is randomly sampled from the original dataset.

Below are the tasks with different weights. Please read each task carefully, and give the complete Spark solutions for getting full marks. Also, **task 1 is very important as some of the remaining tasks might use task 1 results.**

### **Task 1 (15 points): Merging Datasets**

1. (2 points) Load rideshare\_data.csv and taxi\_zone\_lookup.csv.
2. (5 points) Apply the 'join' function based on fields **pickup\_location** and **dropoff\_location** of rideshare\_data table and the **LocationID** field of taxi\_zone\_lookup table, and rename those columns as **Pickup\_Borough, Pickup\_Zone, Pickup\_service\_zone , Dropoff\_Borough, Dropoff\_Zone, Dropoff\_service\_zone**. The join needs to be done in two steps. once using pickup\_location and then output result is joined using dropoff\_location. you will have a new dataframe (as shown below) with six new columns added to the original dataset.  
    |-- business: string (nullable = true)  
    |-- pickup\_location: string (nullable = true)  
    |-- dropoff\_location: string (nullable = true)  
    |-- trip\_length: string (nullable = true)  
    |-- request\_to\_pickup: string (nullable = true)  
    |-- total\_ride\_time: string (nullable = true)  
    |-- on\_scene\_to\_pickup: string (nullable = true)  
    |-- on\_scene\_to\_dropoff: string (nullable = true)  
    |-- time\_of\_day: string (nullable = true)  
    |-- date: string (nullable = true)  
    |-- passenger\_fare: string (nullable = true)  
    |-- driver\_total\_pay: string (nullable = true)  
    |-- rideshare\_profit: string (nullable = true)  
    |-- hourly\_rate: string (nullable = true)  
    |-- dollars\_per\_mile: string (nullable = true)  
    |-- Pickup\_Borough: string (nullable = true)  
    |-- Pickup\_Zone: string (nullable = true)  
    |-- Pickup\_service\_zone: string (nullable = true)  
    |-- Dropoff\_Borough: string (nullable = true)  
    |-- Dropoff\_Zone: string (nullable = true)  
    |-- Dropoff\_service\_zone: string (nullable = true)
3. (5 points) The date field uses a UNIX timestamp, you need to **convert the UNIX timestamp to the "yyyy-MM-dd" format**. For example, '1684713600' to '2023-05-22'. UNIX timestamp represents the number of seconds elapsed since January 1, 1970, UTC. However, in this dataframe, the UNIX timestamp is converted from (yyyy-MM-dd) without the specific time of day (hh-mm-ss). For example, the '1684713600' is converted from '2023-05-22'.
4. (3 points) After performing the above operations, print the number of rows (69725864) and schema of the new dataframe in the terminal. Verify that your schema matches the above resulting schemas. You need to provide the screenshots of your scheme and the number of rows in your report.

### **Task 2 (20 points): Aggregation of Data**

1. (6 points) Count the number of trips for each business in each month. Draw the histogram with 'business-month' on the x-axis and trip counts on the y-axis. For example, assume if the number of trips for Uber in January (i.e. 'Uber-1') is 222222222, 'Uber-1' will be on the x-axis, indicating 222222222 above the bar of 'Uber-1'.
2. (6 points) Calculate the platform's profits (rideshare\_profit field) for each business in each month. Draw the histogram with 'business-month' on the x-axis and the platform's profits on the y-axis. For example, assume if the platform's profits for Uber in January is 33333333, 'Uber-1' will be on the x-axis, indicating 33333333 above the bar of 'Uber-1'.
3. (5 points) Calculate the driver's earnings (driver\_total\_pay field) for each business in each month. Draw the histogram with 'business-month' on the x-axis and the driver's earnings on the y-axis. For example, assume if the driver's earnings for Uber in January is 4444444, 'Uber-1' will be on the x-axis, indicating 4444444 above the bar of 'Uber-1'.
4. (3 points) When we are analyzing data, it's not just about getting results, but also about extracting insights to make decisions or understand the market. Suppose you were one of the stakeholders, for example, the driver, CEO of the business, stockbroker, etc, What do you find from the three results? How do the findings help you make strategies or make decisions?  
    **Note, that the above figures/values do not represent the actual result.**

### **Task 3 (25 points): Top-K Processing**

(7 points) Identify the top 5 popular pickup boroughs each month. you need to provide a screenshot of your result in your report. The columns should include, Pickup\_Borough, Month, and trip\_count. you need to sort the output by trip\_count by descending in each month. For example,  
  
 | Pickup\_Borough | Month | trip\_count |

|----------------|-------|------------|

| Manhattan | 1 | 5 |

| Brooklyn | 1 | 4 |

| Queens | 1 | 3 |

| Bronx | 1 | 2 |

| Staten Island | 1 | 1 |

| ... | ... | ... |

1. **Note, that the above figures/values of the trip\_count field do not represent the actual result.**

(7 points) Identify the top 5 popular dropoff boroughs each month. you need to provide a screenshot of your result in your report. The columns should include, Dropoff\_Borough, Month, and trip\_count. you need to sort the output by trip count by descending in each month. For example,  
  
 | Dropoff\_Borough | Month | trip\_count |

|-----------------|-------|------------|

| Manhattan | 2 | 5 |

| Brooklyn | 2 | 4 |

| Queens | 2 | 3 |

| Bronx | 2 | 2 |

| Staten Island | 2 | 1 |

| ... | ... | ... |

1. **Note, that the above figures/values of the trip\_count field do not represent the actual result.**

(8 points) Identify the top 30 earnest routes. Use 'Pickup Borough' to 'Dropoff\_Borough' as route, and use the sum of 'driver\_total\_pay' field as the profit, then you will have a route and total\_profit relationship. The columns should include Route and total\_profit. You need to provide a screenshot of your results in your report. **Do not truncate the name of routes**. For example,  
  
 | Route | total\_profit |

|----------------------|--------------|

| Queens to Queens | 222 |

| Brooklyn to Queens | 111 |

| ... | ... |

1. **Note, that the above figures/values of the total\_profit field do not represent the actual result.**
2. (3 points) Suppose you were one of the stakeholders, for example, either the driver, CEO of the business, or stockbroker, etc, What do you find (i.e., insights) from the previous three results? How do the findings help you make strategies or make decisions?

### **Task 4 (15 points): Average of Data**

(4 points) Calculate the average 'driver\_total\_pay' during different 'time\_of\_day' periods to find out which 'time\_of\_day' has the highest average 'driver\_total\_pay'. You need to provide a screenshot of this question in your report. The columns should include, time\_of\_day, average\_drive\_total\_pay. You need to sort the output by average\_drive\_total\_pay by descending. For example,  
  
 | time\_of\_day | average\_drive\_total\_pay |

|-------------|-------------------------|

| afternoon | 25 |

| night | 22 |

| evening | 20 |

| morning | 18 |

1. **Note, that the above figures/values for average\_driver\_total\_pay do not represent the actual result.**

(4 points) Calculate the average 'trip\_length' during different time\_of\_day periods to find out which 'time\_of\_day' has the highest average 'trip\_length'. You need to provide a screenshot of this question in your report. The columns should include, time\_of\_day, average\_trip\_length. You need to sort the output by average\_trip\_length by descending. For example,  
  
 | time\_of\_day | average\_trip\_length |

|-------------|--------------------|

| night | 25 |

| morning | 22 |

| afternoon | 20 |

| evening | 18 |

1. **Note, that the above figures/values for average\_trip\_length do not represent the actual result.**

(5 points) Use the above two results to calculate the average earned per mile for each time\_of\_day period. You need to use 'join' function first and use average\_drive\_total\_pay divided by average\_trip\_length to get the average\_earning\_per\_mile. You need to provide a screenshot of this question in your report. The columns should include, time\_of\_day, and average\_earning\_per\_mile. For example,  
  
 | time\_of\_day | average\_earning\_per\_mile |

|-------------|-------------------------|

| night | 7 |

| morning | 5 |

| afternoon | 6 |

| evening | 8 |

1. **Note, that the above figures/values for average\_earning\_per\_mile do not represent the actual result.**
2. (2 points) What do you find (i.e., insights) from the three results? How do the findings help you make strategies or make decisions?

### **Task 5 (15 points): Finding anomalies**

1. (10 points) Extract the data in January and calculate the average waiting time (use the "request\_to\_pickup" field) over time. You need to sort the output by day. Draw the histogram with 'days' on the x-axis and ‘average waiting time’ on the y-axis. For example, assume that the ‘average waiting time on 'January 10' is '999', '10' will be on the x-axis, indicating the average waiting time value of 999 for day 10 in January.
2. (3 points) Which day(s) does the average waiting time exceed 300 seconds?
3. (2 points) Why was the average waiting time longer on these day(s) compared to other days?

### **Task 6 (15 points): Filtering Data**

(5 points) Find trip counts greater than 0 and less than 1000 for different 'Pickup\_Borough' at different 'time\_of\_day'. You need to provide a screenshot of this question in your report. The columns should include, Pickup\_Borough, time\_of\_day, and trip\_count. For example,  
  
 | Pickup\_Borough | time\_of\_day | trip\_count |

|----------------|-------------|------------|

| EWR | afternoon | 8 |

| EWR | morning | 8 |

| ... | ... | ... |

1. **Note, that the above figures/values for trip\_count do not represent the actual result.**

(5 points) Calculate the number of trips for each 'Pickup\_Borough' in the **evening time** (i.e., time\_of\_day field). You need to provide a screenshot of this question in your report. The columns should include, Pickup\_Borough, time\_of\_day, trip\_count. For example,  
  
 | Pickup\_Borough | time\_of\_day | trip\_count |

|----------------|-------------|------------|

| EWR | evening | 23333 |

| Unknown | evening | 2222 |

| ... | ... | ... |

1. **Note, that the above figures/values for trip\_count do not represent the actual result.**

(5 points) Calculate the number of trips that started in Brooklyn (Pickup\_Borough field) and ended in Staten Island (Dropoff\_Borough field). Show 10 samples in the terminal. You need to provide a screenshot of this question (the 10 samples) and the number of trips in your report. The columns should include, Pickup\_Borough, Dropoff\_Borough, and Pickup\_Zone. **do not truncate the name of Pickup\_Zone**. For example,  
  
 | Pickup\_Borough | Pickup\_Borough | Pickup\_Zone |

|----------------|------------------|-------------|

| Brooklyn | Staten Island | Bay Ridge |

| ... | ... | ... |

1. **Note, that the above record is just one of my results, you may or may not have the same record.**

### **Task 7 (15 points): Routes Analysis**

1. (15 points) You need to analyse the 'Pickup\_Zone' to 'Dropoff\_Zone' routes to find the top 10 popular routes in terms of the trip count. Each total count of a route should include the trip counts for each unique route from Uber and Lyft (i.e., the sum of Uber and Lyft counts on the same route). Then you can get the top 10 in total count (e.g., sorting the result by total counts and displaying the top 10 routes or any other way). You may need to create a new column called Route which concats column Pickup\_Zone with 'to' with column Dropoff\_Zone. You need to give a screenshot of your result in your report, **do not truncate the name of routes**. The result table should include 'route', 'uber\_count', 'lyft\_count', and 'total\_count'. For example,

| Route | uber\_count | lyft\_count | total\_count |

|----------------------|------------|------------|-------------|

| JFK Airport to NA | 253213 | 45 | 253258 |

| Astoria to Jamaica | 253212 | 12 | 253224 |

| ...... | ...... | ...... | ...... |

**Note, that the above figures/values for uber\_count, lyft\_count, and total\_countdo not represent the actual result.**

### **OPTIONAL - Task 8 (20 points): Graph Processing**

**Note, the tables provided in task 8 are from my results, you may or may not have the same results depending on how you process and show results.**

1. (2 points) define the StructType of vertexSchema and edgeSchema. Use the **taxi\_zone\_lookup.csv** as vertex information and the 'pickup\_location' field and the 'dropoff\_location' field of **rideshare\_data.csv** as 'src' and 'dst' information.
2. (4 points) construct edges dataframe, and vertices dataframe. Show 10 samples of the edges dataframe and vertices dataframe. You need to give a screenshot of your results in your report. **Do not truncate the name of fields**. The vertices table should include 'id', 'Borough', 'Zone', and 'service\_zone'. The edges table should include 'src', 'dst'. For example,

| id | Borough | Zone | service\_zone |

|----|---------|---------------------------|--------------|

| 1 | EWR | Newark Airport | EWR |

| 2 | QUEENS | Jamaica Bay | Boro Zone |

| 3 | Bronx | Allerton/Pelham Gardens | Boro Zone |

| ...| ... | ... | ... |

| src | dst |

|-----|-----|

| 151 | 244 |

| 244 | 78 |

| 151 | 138 |

| ... | ... |

1. (4 points) Create a graph using the vertices and edges. Print 10 samples of the graph DataFrame with columns ‘src’, ‘edge’, and ‘dst’. You need to give a screenshot of your results in your report. For example,

| src | edge | dst |

|---------------------------------------------------------|-------------------|--------------------------------------------------------|

| [151, Manhattan, Manhattan Valley, Yellow Zone] | [151, 244] | [244, Manhattan, Washington Heights South, Boro Zone] |

| [244, Manhattan, Washington Heights South, Boro Zone] | [244, 78] | [78, Bronx, East Tremont, Boro Zone] |

| ... | ... | ... |

1. (5 points) count connected vertices with the same Borough and same service\_zone. And, select 10 samples from your result. The table colums should include 'id'(src), 'id'(dst), 'Borough', and 'service\_zone'. You need to give a screenshot of your 10 samples and the number of counts in your report. For example,

| id | id | Borough | service\_zone |

|-----|-----|---------|--------------|

| 182 | 242 | Bronx | Boro Zone |

| 248 | 242 | Bronx | Boro Zone |

| 242 | 20 | Bronx | Boro Zone |

| 20 | 20 | Bronx | Boro Zone |

| ... | ... | ... | ... |

1. (5 points) perform page ranking on the graph dataframe. You will set the 'resetProbability' to 0.17 and 'tol' to 0.01. And sort vertices by descending according to the value of PageRank. 'Show the 5 samples of your results. You need to give a screenshot of your results in your report. The table columns should include 'id', 'pagerank'. For example:

| id | pagerank |

|-----|---------------------|

| 265 | 11.105433344108409 |

| 1 | 5.4718454249206525 |

| ... | ... |

# 