Big Data Engineering

Assignment 2: Data processing on a big dataset with Databricks Spark

Aim:

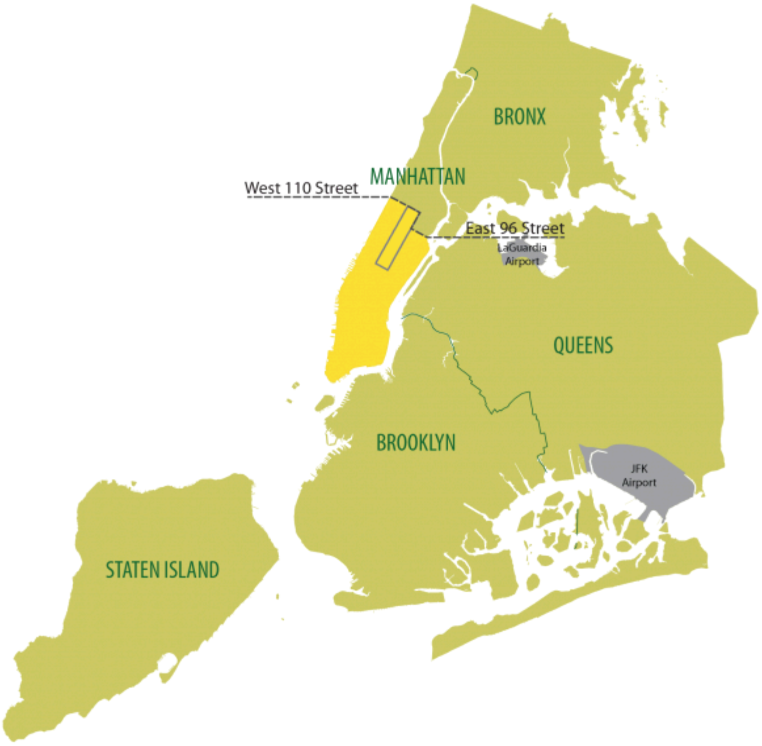
The goal of this assignment is to analyse a large dataset using Spark. You will have to load the available data, perform data transformation and analysis on it and finally train a Machine Learning algorithm for predicting a continuous outcome.

Introduction to the dataset

The New York City Taxi and Limousine Commission (TLC) is the agency responsible for licensing and regulating New York City's taxi cabs since 1971. TLC has publicly published millions of trip records from both yellow and green taxi cabs.
Each record includes fields capturing pick-up and drop-off dates/times, locations, trip distances, itemised fares, rate types, payment types, and driver-reported passenger counts.

Yellow taxi cabs are the iconic taxi vehicles from New York city that have the right to pick up street-hailing passengers anywhere in the city. There are around 13,600 authorised taxis in New York City and each taxi must have a yellow medallion affixed to it.

Green taxis were introduced in August 2013 to improve taxi service and availability in the boroughs. Green taxis may respond to street hails, but only in certain designated areas.



You can find more information about the dataset on this link.

Tasks:

Your tasks will be:

PART 1: Data Ingestion and Preparation

1. Download the dataset for yellow and green taxi cabs from **2015 to 2022** and load it into an Azure Blob Storage:
    a. Yellow:
        i. 2015
        ii. 2016
        iii. 2017
        iv. 2018
        v. 2019
        vi. 2020
        vii. 2021
        viii. 2022
    b. Green:
        i. 2015
        ii. 2016
        iii. 2017
        iv. 2018
        v. 2019
        vi. 2020
        vii. 2021
        viii. 2022

    Note: If you have issue with your wifi speed or Azure (no more credits) you can use this notebook (or the second one if the first one doesn't work due to G Drive blocking too many attempts):
    https://drive.google.com/file/d/17zR27Xwctwi1pCCFDn9gSjbRsTRYWk3_/view?usp=drive_link

    https://drive.google.com/file/d/1qU-bFZRsQwXuHJu5iv3EweZvxGHp0CTv/view?usp=sharing

2. On Databricks, read the files from the Azure storage and make a copy of it into DBFS.

3. Count the total numbers of rows for each taxi colour (yellow and green) by reading the files stored on DBFS:
    a. Green taxi: 66,200,401
    b. Yellow taxi: 663,055,251

4. Download and load the location referential csv: link

5. Convert the **"Green" 2015** parquet into a csv file and send it to your Azure Blob Storage. Compare the size of the parquet file against its csv version then explain why parquet format makes more sense than csv.

   Note: If you have issues with your Azure (no more credits) you can download the csv into your local laptop instead of sending it to Azure Blob Storage.

   Note: From now on, you do not need Azure Blob Storage anymore, read and write data using only Databricks (DBFS, Tables, etc…).

6. Explore the dataset and perform any required data cleaning to remove unrealistic trips (You can use pyspark or sparksql) such as:
   a. Trips finishing before the starting time
   b. Trips where the pickup/dropoff datetime is outside of the range
   c. Trips with negative speed
   d. Trips with very high speed (look for NYC and outside of NYC speed limit)
   e. Trips that are travelling too short or too long (duration wise)
   f. Trips that are travelling too short or too long (distance wise)
   g. Any other logic you think is important

Warning: removing trips only because they are missing data in non used fields will be considered incorrect.

7. Combine the yellow and green taxi dataset together (their schema are not exactly the same).

8. Combine the new dataframe with the location data (there are two locations in each trip, pick up location and drop off location)

9. Export the combined data into a parquet file in DBFS and then load it as a **table or view.**

PART 2: Business Questions (Only use SparkSQL + Take screenshots of results and add in the report)

1. For each year and month (e.g January 2020 => "2020-01-01" or "2020-01" or "Jan 2020":
   a. What was the total number of trips ?
   b. Which day of week (e.g. monday, tuesday, etc..) had the most trips ?
   c. Which hour of the day had the most trips ?
   d. What was the average number of passengers ?
   e. What was the average amount paid per trip (using *total_amount*) ?
   f. What was the average amount paid per passenger (using *total_amount*) ?

=> In a Single table/dataframe/output

2. For each taxi colour (yellow and green):
   a. What was the average, median, minimum and maximum trip duration in minutes (with 2 decimals, eg. 90 seconds = 1.50 min) ?

   b. What was the average, median, minimum and maximum trip distance in **km** ?
   c. What was the average, median, minimum and maximum speed in **km per hour** ?

=> In a Single table/dataframe/output

3. For each taxi colour (yellow and green), each pair of pick up and drop off locations (use boroughs not the id), each month, each day of week and each hours:
   a. What was the total number of trips ?
   b. What was the average distance ?
   c. What was the average amount paid per trip (using *total_amount*) ?
   d. What was the total amount paid (using *total_amount*) ?

=> In a Single table/dataframe/output

4. What was the percentage of trips where drivers received tips?

5. For trips where the **driver received tips**, what was the percentage where the driver received tips of at least $5

6. Classify each trip into bins of durations:
   a. Under 5 Mins
   b. From 5 mins to 10 mins
   c. From 10 mins to 20 mins
   d. From 20 mins to 30 mins
   e. From 30 mins to 60 mins
   f. At least 60 mins

   Then for each bins, calculate:
   a. Average speed (km per hour)
   b. Average distance per dollar (km per $)

=> In a Single table/dataframe/output

7. Which duration bin will you advise a taxi driver to target to **maximise his income**?

PART 3: Machine Learning

1. Build at least two different ML models (two different algorithms) using Spark ML pipelines + a baseline model to predict the **Total amount** of a trip:
   a. Build a baseline model by using the answer of Part 2 Q3c (average paid) and calculate its RMSE.
   b. Use all data except **October/NovemberDecember 2022** to train and validate your models and use the RMSE score to assess your models.
   c. Choose your best model and explain why you choose it (processing time, complexity, accuracy, etc).
   d. Using your best model, predict the **October/NovemberDecember 2022** trips and calculate the RMSE on your predictions. Does your model beat the baseline model

Note: **You are not allowed to use “Fare_amount” and “Tolls_amount” as features for your model.**

Advices:

- Have a read at:
    - Trip Record User Guide
    - Yellow Trips Data Dictionary
    - Green Trips Data Dictionary

- Avoid the data scientist mindset of trying to load all the data and process all data at once. Start small to build and validate your code before running on the entire dataset.

- Think about the order of processing, creating a lot of unused variables in the beginning will increase the processing time. It is better to create them only when you need them.

- Similarly using/creating a lot of variables in the ML part can blow up the processing time. Test with a minimum amount of variables before using more.

- Sometimes restarting the notebook can also help Spark release a lot of memory.

- Databricks will shut down after 60 mins of idling, it is strongly recommended to regularly save as a parquet or table any intermediate steps.

Deliverables:

Each student will have to submit

- At least 1 notebook containing their code in **BOTH ipynb** and **HTML** format
- A “handover” written report (containing the output of every question). (**PDF**)
- Any other relevant documents

The report should not exceed 2500 words (figures and tables are not counted).

Compress all deliverables into a single zip file and use the following file naming format for the submission:
Assignment_2_FirstName_LastName.zip

A good “handover” report should contained:

1. High-level view of your project.
2. Explanation for the different steps of your project.
3. Any issues/bugs you faced and how you solved them.
4. Answers to the business questions.
5. Choices and results of your ML model.
6. Relevant screenshots/images/diagrams/flows if necessary.

You can assume that the reader of your report will have a similar understanding and knowledge of any technical skills (Python/SQL/ML).

A good way to know if you have a good “handover” report is to ask one of your classmates/groupmates to read through it and see if he/she will be confident to “take over” your work.

Assessment Criteria:

- Quality of code .
- Justification of data transformation, data formats, data storage and accuracy of results with evidence supporting claims.
- Quality of findings and recommendations for business questions.
- Quality of ML models
- Clarity and quality of written report.

Criteria Details and weights:

Criteria	Further Details
Quality of code	1. Code can be executed without raising an error. 2. Code is well commented.
Justification of data transformation, data formats, data storage and accuracy of results with evidence supporting claims.	1. High level explanation of each major step and decision. 2. Follows the good “handover” report guidelines
Quality of findings and recommendations for business questions.	1. Correct answers to the business questions. 2. Recommendations to the business are relevant.
Quality of ML models	1. Multiple models created 2. Models are relevant for the problem
Clarity and quality of written report.	1. Complete and professionally formatted report (spelling, grammar, punctuation, layout). 2. Report is not exceeding the maximum length

Due Date:

All assignments need to be submitted before the **due date (7th October)** on Canvas.
Penalties will be applied for late submission

Late submission will be penalised 10 pts per day after the due date.