Milestone 3 - Pre-processing and analysis with PySpark

Deadline - Sunday, 10th of December @11.59 pm

The goal of this milestone is to preprocess the dataset 'New York yellow taxis' by performing basic data preparation and basic analysis to gain a better understanding of the data using PySpark.

Use the same month and year you used for the green taxis in milestone 1. Datasets (download the yellow taxis dataset).

Important Notes:

- You MUST use this notebook template/structure. not doing so will result in marks deduction.
- You MUST have the cells run and output shown similar to milestone 1. I will NOT RUN YOUR NOTEBOOK.

Submission guidelines: same as milestone 1.

Notebook name must be same format as the file you named in miletsone 1. Just M2 instead of M1.

IMPORTANT: You are only allowed to use PySpark unless explicitly told otherwise(i.e last task).

Useful resource/documentation (highly recommended) - PySpark examples

Weight dist.

- Loading the dataset : 5%
- Basic cleaning: 30%
  - column renaming: 10%
  - detect missing: 35%
  - Handle missing: 35%
  - Check missing : 20%
- Analyses: 30%
- Encoding: 20%
- Lookup table: 10%
- Writing the cleaned and lookup table back as parquet and csv files: 5%.

Tasks:

Load the dataset.

Preview first 20 rows.

How many partitions is this dataframe split into?

Basic cleaning

rename all columns (replacing a space with an underscore, and making it lowercase)

Detect and remove duplicates

- Duplicates are trips with same pickup time,pickup location, dropoff time,drop off location and trip distance

check that there is are no duplicates

Detect missing

- Create a function that takes in the df and returns any data structrue of your choice(df/dict,list,tuple,etc) which has the name of the column and percentage of missing entries from the whole dataset.
- Tip : storing the missing info as dict where the key is the column name and value is the percentage would be the easiest.

Prinout the missing info

Handle missing

- For numerical features replace with 0.
- For categorical/strings replace with 'Unknown'

check that there are no missing values

Feature engineering -

Write a function that adds the 3 following features. Use built in fucntions in PySpark (from the functions library) check lab 8, Avoid writing UDFs from scratch.

- trip duration (the format/unit is up to you)
- is_weekend. whether the trip occurred on Saturday or Sunday.
- week number (relevant to the month and not year, i.e 1,2,3,4 nto 31,32,33...)

Preview the first 20 rows (only select the following features: pickup and droptime, and the 3 features you added).

Analyses - Answer the following 5 questions (by showing the output and and a short 1-2 sentences regarding your observation/answer)

MUST Use the PySpark SQL API.

DO NOT explicitly write SQL queries. Doing so will result in 50% deduction (for the question). Check lab 7.

You are free to add columns if it will help in answering a question and add useful info to the dataset.

1- What is the average fare amount per payment type

2- Do people tend to go on a longer trips during the weekend or weekdays?

3 - which day recorded the most trips?

4- What is the average "total amount" of trips with more than 2 passengers?

5- On average, when is it more likely that the tip is higher, when there are multiple passengers or just 1.?

6- What is the most frequent route on the weekend.

Encoding

- Label encode all categorical fetaures.
- Create a lookup table for these label encoded features. You can use the same format/example as the lookup table in Milestone 1 description.

(You are allowed to store and manipulate the lookup table as a pandas dataframe, it does not have to be a PySpark df).

- Remove the original unencoded categorical features from the df after encoding.

Preview first 20 rows of the label encoded features

Preview first 20 rows of your lookup table

Load the cleaned PySpark df to a parquet file and the lookup table to a csv file.

Bonus - Load the cleaned parquet file and lookup table into a Postgres database.

Note that if you decide to do the bonus, you must include not only your notebook but the docker-compose.yaml file aswell.

Screenshot of the table existing in the database and a simple query such as `select count(*) from table_name` or `select * from table_name limit 10`

(You can just copy paste the screenshots in the markdown cells below)