Homework-3: NoSQL Databases

Deadline: September 26th, 11:59PM ET.

In this homework, you will practice the usage of NoSQL Databases.

Your homework submission should be on GitHub. Use the following GitHub classroom to access the assignment: https://classroom.github.com/a/ncYALVvS

You should submit the URL for your GitHub repository on Canvas. Grading penalty will be applied if otherwise.

Cite any external sources you use. External sources shouldn't exceed more than 30% of the final solution.

Submit your answers in one Jupyter notebook. If you need to include any scripts outside of Jupyter, reference it in your notebook – as shown in the below screenshot -. If you need to include output or screenshots, you may include them in your notebook or attach them as separate image files using the naming convention qx.png (where x is the question number). If you have more than one image for a given question, name the files as qx_1.png, qx_2.png, etc.

A sample structure of your notebook is shown below

Q1

Refer to q1.sql and q1.png

Q2

```
from pyspark.sql.functions import monotonically_increasing_id


### Some code here
## For output, refer to q2.png
```

In this homework, our goal is to build the infrastructure of a recommendation engine for various shopping trends.

We will use the Shopping Trends Dataset. You can access it from this URL: https://www.andrew.cmu.edu/user/mfarag/static/shopping_trends.csv

Q1. (10%): Load the dataset into Spark and display the descriptive statistics associated with the dataset (e.g., mean, standard deviation, etc.).

- Submit both code and output screenshot of the description statistics.

Q2. (30%): Graph databases excel in powering recommendation engines. Your task is to design and implement a Python script to store a dataset of user purchases in Neo4j,

creating an intuitive, scalable model for analyzing relationships between users and their purchased items. Consider the following key requirements in your design:

- Capture user attributes like age and location for future analysis.
- Efficiently store items with unique nodes, ensuring each item (e.g., "Blouse" or "Sweater") appears only once in the graph.
- Add purchase-specific details directly to the relationships between users and items.
- Programmatically insert all 3900 user-item relationships without hard-coding them.
- Provide both your Python code and a screenshot of a query that retrieves the stored records.

At this point, assume you no longer have access to the CSV file and use your graph store to answer the questions below.

Q3. (15%): Develop a python function to find the overall percentage of users who are 50+ years old in the dataset.

- Submit both code and output screenshot.

Q4. (15%): Develop a python function to find the most purchased item in Hawaii.

- Submit both code and output screenshot.
- Hint: consider using several functions at Neo4J. You may find the following resources useful:
  - Count: https://neo4j.com/docs/cypher-manual/current/functions/aggregating/
  - Limit: https://neo4j.com/docs/cypher-manual/current/clauses/limit/

Q5. (15%): What is the most popular season to shop in? and what is the most popular shipping method in this season?

- Note: you can ignore the frequency of purchases from your calculations/assumptions.
- Submit both code and output screenshot.

Q6. (15%): Develop a python function to recommend to new shoppers in any US state a total of 3 (or fewer) popular items to view based on what people like to purchase in this state.

- You may ignore gender, age, and other factors for simplicity.
- Submit both code and output screenshot for a new shopper in Kentucky.