Heather Hanwacker

MET CS 777 – Spring O2 2022

March 23, 2022

**Assignment 1 – Read Me**

Spark Data Wrangling (20 points)

Text

Description automatically generatedTable

Description automatically generatedTable

Description automatically generated

Text

Description automatically generated

For both tasks, I took the small data set and implemented it in Google Colab. To confirm that my answers were accurate, and expected I put this data set into Tableau and went through the logic in the tasks. Note: Tableau headers start at 1 and not 0 so medallion is p[0] in Google Colab but F1 in Tableau. The formatting and cleaning of the data is what was presented to use in the starter for assignment 1.

Text

Description automatically generated

For Task 1 the objective is to find the unique number of drivers for each taxi and present the top 10 taxis with the highest number of unique drivers as a set of taxi (medallion), and distinct count of drivers (number of drivers)

**Small Data Set:**

* Tableau – I used index to see where the cut off for top 10 was, 8-11 all have the same number of distinct drivers meaning depending on how you sort the data, depends on which 3 of the 4 taxis would be in the results

Text

Description automatically generated

* Google Colab – the job ran in 1 minute and 42 seconds. The output matches what I have in Tableau, meaning moving to the bigger data set would produce accurate results as well.

Text

Description automatically generated

A screenshot of a computer

Description automatically generated

**Large Data Set:**

In Google Colab the job ran in 1 hour and 33 minutes.

Text

Description automatically generated

Timeline

Description automatically generated with medium confidence

Text

Description automatically generated

This task was a little bit tricker due to the need to combine fields to get a cost per minute to average. This required converting the trip time in seconds to minutes by dividing by 60 (60 seconds in 1 minute) then using the total cost attribute and dividing it by the trip time in minutes. Once that calculation was complete, averaging those values for each unique driver.

**Small Data Set:**

* Tableau – I used index to see where the cut off for top 10 was

Graphical user interface, text

Description automatically generated

* Google Colab – the job ran in 1 minute and 31 seconds. The output matches what I have in Tableau, meaning moving to the bigger data set would produce accurate results as well.

Text

Description automatically generated

A screenshot of a computer

Description automatically generated

**Large Data Set:**

In Google Colab the job ran in 1 hour and 34 minutes.

Text

Description automatically generated

Graphical user interface, application, table

Description automatically generated

**Overall run time for all 4 jobs**

* job-34976a42: task2 on large data
* job-4bfb54b5: task1 on large data
* job-1153eda3: task2 on small data
* job-335608c9: task1 on small data

Table

Description automatically generated

Files Included in submission:

* **readme\_assignment1**– current file that explains assignment and results
* **Screenshot\_main\_\_all\_spark\_job** – screenshot of page with all 4 job runs with start time and elapsed time
* 4.1 Task 1: Top-10 Active Taxis
  + **main\_task1.py** - .py file to execute task 1
  + Input – first argument is data set, second argument is where to store output
  + Output
    - **assignment1\_main\_task1\_output\_lg\_part-00000** – file with top 10 output
  + **Screenshot\_main\_\_task1\_spark\_job** – Screenshot of run time of spark job
* 4.2 Task 2 - Top-10 Best Drivers
  + **main\_task2.py** - .py file to execute task 2
  + Input – first argument is data set, second argument is where to store output
  + Output
    - **assignment1\_main\_task2\_output\_lg\_part-00000** – file with top 10 output
  + **Screenshot\_main\_\_task2\_spark\_job** – Screenshot of run time of spark job