CS777 – Week 1 Homework Submission Template

Task 1 – Top-10 Active Taxis

Many different taxis have had multiple drivers. Write and execute a Spark Python program that computes the top ten taxis that have had the largest number of drivers. Your output should be a set of (medallion, number of drivers) pairs.

***Note**: You should consider that this is a real-world data set that might include wrongly formatted data lines. You should clean up the data before the main processing, a line might not include all of the fields. If a data line is not correctly formatted, you should drop that line and do not consider it.*

- Print a list of top 10 taxis having the largest number of drivers

```
('6FFCF7A4F34BA44239636028E680E438', 576)

('D5C7CD37EA4D372D00F0A681CDC93F11', 557)

('849E486825860106403FB991A763BCC3', 547)

('DA1A4CB0E75444C73D1D1633E701206E', 546)

('A979CDA04CFB8BA3D3ACBA7E8D7F0661', 540)

('A532B1493C4DD88C450F6796369EAA6F', 536)

('818B2426C5493017D5CFE68EFD34617E', 531)

('075E4BFE6607421289B566A32BC135E5', 530)

('FF40FB8123940D9F96D33EDA1D92A83C', 528)

('4DBFC74756F934CC9D4891F308881281', 528)
```

- Include the relevant code excerpt that you used for finding the top 10 taxis

```
task1 = taxiLinesCorrected.map(lambda x: (x[0],x[1]))

  result1=task1.groupByKey().map(lambda x:(x[0],len(x[1]))).top(10, lambda x: x[1])

  answerFor1 = sc.parallelize(result1)

  answerFor1.coalesce(1).saveAsTextFile(sys.argv[2])
```

Task 2 – Top-10 Best Drivers

We would like to figure out who the top 10 best drivers are in terms of their average earned money per minute spent carrying a customer. The total amount field is the total money earned on a trip. In the end, we are interested in computing a set of (driver, money per minute) pairs.

- Print a list of top 10 best drivers based on earned money per minute carrying a customer

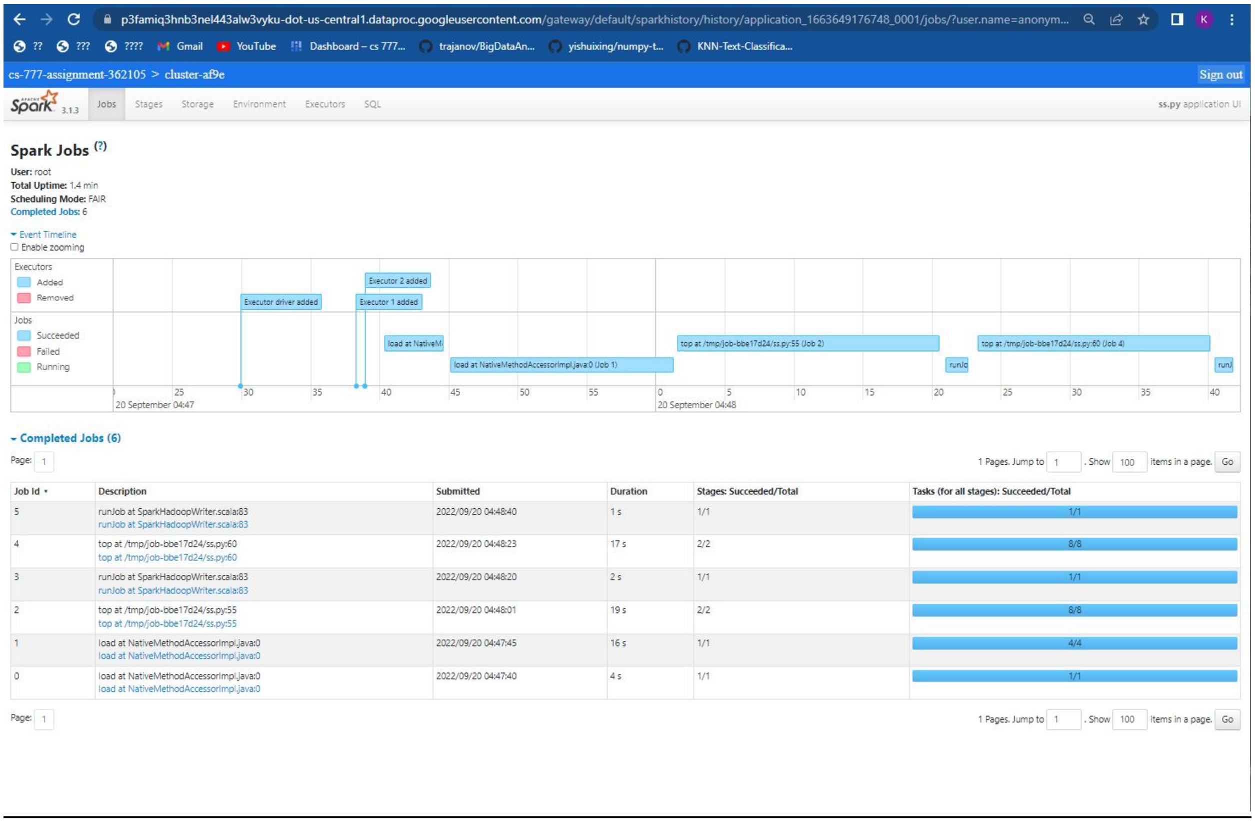
```
('011AE79C7E609378068514E5C992B6D6', 31.47058823529412)

('583D58A6E31DBAF275DDFAD1857448D2', 26.619718309859156)

('C742CFD86A6B2ABFB9CD7228286766CA', 17.313432835820894)

('62A757062319F29FA98D15C8DF8A6BF6', 13.844295302013423)

('7BDFF06419C23F667C5D69EEFBF091BE', 11.313253012048193)

('7F274F176A7BA8D55B5A7D0F2580634C', 7.301946902654868)

('BD50C5800362CB05615C52A0370E1A80', 7.247362012987013)

('8E4805A6BA0B2E51292C14A57D913700', 5.933657351154314)

('49231AC50BA1E05CEF0B59B7B64EA463', 5.909090909090908)

('ED73A273C4149DA01C0B09DFF6B0D6E7', 5.769230769230769)
```

- Include the relevant code excerpt that you used for finding the top 10 best drivers

```
   task2 = taxiLinesCorrected.map(lambda x: (x[1],x[16] / (x[4]/60)))

   result2=task2.mapValues(lambda x:(x,1)).reduceByKey(lambda
x,y:(x[0]+y[0],x[1]+y[1])).map(lambda x:(x[0],x[1][0]/x[1][1])).top(10, lambda x: x[1])

   answerFor2 = sc.parallelize(result2)

   answerFor2.coalesce(1).saveAsTextFile(sys.argv[3])
```

Spark History Output:

To demonstrate that you did execute your code on the cloud it is important to include URLs in the screenshots. Otherwise, there is no way for us to verify if the code was executed in your cloud account.


p3famiq3hnb3nel443alw3vyku-dot-us-central1.dataproc.googleusercontent.com/gateway/default/sparkhistory/history/application_1663649176748_0001/jobs/?user.name=anonym...
K
??
???
????
Gmail
YouTube
Dashboard – cs 777...
trajanov/BigDataAn...
yishuixing/numpy-t...
KNN-Text-Classifica...
cs-777-assignment-362105 > cluster-af9e
Sign out
Spark 3.1.3
Jobs
Stages
Storage
Environment
Executors
SQL
ss.py application UI

Spark Jobs (?)

User: root
Total Uptime: 1.4 min
Scheduling Mode: FAIR
Completed Jobs: 6

▾ Event Timeline
□ Enable zooming

▾ **Completed Jobs (6)**

Page: 1   1 Pages. Jump to 1 . Show 100 items in a page. Go

Job Id ▾	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
5	runJob at SparkHadoopWriter.scala:83 runJob at SparkHadoopWriter.scala:83	2022/09/20 04:48:40	1 s	1/1	1/1
4	top at /tmp/job-bbe17d24/ss.py:60 top at /tmp/job-bbe17d24/ss.py:60	2022/09/20 04:48:23	17 s	2/2	8/8
3	runJob at SparkHadoopWriter.scala:83 runJob at SparkHadoopWriter.scala:83	2022/09/20 04:48:20	2 s	1/1	1/1
2	top at /tmp/job-bbe17d24/ss.py:55 top at /tmp/job-bbe17d24/ss.py:55	2022/09/20 04:48:01	19 s	2/2	8/8
1	load at NativeMethodAccessorImpl.java:0 load at NativeMethodAccessorImpl.java:0	2022/09/20 04:47:45	16 s	1/1	4/4
0	load at NativeMethodAccessorImpl.java:0 load at NativeMethodAccessorImpl.java:0	2022/09/20 04:47:40	4 s	1/1	1/1

Page: 1   1 Pages. Jump to 1 . Show 100 items in a page. Go