Course ID: CSE-557
Total Points 100

Assignment 1

Deadline: March 4, 2022

Instructions

1. The assignment is to be attempted in pairs.
2. Programming Language: Python or in some cases Java.
3. For Plagiarism, institute policy will be followed
4. You need to submit the readme.pdf, Code files and Model files.
5. Report, models and code in .py format should be submitted in the classroom in a zip folder with the name A1_RollNumber1_RollNumber2.zip.
6. Attach a Report.pdf including all your hypothesis, procedure, steps and outcomes in the zip file.
7. You **can use any library** for pre-processing, training, doing experiments and post-processing in all questions.
8. One member should submit on google classroom while other member can mark turn in without the attachment.
9. In case of doubts, please comment on the classroom.

Dataset: Attached in the assignment post.

1 Patents and Publications dataset

Q1: (40 points)

Please download the Patent data and publications data from https://drive.google.com/drive/folders/1H__WxZMqGOMOBEmf-1c5azJP-XOt7R4N?usp=sharing. Convert it to a suitable format to be persisted in Kudu (create Kudu tables and persist the data programatically. You can do it using Impala and pyspark as suggested here https://medium.com/@sciencecommitter/how-to-read-from-and-write\-to-kudu-tables-in-pyspark-via-impala-c4334b98cf05 and https://stackoverflow.com/questions/45361525/load-a-text-file-into-apache-kudu-table. Other approaches are welcome too). Answer the following queries using Spark (you can use spark-sql from python client):

1. Find the affiliation of the author with the maximum number of patents. (**Points:5**)
2. Find the author who has filed the maximum number of patents under the patent code with the prefix 707/E17. (*Hint : join patent_inventors_data and patents_only on author_id*). (**Points:5**)
3. Output the top 10 patents that have been reissued in descending order of the number of times they have been reissued (hint the repetition of patent name with different codes indicates it has been reissued). (**Points: 5**)
4. Find the total number of patents filed by authors at Google and Microsoft. (**Points: 10**)
5. Find the total number of authors who have published a patent and a publication (hint: you have to sue both data sources and necessary csvs from both. (**Points: 5**)
6. Find the patent category name that has the maximum number of patents filed under it. (Hint: Use data in subclass_current.csv in addition to other two data files) (**Points: 5**).
7. Report the number of authors and their names who have data accuracy as one of the research interests. (**Points: 5**).

Q2 Repeat the above queries with psypark over HDFS instead of Kudu **Points: 40**

2 GhTorrent (Github) dataset

Q1: (40 points) Please download the pullrequests, repos and events data from https://drive.google.com/drive/folders/1XZJkuUu2IFluqxeadBc37B_PjNMcfZy0?usp=sharing. Convert it to a suitable format to be persisted in Kudu (create Kudu tables and persist the data programatically. You can do it using Impala and pyspark as suggested here https://medium.com/@sciencecommitter/how-to-read\-from-and-write-to-kudu-tables-in-pyspark-via-impala-c4334b98cf05 and https://stackoverflow.com/questions/45361525/load-a-text-file-into-apache-kudu-table. Other approaches are welcome too). Answer the following queries using Spark (you can use spark-sql from python client):

1. Report the number of pull requests a. opened per day. b. Discussed per day. (Hint: use pull requests csv). (**Points:5**)
2. Find the users common in ghtorrent_logs.csv and pull_requests csv files. (**Points:5**)
3. Find the number of issues closed by "josevalim" (Hint: You have to use both ghtorrent_logs.csv and pull_requests csv). (**Points: 10**)
4. How many repositories were processed in total? Use the api_client. (Focus on WARN messages and use the suffix part of the URL after repos till ?.) (**Points: 10**)
5. Count the number of INFO messages. (**Points: 5**)
6. Which of the repositories has the most failed API calls (**Points: 5**).

Q2 Repeat the above queries with psypark over HDFS instead of Kudu **Points: 40**

3 Delhi Public Transport Dataset

Q1: (40 points) Please download the Delhi Public Transport Dataset from https://drive.google.com/drive/folders/1rle8fEQtb95qrebMOk0kkb2AwZDQwZZa?usp=sharing. You can also explore the data here https://dataspace.mobi/dataset/delhi-gtfs-static. Convert it to a suitable format to be persisted in Kudu (create Kudu tables and persist the data programatically. You can do it using Impala and pyspark as suggested here https://medium.com/@sciencecommitter/how-to-read\-from-and-write-to-kudu-tables-in-pyspark-via-impala-c4334b98cf05 and https://stackoverflow.com/questions/45361525/load-a-text-file-into-apache-kudu-table. Other approaches are welcome too). Answer the following queries using Spark (you can use spark-sql from python client):

1. Report number of trips and number of stops for each route. **(Points:5)**
2. Report the following for each route: **(Points:5)**
   (a) Name of the first stop, and its latitude and longitude
   (b) Name of the last stop, and its latitude and longitude
3. Report timings of arrival for all trips for the shortest route that goes through the stop Govind Puri Metro Station (There are two stops with the name, you can do for either of the stop or for both) **(Points:10)**
4. Print trips in order of descending total time of travel **(Points:5)**
5. Find the name of all stops that are NOT on any of the routes with Govind Puri Metro Station. **(Points:10)**
6. Print name and id of most frequent 3 stops which are starting points for a trip. **(Points:5)**

Q2 Repeat the above queries with psypark over HDFS instead of Kudu **Points: 40**

Assignment 1

Milestones progress

(20 Points) for completing milestones and setup.

Milestones include

- Reading related papers in first week.
- Reporting progress and asking help to the TA every week at a time convenient for both.

PS: If the code doesn't run or gives any error, 0 points will be awarded.

Best of luck for the assignment.