Scalable Data Mining (Autumn 2018)

Assignment 2 (Full Marks: 150)

Steps for Spark installation:

1. Follow the guidelines given in this link to install Spark in your system:

https://medium.com/@josemarcialportilla/installing-scala-and-spark-on-ubuntu-5665ee4b62b1

Instructions: Please submit your answers to the following questions as a write-up in a PDF file and your codes via Moodle.

Question 1 (Marks = 25+25)

Download the file from this link on google drive: data2_1 . Write a function to load this data in an RDD and name it as 'assignment2_1'. Make sure you use a case class to map the file fields.

Each line in this file contains the following fields: debug_level**:** String, timestamp**:** Date, download_id**:** Integer, retrieval_stage**:** String, rest**:** String

Example: **DEBUG, 2017-03-24T12:06:23+00:00, ghtorrent-49 -- ghtorrent.rb: Repo Shikanime/print exists**

Here, debug_level = DEBUG ; timestamp = 2017-03-24T12:06:23+00:00 ; download_id = ghtorrent-49 ; retrieval_stage = ghtorrent.rb ; rest = Repo Shikanime / print exists

Process this data to answer the following questions:

a. How many lines does the RDD contain?
b. Count the number of "WARN" messages.
c. How many repositories were processed in total when the retrieval_stage is "api_client" ?

   [Take the contents of the field 'rest' and search for 'Repo' or 'repos'.
   For example: **DEBUG, 2017-03-24T12:06:23+00:00, ghtorrent-49 -- ghtorrent.rb: Repo Shikanime/print exists -->** the name of the repository for this entry is '**Shikanime/print**'.

INFO, 2017-03-23T13:00:55+00:00, ghtorrent-42 -- api_client.rb: Successful request URL:https://api.github.com/repos/CanonicalLtd/maas-docs/issues/365/events?per_page=100, Remaining: 4943, Total: 88 ms --> the name of the repository for this entry is '**CanonicalLtd/maas-docs' .]**

d. Using retrieval_stage as "api_client", find which clients did the most HTTP requests and FAILED HTTP requests from the download_id field.
e. Find the most active hour of the day and most active repository.
f. Which access keys are failing most often?

Question 2 (Marks = 12.5+12.5)

Using the same data file from Question 1, perform the following operations:

a. Create a function that given an RDD and a field (e.g. download_id), it computes an inverted index on the RDD for efficiently searching the records of the RDD using values of the field as keys.
b. Compute the number of different repositories accessed by the client 'ghtorrent-22' (without using the inverted index).
c. Compute the number of different repositories accessed by the client 'ghtorrent-22' using the inverted index calculated above.

Question 3 (Marks = 12.5+12.5)

Download the file from this link on google drive: data2_2 **.** The format of the file is in CSV, and the meaning of the fields are self-explanatory as given in the file. Process this file to answer the following questions:

a. Read in the file to an RDD and name it as 'assignment2_2' and count the number of records.
b. How many records in the log file (used in the last 2 questions) refer to entries in the 'assignment2_2' file ?
[Hint: *You need to key both the RDDs ('assignment2_1' and 'assignment2_2') by the substring for repository name in 'assignment2_1' matching with 'name' field in 'assignment2_2' and perform a JOIN operation*.
For example: If the 'name' field in 'assignment2_2' is '**print**' and the repository name in 'assignment2_1' is '**Shikanime/print**', the corresponding records will be joined.]

c. Which of the 'assignment2_2' repositories has the most failed API calls?

Question 4 (Marks = 50)

Implement the K-means clustering algorithm on the data given here: data2_3 to find the clusters for K = 5. This data consists of ‘n’ features for each instance. Run the developed tensorflow program using :

1. Using CPU only
2. Using multiple devices (both CPU and GPU).

Report the running time of algorithm with both the options. Show the visualization of the clusters using PCA.

Submission Instructions:

In your write-up, please provide a description of how you are going to use Spark to solve each problem using Scala. Don’t write more than 3 to 4 sentences for this; we only want a very high-level description of your strategy to tackle the problems.

You will submit 2 files for each question in the following format:

1. Submit your code using the filename *RollNo_AssignmentNo_QuesNo.scala*
2. Submit the output file using the filename *RollNo_AssignmentNo_QuesNo.txt*
3. Submit the write-up using the filename *RollNo_AssignmentNo_QuesNo.pdf*
4. For Question 4, submit the visualization in .png file format using the filename *RollNo_AssignmentNo_QuesNo.png*