CS 6350- Big Data Analytics and Management

Summer 2015 Homework # 3

Topic: Spark & Recommendation System

Due July,29

In this homework you will learn how to solve problems using Apache Spark. Please apply ***Apache Spark*** interactive shell (for **scala** or **python**) / run from command line (**scala/java/python**) to derive some statistics from **Yelp dataset**. You can find the dataset in ***elearning***. Copy the data into your hadoop cluster and use it as input data. You can use the put or ***copyFromLocal*** HDFS shell command to copy those files into your HDFS directory.

All dataset files are (::) separated.

The files are business3.csv, user3.csv, review3.csv.

The csv files has 24 columns, namely

Column id : Name of Column

Column 0 :review_id

Column 1: text

Column 2: business_id

Column 3: full_address

Column 4: schools

Column 5: longitude

Column 6: average_stars//this is for the business entity type only

Column 7: date

Column 8: user_id

Column 9: open

Column10: categories

Column11: photo_url

Column12: city

Column13: review_count

Column14: name

Column15: neighborhoods

Column 16: url

Column 17: votes.cool

Column 18: votes.funny

Column 19: state

Column 20: stars:: //this is for review entity type only

Column 21: latitude

Column 22: type

Column 23: votes.useful

Q1. Given input ***address(any part of the address e.g., city or state)***, find all the ***business ids located at the address***. You must take the input ***address*** in the command line. [For example, if the input ***address*** is ***Stanford*** then you need to find all businesses with stanford ***in the address column***] [You only need ***business3.csv*** file to get the answer.]

Q2.

a. Start spark-shell in local mode using all the processor cores on your system or the cluster. (Very important)

List the business_id of the Top 10 businesses using the average ratings. This will require you to use review3.csv.

Please answer the question by calculating the average ratings given to each business using the review3.csv file. Do not use the already calculated ratings (average_stars) contained in the business entity rows.

b. Rerun Q2a using Yarn mode. Please solve using cs6360 cluster. This questions shows how spark can be used on multiple systems in a cluster.

Load all the dataset to hadoop cluster as you did in homework1.

Use the address of the file on the cluster as input to your scala script.

Start spark-shell in **YARN mode** using Cs6360 spark cluster.

This spark cluster consist **6 hadoop machine nodes**.

Using the following parameters Rerun your scala script from question 2a.

Set executor memory =2G

executor cores = 6.

num-executors = 6

example

```
spark-shell --master yarn-client --executor-memory 4G --executor-cores 7 --num-executors 6
```

How does it affect the execution time and efficiency of your program?

Note: Spark supports only scala or java in YARN mode.

Submission:

You have to upload your submission via e-learning before due date. Please upload the following to eLearning:

1. Two scripting file like, Q1.txt, and Q2.txt separately if you use scala / python/java interactive shell. Each file contains the scala /python/java code. If you use java, then submit all the java files.
2. Also, submit the commands to start the spark shell for Question 2a. and 2b.

Part II: Mahout & Recommendation

Q3. Read the following link related to co-occurrence based recommendation system implemented in mahout.

https://mahout.apache.org/users/algorithms/intro-cooccurrence-spark.html

Currently Mahout switched from MapReduce to Apache Spark. It has an interactive shell (showed in the class, lecture contains how to install it). Using that, apply item-based collaborative filtering using mahout's ***spark-itemsimilarity***.

The review3.csv contains ratings for each business. Using businesses rated as 4 by the users, recommend other businesses, a user may be interested in. For each business, show maximum of 5 recommended businesses"

Steps to follow:

1. Read the above link carefully and construct the item-similarity matrix of each business having rating 4 (use review3.csv). The output should be like this:

```
liWxota5DH7Roo-iv0pTmw  LhmoZDDMBYRgHdWUw0L3tw:15.809918182669207
Mg1CS5aRT_eV4qaGpEvnIQ:13.957741537247784
e_8TvfKT6QT81snfrqYYTw:12.667998329270631

CSn2-XpArLLYeZ_k2BKzrg  etLsW18rOOhxQc0sQ-BGFQ:18.692827314604074
rRc6aK4n2oSXLQniRCT4uw:18.692827314604074
hDhC_DgEIKK5D961doKOXQ:15.920379192451946 JZXLgxJLKd9T-
```

R6hrJhRZg:15.920379192451946 ViJjcvNH_Up1mXZ5NieApg:14.87402350991033 GSe0S6LTB01O5z-6V6T87g:14.19456797267776 wsuzuLD3KV7L8oMinLCI0A:14.19456797267776 gnqHyA6gY2-m9S-FKt6Xqw:14.19456797267776 epvLkQNL6MOvk3s6JlTntA:13.286795857711695 C31ExBTn_6UxbTVkWPtNkg:12.95204253285192 RWd83o4drIOE60W1meaESA:12.95204253285192 miMx3VNOW8qPJC5wEGaIOw:11.992239395505749 e0prCZXtHGQIKeQ_wTW3uw:11.810310593340546 kNtToQSP_Y5U8tznLXuCaw:11.088662921916693 0-WocGTpO3Zm4q1Zzz49Rw:11.088662921916693

2. Please save the above file to HDFS. Now Run Apache spark interactive shell. From the shell, take the user id as input (you can fix the id, e.g., val userID = "Wi-IiJA36kpnk_Vphuq0zQ"). Now finds all the business that he rates as 4.
3. Please Load/read the above file (item-similarity file) and find the businesses that match with the user’s rated business with the key of the item-similarity file.

   For example, suppose a user has id 7 and he rates 955 and 123 as 4. After executing this, you will get a matrix like following(a sample):

   ```
   955     898,951,910,905,1269
   123     3265,1218,1089,3224,247
   ```

   Hint: This reformats the output from the item-similarity matrix, listing the top five recommended businesses.

Submission:

Please upload your submission via e-learning before due date.

1. A scripting file like, Q3_1.txt that shows the building of spark-itemsimilarity and another scripting file Q3_2.txt shows the scala/java/python program (contains codes for step 2 - 3).

   If you use java, then submit all the java files.