CS6350
Big data Management Analytics and Management
Fall 2017
Homework 2
Submission Deadline: 11:59 pm, 29th October, 2017

In this homework, you will use spark (**spark, spark dataframe, spark sql**) to solve the following problems. **In particular, for each question you will implement it first using spark and then using spark sql along with data frame. In other words, for each question, you will end up 2 separate implementations/answers**.

Q1

Write a spark script to find total number of common friends for any possible friend pairs. The key idea is that if two people are friend then they have a lot of mutual/common friends.

For example,
Alice's friends are Bob, Sam, Sara, Nancy
Bob's friends are Alice, Sam, Clara, Nancy
Sara's friends are Alice, Sam, Clara, Nancy

As Alice and Bob are friend and so, their mutual friend list is [Sam, Nancy]
As Sara and Bob are not friend and so, their mutual friend list is empty. (**In this case you may exclude them from your output**).

Input:
Input files
1. soc-LiveJournal1Adj.txt located in /socNetData/networkdata in hdfs on cs6360 cluster (I will also provide this file so that you can test it locally)

The input contains the adjacency list and has multiple lines in the following format:
<User><TAB><Friends>

Here, <User> is a unique integer ID corresponding to a unique user and <Friends> is a comma-separated list of unique IDs (<User> ID) corresponding to the friends of the user. Note that the friendships are mutual (i.e., edges are undirected): if A is friend with B then B is also friend with A. The data provided is consistent with that rule as there is an explicit entry for each side of each edge. So when you make the pair, always consider (A, B) or (B, A) for user A and B but not both.

Output: The output should contain one line per user in the following format:
<User_A>, <User_B><TAB><Mutual/Common Friend Number>

where <User_A> & <User_B> are unique IDs corresponding to a user A and B (A and B are friend). < Mutual/Common Friend Number > is total number of common friends between user A and user B.

Q2.

Please answer this question by using dataset from Q1 and 'userdata' data set below.

1. soc-LiveJournal1Adj.txt

2. userdata.txt

The userdata.txt consists of
column1 : userid
column2 : firstname
column3 : lastname
column4 : address
column5: city
column6 :state
column7 : zipcode
column8 :country
column9 :username
column10 : date of birth.

Find top-10 friend pairs by their total number of common friends. For each top-10 friend pair print detail information in decreasing order of total number of common friends. More specifically the output format can be:

<Total number of Common Friends><TAB><First Name of User A><TAB><Last Name of User A> <TAB><address of User A><TAB><First Name of User B><TAB><Last Name of User B><TAB>
<address of User B>
…

Q3.

In this question, you will apply Spark to derive some statistics from **Yelp Dataset.**

---------- **Data set Info** ----------
The dataset files are as follows and columns are separated using '**::**'
business.csv.
review.csv.
user.csv.

Dataset Description.
The dataset comprises of **three** csv files, namely user.csv, business.csv and review.csv.

Business.csv file contain basic information about local businesses.
Business.csv file contains the following columns
"business_id"::"full_address"::"categories"

'business_id': (a unique identifier for the business)
'full_address': (localized address),
'categories': [(localized category names)]

review.csv file contains the star rating given by a user to a business. Use user_id to associate this review with others by the same user. Use business_id to associate this review with others of the same business.

review.csv file contains the following columns
"review_id"::"user_id"::"business_id"::"stars"
'review_id': (a unique identifier for the review)
'user_id': (the identifier of the reviewed business),
'business_id': (the identifier of the authoring user),
'stars': (star rating, integer 1-5), the rating given by the user to a business

user.csv file contains aggregate information about a single user across all of Yelp
user.csv file contains the following columns "user_id"::"name"::"url"
user_id': (unique user identifier),
'name': (first name, last initial, like 'Matt J.'), this column has been made anonymous to preserve privacy
'url': url of the user on yelp

NB: :: is Column separator in the files.

List the business_id, full address and categories of the Top 10 mostly-reviewed businesses based on number of users that have rated these businesses.
This will require you to use **review.csv** and **business.csv files.**

Sample output:

business id	**full address**	**categories**	**number rated**
xdf12344444444,	CA 91711	List['Local Services', 'Carpet Cleaning']	100

Q4.
Use **Yelp Dataset**

List the 'user id' and 'rating' of users that reviewed businesses located in “Palo

Alto”
Required files are 'business' and 'review'.

Sample output

User id	**Rating**
0WaCdhr3aXb0G0niwTMGTg	4.0

Submission Instructions:

You have to upload your submission via e-learning before due date.
Please upload the following to eLearning:
The zip files, one for each problem.
- source code.
- An output of your program
- ***A Readme text file about how to run your script. Give the command to run your script.