COMP9313 2017s1 Project 3

Problem 1 (15 pts): Data Analytics Using Spark

Download the sample input file "Votes.csv" from: https://webcms3.cse.unsw.edu.au/COMP9313/16s2/resources/5239, and put it in HDFS folder "/user/comp9313/input". In this file, the fields are separated by ',' and the lines are separated by '\n'. The data format of "Votes.csv" is as below:

```
- Id
- PostId
- VoteTypeId
   - ` 1`: AcceptedByOriginator
   - ` 2`: UpMod
   - ` 3`: DownMod
   - ` 4`: Offensive
   - ` 5`: Favorite - if VoteTypeId = 5 UserId will be populated
   - ` 6`: Close
   - ` 7`: Reopen
   - ` 8`: BountyStart
   - ` 9`: BountyClose
   - `10`: Deletion
   - `11`: Undeletion
   - `12`: Spam
   - `13`: InformModerator
- UserId (only for VoteTypeId 5)
- CreationDate
```

Question 1 (5 pts). Find the top-5 VoteTypeIds that have the most distinct posts. You need to output the VoteTypeId and the number of posts. The results are ranked in descending order according to the number of posts, and each line is in format of: VoteTypeId\tNumber of posts.

Question 2 (10 pts). Find all posts that are favoured by more than 10 users. You need to output both PostId and the list of UserIds, and each line is in format of:

$$\text{PostId\#UserId}_1\text{,UserId}_2\text{,UserId}_3\text{,}\ldots\text{,UserId}_n$$

The lines are sorted according to the NUMERIC values of the PostIds in ascending order. Within each line, the UserIds are sorted according to their NUMERIC values in ascending order.

(Hint: In both questions, you need to format your output as specified, and the mkString function is useful.)

Code Template

The code template is provided, and you can download it at: https://webcms3.cse.unsw.edu.au/COMP9313/17s1/resources/7440. You only need to write your code in the two functions.

In order to run the code, you need to create a Scala project in Eclipse, and create a package "comp9313.ass3" in the project, and put the code template "Problem1.scala" in this package.

Problem 2 (10 pts): Top-*k* Most Frequent Co-occurring Term Pairs

Given a large text file, your task is to find out the top-*k* most frequent co-occurring term pairs. The co-occurrence of (w, u) is defined as: u and w appear in the same line (i.e., (w, u) and (u, w) are treated equally).

- Ignore the letter case, i.e., consider all words as lower case.
- Ignore terms starting with non-alphabetical characters, i.e., only consider terms starting with "a" to "z".
- The length of the term is obtained by the length() function of String. E.g., the length of "text234sdf" is 10.
- Use the following split function to split the documents into terms:

```
split("[\\s*$&#/\"'\\,.:;?!\\[\\](){}<>~\\-_]+")
```

You can use the text file pg100.txt (available at: http://www.gutenberg.org/cache/epub/100/pg100.txt) as the sample input.

Output format:

Your Spark program should generate a list of ***k*** key-value pairs ranked in descending order according to the frequencies, where the keys are the pair of terms (the two terms are sorted in alphabetical order and separated by "**,**"), and the values are the co-occurring frequencies, and keys and values are separated by "**\t**", like:

a, one**\t**4452

one, some**\t** 3534

… …

mine,yours**\t** 2545

Name your scala file as “Problem2.scala”, the object as “Problem2”, and put it in a package “comp9313.ass3”. Your program should take three parameters: the input text file, the output folder, and the value of ***k***.

Documentation and code readability

Your source code will be inspected and marked based on readability and ease of understanding. The documentation (comments of the codes) in your source code is also important. Below is an indicative marking scheme:

Result correctness: 90%
Code structure, Readability, and Documentation: 10%

Submission:

Deadline: Sun 14th May 21:59:59

Log in any CSE server (williams or wagner), and use the give command below to submit your solutions:

$ give cs9313 assignment3 Problem1.scala Problem2.scala

Or you can submit through:
https://cgi.cse.unsw.edu.au/~give/Student/give.php

If you submit your assignment more than once, the last submission will replace the previous one. To prove successful submission, please take a screenshot as assignment submission instructions show and keep it by yourself.

Late submission penalty

10% reduction of your marks for the 1st day, 30% reduction/day for the following days.

Plagiarism:

The work you submit must be your own work. Submission of work partially or completely derived from any other person or jointly written with any other person is not permitted. The penalties for such an offence may include negative marks, automatic failure of the course and possibly other academic

discipline. Assignment submissions will be examined manually.

Relevant scholarship authorities will be informed if students holding scholarships are involved in an incident of plagiarism or other misconduct.

Do not provide or show your assignment work to any other person - apart from the teaching staff of this subject. If you knowingly provide or show your assignment work to another person for any reason, and work derived from it is submitted you may be penalized, even if the work was submitted without your knowledge or consent.