**Individual, Assignment 3, Spring 2019, Due: March 15 by 11:59PM to Blackboard**

***Individual Category A – Questions 1 – 3***

**Submission:**

Please put all of your files (code and output) into a single folder, ZIP it, and then submit via Blackboard. **Save your ZIP file as LastName-FirstName.ZIP.** (If you have issues uploading to Blackboard, then email it to me directly. Email **both** my Mason and GMAIL.)

**Emails:** Joe.Wilck@mason.wm.edu Joe.Wilck@gmail.com

**Submission Files:**

**Question 1: code (Scala) and output (Text)**

**Question 2: Word or PowerPoint (or both?)**

**Question 3: Word**

**Grade:**

This assignment will count as 20% of your course grade.

**Question 1:**

**Data:** You will also need the ml-100k data (Lesson 2)

Create a Scala Spark program that calculates the average rating for every movie with at least 100 ratings. Sort the output so that it includes average ratings from smallest to largest. The output should be formatted <Movie Name>”,”<Average>”,”<Number of Ratings> (or something similar). [Hint: You may need to start this one from scratch, since it combines a number of concepts from the examples we went over in prior lessons.]

*Note, this is the same question that was asked on the Lesson 7 (Group Assignment 1). Note, some of the groups did not have the correct number of ratings in their final output, but did have the correct average (but somehow overwrote the number of ratings). So be careful! The solution is posted in the Assignments folder.*

Include your scala code and your output.

**Question 2:**

*(Solution to this example in Python and MapReduce was provided in Lesson 6)*

Create a plan (either a flowchart or detailed explanation, or both) for how to accomplish this in Spark and Scala. At minimum your flowchart and/or explanation should be at the same detail as Lesson 12 Slides (Slide #16).

**Note, I am not asking you to code this in Spark – Scala. I’m asking for you to create a plan for how you would code it.**

**Input File:** input-f.txt (Lesson 6)

The input ﬁle contains the adjacency list and has multiple lines in the following format:

<User><TAB><Connections>

Here, <User> is a unique integer ID corresponding to a unique user and **<Connections>** is a comma-separated list of unique IDs **corresponding to the connections of the user** with the unique ID <User>. Note that the connections are mutual (e.g., if A is connected with B then B is also connected with A). The data provided is consistent with that rule as there is an entry for each side of each connection.

**Approach:**

Use a simple approach such that, for each user X, the program recommends **ten users who are not already connections with X,** but **have the largest number of mutual connections** in common with X. Note, it is possible that X may get less than ten recommendations (even zero); since you only need to look at mutual connections. (Solution in Python and MapReduce is the code file: rec-f.py)

**Output:**

The output should contain one line per user in the following format:

<User><TAB><Recommendations>

where <User> is a unique ID corresponding to a user and <Recommendations> is a comma-separated list of unique IDs corresponding to the program’s recommendation of people that <User> might know, ordered by decreasing number of mutual connections. Even if a user has fewer than 10 second-degree connections, output all of them in decreasing order of the number of mutual connections. If a user has no connections, you can provide an empty list of recommendations. If there are multiple users with the same number of mutual connections, ties are broken by ordering them in a numerically ascending order of their user IDs.

**Question 3:**

Using a Spark – Scala script from one of the Lessons as inspiration or one of the questions above, write an email to your “boss” that is 100 words or less (based on Microsoft Word) that explains what we did with the Spark program, the results, and the “big picture” (i.e., business concept) realization. This is not an Executive Summary, but should be written in a professional manner. Assume your boss understands the technical aspects of Spark and Scala well, but needs the “highlights” in a quick and succinct manner. (Note, this paragraph is 100 words long, according to Microsoft Word.)