Technische Universität Berlin
Database Systems and Information Management Group
C. Boden, Prof. V. Markl, Dr. S. Schelter, J. Soto

AIM-3 Scalable Data Science
Summer Semester 2016

Assignment 3

Recommender Systems, Stream Processing, Network Analysis, and Log Analysis

Due on Thursday, July 21st, 2016 at 18:00

1. Recommender Systems: Item-based Collaborative Filtering (Total 5 pts.)

Suppose you are given a dataset of ratings as depicted in the table below:

	Movie A	Movie B	Movie C	Movie D
Ryan	1	5	3	5
Stavros	-	-	2	-
Brahma	-	4	1	1
Brodie	4	3	-	2
Zosimus	-	5	1	-

1. Compute the **item similarity matrix** (denoted by R) using **Pearson Correlation** as a measure.
2. Compute the **item similarity matrix** (denoted by S) using **Jaccard Coefficient** as a measure.
3. Use matrices R and S to compute the corresponding ratings that *Zosimus* would give to *Movies A & D.*
4. Use matrices R and S to compute the corresponding ratings that *Stavros* would give to *Movies A & B.*

For each sub-problem 1-4, show how you arrived at your solution.

2. Collaborative Filtering (Total 5 pts.)

Solve exercise 9.3.4 on page 327 (MMDS book). Show all work.

3. Stream Processing (Total 10 pts.)

In this exercise, you will familiarise yourself with streaming in Flink.

A. Preparation

As a starting point, read both the Flink Basic API Concepts[1] and the Flink Streaming Guide[2]. In addition, for this specific exercise, you should use Apache Flink version 1.0.3.

[1] Basic API Concepts - `https://ci.apache.org/projects/flink/flink-docs-release-1.0/apis/common/index.html`

[2] Flink Streaming Guide - `https://ci.apache.org/projects/flink/flink-docs-release-1.0/apis/streaming/index.html`

B. Foundations

Prior to answering the following questions, compare the Word Count example referenced in the Batch API[3] with the corresponding one referenced in the Streaming API[4].

1. What are the key differences between batch and stream data sources?

2. What problem will we encounter, if blocking operators, such as aggregations are used in streaming queries?

3. How do streaming engines address the problem referenced just above? Hint: Examine the aforementioned Word Count examples.

4. Batch processing engines like Apache Spark simulate streaming behavior using discretized streams (D-Streams). What is the main idea behind D-Streams?

5. What are the disadvantages attributed to using D-Streams on a batch processing engine, as opposed to using a runtime engine that offers native streaming support?

C. Monitoring a Fleet of Taxis

Suppose an Administrator (named Susan) at a Taxi Corporate Office monitors her companies fleet using a software system that tracks GPS signals. She has contacted you and requests an upgrade to the software system, i.e., extend its capabilities to monitor additional events described further below).

1. Examine the TopSpeedWindowing example and illustrate the query in a flow graph. Describe which data are transferred along the different edges.

2. Adjust the TaxiMonitorStub (which is based on the TopSpeedWindowing example) stream data generator as follows:
   (a) Urban taxi drivers seldom drive faster than 50 km/h, but do so on occasion.
   (b) Urban taxi drivers experience long wait times between customers periodically.

3. Using your adapted stream data generator, display an alert, whenever a taxi drives faster than 50 km/h (higherThan50 method).

4. Using your adapted stream data generator, display an alert that continuously notifies (each second) about all taxis that are idle for over 5 seconds (parkingSince5Seconds method).

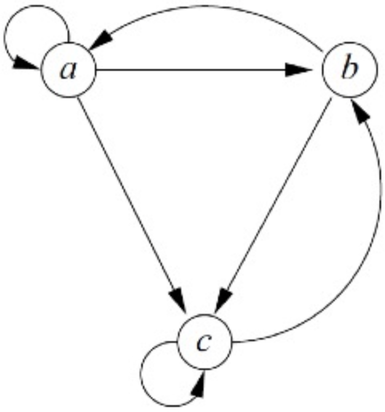
5. Update your illustration to reflect all of the changes you have made.

[3] Word Count Batch - `https://github.com/apache/flink/blob/release-1.0.3-rc3/flink-examples/flink-examples-batch/src/main/java/org/apache/flink/examples/java/wordcount/WordCount.java`

[4] Word Count Stream - `https://github.com/apache/flink/blob/release-1.0.3-rc3/flink-examples/flink-examples-streaming/src/main/java/org/apache/flink/streaming/examples/windowing/WindowWordCount.java`

4. The PageRank Algorithm (Total 5 pts.)

This exercise is based on exercise 5.1.1 in the MMDS book on page 175. Compute the PageRank vector for the following graph assuming no taxation. Show your work including the adjacency matrix, the initial PageRank vector, and the values of the PageRank vector for the first three iterations, as well as an approximation of the converged result rounded to two significant figures.



5. The PageRank Algorithm (Total 5 pts.)

Solve exercise 5.1.2 on page 176 (MMDS book). Show all work.

6. Weblog Analysis (Total 10 pts.)

In this exercise, you will create a SQL query and build a Flink Job based on it. To get started, have a look at the WebLogAnalysis.java and the WebLogData.java class inside the repository. The WebLogData.java class contains the data set that you should use, the WebLogAnalysis.java class needs to be adjusted so that it answers the following questions:

Question 1: "Which IPs visit what webpages with a rank bigger than 90 that utilize the term 'Lorem'?"
Question 2: "What is the rank for the most visited website based on IPs that do not originate in Germany?"

Before you start working on the Flink Job create two copies of the WebLogAnalysis.java class and adjust the SQL query that are also provided in the comments so that they answer the above described questions. Your solution should also work with different data sets that have the same format as the provided one. The following needs to be submitted:

1. A description of what the original WebLogAnalysis.java class does (i.e. asking the question that is answered by it).
2. The adjusted WebLogAnalysis.java files that contain the SQL queries and the Flink Jobs.
3. The answers to the above asked questions, including the output of your jobs.
4. Two ideas for other questions that can be answered using the provided data set.

Deadline and General Instructions

The source code stubs for exercises 3 and 6 are available in our GitLab repository. You will need to upload your results in ISIS as a zip archive (adhering to the structure and naming conventions listed below) by no later than July 21st, 2016 at 18:00. On Friday, July 22nd, 2016 you will need to bring a stapled, printed copy of your homework assignment, including the key source code file(s), runs (i.e., the output), and hand-written solutions. Also, be aware that it is plausible you may be asked to meet with one of the instructors to run your codes. Lastly, be certain to work individually, you are free to drop some hints, but try your best to solve these problems on your own.

```
aim3-SS16-<name>.zip
├── author.txt (contains your name and matriculation number)
├── task 3
│   └── patch file
├── task 6
│   └── adjusted WebLogAnalysis.java files
└── documentation.pdf (your answers to every question and documentation)
```