Assignment #4

CSCI 5408 (Data Management, Warehousing, Analytics)
Faculty of Computer Science, Dalhousie University

Date Given: Feb 28, 2022
Due Date: Mar 13, 2022 at 11:59 pm

Late Submissions are not accepted and will result in a late penalty of 10% deductions / day in the assignment.

Disclaimer: This assignment requires students to work on Spark framework for unstructured data processing, MongoDb for data storing, and Neo4j graph database for visualization. Submissions related to this assignment will not be used for commercial purposes.

Objective:

- The objective of this assignment is to understand Big Data processing problems, and NoSQL database (document, and graph).

Plagiarism Policy:

- This assignment is an individual task. Collaboration of any type amounts to a violation of the academic integrity policy and will be reported to the AIO.
- Content cannot be copied verbatim from any source(s). Please understand the concept and write in your own words. In addition, cite the actual source. Failing to do so will be considered as plagiarism and/or cheating.
- The Dalhousie Academic Integrity policy applies to all material submitted as part of this course. Please understand the policy, which is available at: https://www.dal.ca/dept/university_secretariat/academic-integrity.html

Assignment Rubric

	Excellent (25%)	Proficient (15%)	Marginal (5%)	Unacceptable (0%)	This Rubric Applied to
Completeness including Citation	All required tasks are completed	Submission highlights tasks completion. However, missed some tasks in between, which created a disconnection	Some tasks are completed, which are disjoint in nature.	Incorrect and irrelevant	Problem #1 Problem #2
Correctness	All parts of the given tasks are correct	Most of the given tasks are correct However, some portions need	Most of the given tasks are incorrect. The submission	Incorrect and unacceptable	Problem #1 Problem #2

		minor modifications	requires major modifications.		
Novelty	The submission contains novel contribution in key segments, which is a clear indication of application knowledge	The submission lacks novel contributions. There are some evidences of novelty, however, it is not significant	The submission does not contain novel contributions. However, there is an evidence of some effort	There is no novelty	Problem #1 Problem #2
Clarity	The written or graphical materials, and developed applications provide a clear picture of the concept, and highlights the clarity	The written or graphical materials and developed applications do not show clear picture of the concept. There is room for improvement	The written or graphical materials, and developed applications fail to prove the clarity. Background knowledge is needed	Failed to prove the clarity. Need proper background knowledge to perform the tasks	Problem #1 Problem #2

Citation:
McKinney, B. (2018). The impact of program-wide discussion board grading rubrics on students' and faculty satisfaction. Online Learning, 22(2), 289-299.

This assignment requires you to submit programming codes on gitlab, and a PDF file(s) on Brightspace.

Problem #1: This problem contains two tasks.

Task 1: Cluster Setup - Apache Spark Framework on GCP

Using your GCP cloud account, configure and initialize Apache Spark cluster. This cluster will be used for Problem #2.

(Follow the tutorials provided in Lab session).

Create a flowchart or write **½ page explanation on how you completed the task**, include this part in your PDF file.

Task 2: Data Extraction and Preprocessing Engine: Sources – Twitter messages

Steps for Twitter Operation

Step 1: Create a Twitter developer account
Step 2: Explore documentation of the Twitter **search** and **streaming** APIs and required data format. In your own words, write ½ page summary about your findings.

Step 3: Write a well-formed program using Java to extract data (**Extraction Engine**) from Twitter. Execute/Run the program on local machine. You can use Search API or Streaming API or both.

(Do not use any online program codes. You can only use API specification codes given within official Twitter documentation)

- The search keywords are "**mask**", "**cold**", "**immune**", "**vaccine**", "**flu**", "**snow**".

Step 4: You need to **include a flowchart/algorithm of your tweet extraction program in your problem#1 PDF file**.

Step 5: You need to extract the tweets and metadata related to the given keywords.

- For some keywords, you may get less number of tweets, which is not a problem. Collectively, you should get approximately 3000 to 5000 tweets.

Step 6: If you get less data, run your method/program using a scheduler module to extract more data points from Twitter at different time intervals. **Note**: Working on small datasets will not use huge cloud resource or your local cluster memory.

Step 7: You should extract tweets, and retweets along with provided meta data, such as location, time etc.

Step 8: **The captured raw data should be kept (programmatically) in files. Each file should not contain more than 100 tweets. These files will be needed for Problem #2**

Step 9: Your program (**Filtration Engine**) should automatically clean and transform the data stored in the files, and then upload each record to new MongodB database **myMongoTweet**

- For cleaning and transformation -Remove special characters, URLs, emoticons etc.
- Write your own regular expression logic. **You cannot use libraries such as, jsoup, JTidy etc.**

Step 10: You need to **include a flowchart/algorithm of your tweet cleaning/transformation program on the PDF file.**

Problem #2: This problem contains two tasks.

Task 1: Data Processing using Spark – MapReduce to perform count

Step 1: Write a MapReduce program **(WordCounter Engine)** to count (frequency count) the following substrings or words. Your MapReduce should perform the frequency count on the stored raw tweets files

- "**flu**", "**snow**", "**cold**"
- You need to include a flowchart/algorithm of your MapReduce program on the PDF file.

Step 2: In your PDF file, report the words that have highest and lowest frequencies.

Task 2: Data Visualization using Graph Database – Neo4j for graph generation

Step 3: Explore Neo4j graph database, understand the concept, and learn cypher query language

Step 4: Using Cypher, create graph nodes with name: "flu", "snow", "cold"

You should add properties to the nodes. For adding properties, you should check the relevant tweets Collections.

- Check if there are any relationships between the nodes.
- If there are relationships between nodes, then find the direction
- **Include your Cypher and generated graph in the PDF file.**

Assignment 4 Submission Format:
1) Compress all your reports/files into a single .zip file and give it a meaningful name. **2) Submit your reports only in PDF format.**
Please avoid submitting .doc/.docx and submit only the PDF version. You can merge all the reports into a single PDF or keep them separate. **You should also include output (if any) and test cases (if any) in the PDF file.**
3) Your Java code needs to be submitted on gitlab