Assignment 4

CSCI 5410 (Serverless Data Processing)
Date Given: Jun 28, 2022
Due Date: Jul 9, 2022 at 11:59 pm
Late Submissions are not accepted. A deduction of 10% per day will be applied for late submission.

To avoid any additional charges for resource consumption - Delete any AWS service, storage, database after fulfilling the assignment submission requirements

---

Objective:
This assignment will help you learn some key services of AWS platform. In this assignment, you are required to work on GCP ML, and AWS Comprehend.

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 - based on the discussion board rubric (McKinney, 2018)

	Excellent (25%)	Proficient (15%)	Marginal (5%)	Unacceptable (0%)	Problem # where applied
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	Part A Part B Part C
Correctness	All parts of the given tasks are correct	Most of the given tasks are correct However, some portions need minor modifications.	Most of the given tasks are incorrect. The submission requires major modifications.	Incorrect and unacceptable	Part A Part B Part C
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 evidence 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	Part A Part B Part C
Clarity	The written or graphical	The written or graphical materials,	The written or graphical materials,	Failed to prove the clarity. Need proper	Part A Part B

	materials, and developed applications provide a clear picture of the concept and highlights the clarity.	and developed applications do not show clear picture of the concept. There is room for improvement	and developed applications fail to prove the clarity. Background knowledge is needed.	background knowledge to perform the tasks.	Part C

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

All your work must be properly cited (in-line and a separate bibliography at the end)

Tasks:

This assignment has 3 parts. Part A is related to background study and report writing. Part B, and C are related to coding, development, and testing

Part A. Read AWS official documentation on SageMaker and Comprehend. Learn how these services are used in event-driven applications and explore various use cases.

Once you learn about the services, write 1 page summary on these services highlighting the usage, and an overview on how you will be using these services for "**Halifax bike rental application**".

Please ensure that you have included the proper citation. Do not copy paste any content from the source documentations. The summary must be written in your own words.

Part A - Submission requirement:

A pdf file with the summary, your approach, and citation.

Part B. Build an event-driven serverless application using GCP ML.

In this part of the assignment, you need to use GCP file storage, Cloud function.

[B00xxxxxx = your B00 number] used in bucket/GCP storage naming

take screenshots at every step and submit as part of the PDF:

a. Create your 1st storage bucket **SourceDataB00xxxxxx** and upload the files (from 001 to 299) given in the Train folder. You need to use the SDK to upload the files on the bucket.
b. Once a file is uploaded, a cloud function - "generateVector" should extract words from all the files (remove the stop words). Then compute Levenshtein distance*(https://en.wikipedia.org/wiki/Levenshtein_distance) between the Current, and Next word. Consider the sentence "Asia is a big continent"

E.g. If “Asia” is the current word, and “big” is the next word, then Levenshtein distance (or edit distance) will be “3”. **(remove the stop words)**

e.g. Contents of “trainVector.csv” should be

Current_Word	Next_Word	Levenshtein_distance
Asia	big	4
big	continent	8

c. This file (“trainVector.csv”) is saved in a new bucket **TrainDataB00xxxxxx**
d. GCP ML should get the training data for a clustering algorithm (KMenas) from the **TrainDataB00xxxxxx** bucket.
e. Once the training is done, like point (a), upload the test files given in the Test folder (300 to 401) to **SourceDataB00xxxxxx**
f. The cloud function **generateVector** computes the distance vector same as point (b) and store it in “testVector.csv”.
g. This file (testVector.csv) is saved in another new bucket, **TestDataB00xxxxxx**
h. GCP ML should get the test data for KMeans algorithm from the **TestDataB00xxxxxx** bucket.
i. Finally, write a code or configure a service to obtain information about clusters (e.g. centroids, cluster numbers, outliers etc. which are generated by GCP ML), and display the clusters.
j. Test your functions, and entire cluster generation process.

Part B - Submission requirement:

Submit screenshots of every steps. Please do not exclude any steps. Include all screenshots as part of a PDF file. In addition, provide the program/scripts as part of the PDF file. In addition, submit code in gitlab.

* If needed, you can use libraries for Levenshtein distance calculation. However, you muat add citation.

Part C. Build an event-driven serverless application using AWS Comprehend.

In this part of the assignment, you need to use S3 bucket, Lambda Functions, and AWS Comprehend.
[B00xxxxxx = your B00 number] used in bucket naming

take screenshots at every step and submit as part of the PDF:

a. Create your 1st S3 bucket **TwitterDataB00xxxxxx** and upload the given tweets file. You need to write a script or use the SDK to upload the files on the bucket.
b. To perform any pre or post processing of the files, you can write Lambda functions.
c. Once the file containing all the tweets is uploaded on the bucket, AWS Comprehend is used to perform sentiment analysis of tweets.

d. Your output should be captured in a csv or json format.

Data Source Acknowledgement: The tweets file was generated by Alapati Lakshmi Manjari, 2019.

Part C - Submission requirement:

Submit screenshots of every steps. Please do not exclude any steps. Include all screenshots as part of a PDF file. In addition, provide the program/scripts as part of the PDF file. In addition, submit code in gitlab.