ENSF 612: Assignment 3

Marks: 25

The assignment is worth 5% of course marks

Instructions:

1. This assignment must be completed individually. You cannot copy from others or consult with others. Any identification of such unauthorized actions may result in 0 grade for this assignment.
2. This is a take home assignment.
3. Each student will use Databricks community edition notebook to write the assignment. Once completed the student will share the notebook via the D2L drobox
4. To answer each question, use the notebook feature markdown to write your texts and the code blocks to write your code. All code needs to be executable. The grading will be done by reading the texts and the code and by running the code.

Question 1 (5 Marks)

"this is asignmnt 2 and it is jsut great!"

1. How can you fix typos in the above sentence using Levenshtein distance algorithm? (write a small python code to show)
2. For each typo, what kind of edit operation you need to do? (consider the three edit operations: insert, delete, replace)

Question 2 (20 Marks)

There are three files with 150,000 questions that are asked about three topics in Stack Overflow: ML, Spark, and Security. The files are shared in D2L (Assignment 1 files). The posts are collected from Stack Overflow posts table. Details about Stack Overflow posts table can be found here: https://meta.stackexchange.com/questions/2677/database-schema-documentation-for-the-public-data-dump-and-sede

For each topic, your task is to find the most similar question with an accepted answer to the highest scored question in the language that does not have any accepted answer.

Suppose, for ML, you had four questions with score (Q1: 3, Q2: 4, Q3: 5, Q4:3). Suppose Q3 and Q4 do not have any accepted answer. Then Q3 is the highest scored question without any accepted answer. Suppose Q1 and Q2 have accepted answer. You will then see how similar Q1 and Q2 are to Q3. Suppose Q1 has similarity value of 0.8 with Q2 has a similarity value of 0.2 with Q3. Then you will return Q1 as the output. Such approaches are normally used to recommend solutions to an unanswered question in Stack Overflow.

For each topic, you will write the function in python and you will return the top three most similar questions. You do not need to use Spark programming methods for this, but it's fine if you use. For similarity analysis, you will use cosine similarity. For similarity analysis, it is fine to simply use the textual contents in title and body. You will need to do standard preprocessing of the texts before computing similarity:

1. Tokenization into words

2. Stop words removal
3. Noise reduction (e.g., removal of punctuation)
4. Stemming/lemmatization

You are expected to write your own code to compute cosine similarity following the examples of its computation shown in the class lecture. You should not use an existing implementation of the metric in Python.