NYU CSGY 6513 Big Data

Assignment 1 - Hadoop HDSF & MapReduce – 120 points + Extra Credit

Due Date: ***Monday, Feb. 24 , 11:59PM Eastern***

Abstract:

Show proficiency using HDFS and writing MapReduce programs, including submitting to Hadoop and getting results out of HDFS.

Submission Rules

- Submit all code, pictures, pdfs and ***output*** files **as a ZIP**, using your **id and hw1:** *for example, jcr365-hw1.zip.*
- If we cannot run your program(s), you will not get full credit.
- Give attribution to any code you use that is not your original code. **If this involves any AI generated work, include the prompts and LLMs used, as well as give attribution and how it was used.**

1. HDFS 20 points

On any Hadoop (Dataproc, HPC, docker or otherwise), submit screen grabs (jpg, pdf, png) of the following:

a) create a directory **in HDFS** with this format: **hw1-netid** (e.g. mine will be 'hw1-jcr365').
Submit a screen grab of the output of a *Hadoop file listing* showing your home directory and your new directory in it.

b) Create a subdirectory 'data' in `hw1-netid`. Upload the homework file ***and extract*** it in the data folder. Submit a picture of directory listings or otherwise show the input files in this directory.

2. ID Tokenizer (100 points)

Use the word count shown in class as a starting point, or write your own, and **assign an increasing integer ID to each word (word*) in order of decreasing count*. ID's start at 1.**

For example, if the word counts compute to this:

the \t 123
house \t 99
is \t 88
on \t 76

Your output should be a table like this:

1 \t the
2 \t house
3 \t is
4 \t on

Hint: this will require more than one Hadoop job….

The input for this problem: **hw1text.zip** (provided in class website).
Rules for text:

- Normalize all words into lowercase
- Replace all non alphanumeric characters with a space, that is anything not in 0-9 and a-z should be replaced with a space before tokenization.

Problem 3 Extra Credit: Word probability 100 points:

Output the probability of the words with IDs = 10 and 15.

The probability of a word, using Maximum Likelihood Estimation, is given by the word counts.

$$P(word_i) = \frac{Count(word_i)}{total\ words\ seen}$$

Hint: You do not need to compute probabilities. The denominator, $total\ words\ seen$, is constant across all words. So, you can just word-count **and count** words seen. Recall we cannot hold state, so this is a >=2 job problem. You do not to solve the division programmatically. That is, you can hard code the denominator in the next job that needs it.

The input for this problem: **hw1text.zip** (provided in class website).
Rules for text:

- Normalize all words into lowercase
- Replace all non alphanumeric characters with a space, that is anything not in 0-9 and a-z should be replaced with a space before tokenization.