D 351 Mini Project 1

Due: September , 2023

MapReduce Algorithm for Word Counting

In this project, our aim is to employ the MapReduce algorithm to solve word counting problem. Each student has one of two options. You can either choose to implement the MapReduce algorithm in Python so as to perform a word counting task on the provided text file; or, you can use *Apache Spark* to harness the same algorithm for the same task.

Data

You will be working on a dataset called *Fraudulent E-Mail Corpus*, which is a structured text file composed of more than 2,500 fraud letters. In order to gain more information about and download the dataset, please visit the website `https://www.kaggle.com/datasets/rtatman/fraudulent-email-corpus`. On the upper right corner of the page you will see the *Download* link.

Instructions for Python/PySpark Implementation

1. Download and extract the dataset. You must now have a text file named *fradulent_emails.txt*. This will be the input to your program.

2. Your main task is to write a python program that counts the number of words in the input file. You may choose to use PySpark package to utilize the Spark environment, or to implement MapReduce algorithm without using Spark. There are materials related to PySpark available in Module-0 in Modules section in the Canvas page that give the basics on PySpark. Note that this task could be performed in many different ways (like using a dictionary data structure). **We are asking for a very particular one: a program that implements the MapReduce algorithm.** In particular,

    - you will need to think about how to perform the Map task on the input data,

- then how to group the resulting (key, value) pairs in a form accessible by the Reduce task,
- and finally how to process the Reduce task to obtain the required output.

Even though you are free to use whichever data structures you like, or whichever programming style (like object-oriented programming) you prefer, there is one important constraint: you are **not** allowed to collect the (key, value) pairs resulting from the Map operation in a dictionary-like data structure, which is inherently sorted. We expect to see how you handle each key component of the implementation.

3. The first question you are required to answer is this: what are the most frequently used twenty words? In other words, please count the twenty most frequent words in the dataset.

4. You know that your dataset is composed of phishing emails. Do the most frequently used twenty words you found above reflect this nature of your dataset? In other words, just looking at the list of those twenty words and their counts, could you guess that you were dealing with fraudulent emails? Why, or why not?

5. How can you improve this *counting the most frequently used words* technique so as to make your program a better spam-detection tool? Briefly explain.