CS 4740 Programming Assignment 3

Note: you can pause any time during this assignment.

Goals:

Gain hands-on experience with the MapReduce framework. MapReduce, initially developed by Google (see paper below), is a programming paradigm used primarily in Hadoop systems.

MapReduce provides a more efficient way to process a large amount of data. This assignment will have three parts. The first is to introduce MapReduce and the basics of ‘mr\_job’ with the Iris dataset. We will then explore a larger data file (a book).

This assignment will use Python and the built-in ‘mrjob’ library. Link to original Google paper:

<https://www.usenix.org/legacy/events/osdi04/tech/full_papers/dean/dean.pdf>

Step 1: (In each of the following steps, you need to refer to the instructions in this step)

Follow the instructions in the official ‘mrjob’ tutorial: <https://mrjob.readthedocs.io/en/latest/>

While there is a lot of useful information in this tutorial, focus specifically on the installation, “Why mrjob”, “Fundamentals”, “Concepts”, and “Writing Jobs” sections.

If you have never used Python before, you may want to refer to <https://www.youtube.com/watch?v=YYXdXT2l-Gg>(Windows and Mac) or <https://www.youtube.com/watch?v=UXjjEZroOu0>(Linux). You can find a lot of beginning tutorials online.

Step 2:

Download the Iris dataset from the following link:

<https://gist.githubusercontent.com/curran/a08a1080b88344b0c8a7/raw/0e7a9b0a5d22642a06d3d5b9bcbad9890c8ee534/iris.csv>

Read the description file to familiarize yourself with the dataset.

Using Python and mrjob, write a program to find each of the following metrics:

1. the minimum sepal length
2. the maximum petal width
3. the average sepal width for the class “Iris Setosa”
4. the difference in average sepal and petal length for all non-“Iris Setosa” Please refer to the end of this document about what needs to submit.

Step 3:

Now that we have a basic familiarity with MapReduce, we will move to a different example. Please download the attached “harry.txt” from your Collab assignment PA3 link.

Calculate the word frequency for the text file. The word frequency is defined by the number of times a word appears in the book. Specifically, please find the number of occurrences for the following words: ‘magical’, ‘soaring’, and ‘lopsided’ and include the results in your final submission.

You may have noticed that the output list of word frequencies includes the same word with additional punctuation (e.g., "late", "late,", "late.\"). Please refer to the section “Writing your second job” in the ‘mrjob’ tutorial above or search “python remove punctuation” in Google to solve this problem and only show the word without punctuation in the output.

Step 4:

Using the structure from step 3, you can find out the frequencies of the words in *Harry Potter and the Prisoner of Azkaban*. Please list the top ten most frequently used words and their associated word count in the final results.

Hint: You may have noticed that to find the top ten most frequently used words, the previous issue in step 3 with punctuation must be fixed. You may also want to use more than one mapper/reducer (the additional mapper/reducer is for sorting) or more than one program (the second program prints out the final result). Please refer to <https://mrjob.readthedocs.io/en/latest/guides/quickstart.html#writing-your-second-job>for the multi mapper/reducer.

Submission:

Please include ONE PDF in the following format:

* The first page should contain a list of answers to each of the scripts written above in order (in steps 2-4):
  + Example:

- 1) 4.57

- 2) 3.59, etc.

* The second page (and onwards) should contain the corresponding list of python scripts you used to calculate the answers
  + Example:

- 1)

def print\_first():

print(4.57)

Important notes:

* All these programs (especially in the steps 2 and 3) can be done iteratively without using MapReduce. However, to receive credits on these parts, you must perform your calculations using MapReduce.
* Please start this assignment several days in advance of the deadline. This is largely due to the number of programs being written (though most follow a similar general structure)
* When running these programs, include the name of the data file (e.g. ‘harry\_potter.txt’) in the command line argument