ENSF 619.28
Assignment 1 (10%)
Due 12 am Saturday February 15, 2019
This assignment is to be completed individually

Question 1- Building n-grams using Hadoop MapReduce (15 marks)

An n-gram is a contiguous sequence of n items in a sequence of text or document. Computing n- grams has many applications including natural language processing, modeling languages, and speech recognition. For example, calculating the counts for various n-grams can help an AI program predict the next word given a speech pattern.

For this problem, you will use Python and Hadoop streaming to build counts for a word-level 2- gram or a di-gram. Your program should be able to scan a set of text documents and then record all unique 2-word sequences in the document. As a final output, it should list all the unique di- grams along with their counts.

Your program should be scalable to large datasets and hence should use MapReduce. You need to test your program using the romeo-and-juliet.txt file

Deliverables: Need to individually demonstrate working code to TA. Need to submit code via D2L.

Question 2 – Yearly Car Accidents (15 marks)

For this problem, you are going to calculate the number of yearly accidents in Canada between 1994-2014. You should only include accidents that happened in a sunny day by a person of age less than 50 years and of which the accident was fatal.

Your program should be scalable to large datasets and hence should use MapReduce. You need to test your program using the files in the "cars-accident-stat" uploaded on D2L.

Deliverables: Need to individually demonstrate working code to TA. Need to submit code via D2L.

Question 3 - Sorting using Hadoop MapReduce (10 marks) - BONUS

For this problem, you are going to sort words found in a large text document. Specifically, your program should scan the document and then output the words in the document in ascending order. Make sure you remove punctuations in the text document and convert to lower case letters as part of your analysis.

Your program should be scalable to large datasets and hence should use MapReduce and multiple reducers. Single reducer solutions will only get partial credit since they won't be scalable. Part of the challenge involved is to deduce the correct number of reducers that will result in a total order sort, i.e., results appearing sorted across all reducers.

Deliverables: Need to individually demonstrate working code to TA. Need to submit code via D2L.