**HU Extension Assignment 04 E63 Big Data Analytics**

Issued on: February 20, 2016 Due by 11:30PM EST, February 26, 2016

**Problem 1.** Write a working WordCount program using Spark Java API that reads a file, e.g. Ulysis/4300.txt from an HDFS directory and writes the results of your calculations to an HDFS file. To improve your word count, remove any punctuation that might have attached itself to your words. Also transform all words into lower case so that the capitalization does not affect the word count. The original code used in lecture notes is provided in the attached mini-example-java.tar file. That archive also contains Maven’s pom.xml file. Run your program and demonstrate that it works. Submit working code inside the customary MS Word Document. Describe steps in your program.

**Problem 2.** Write a working WordCount program using Spark Scala API that reads a file, e.g. Ulysis/4300.txt from a local file system directory and writes the results of your calculations to a local file. To improve your word count, remove any punctuation that might have attached itself to your words. Also transform all words into lower case so that the capitalization does not affect the word count. The original code is provided in the attached mini-example-scala.tar file. That archive also contains Scala Build Tool build.sbt file. Run your program and demonstrate that it works. Submit working code inside the customary MS Word Document. Describe steps in your program.

**Problem 3.** Write a working WordCount script using Spark Python API. Read Ulysis (4300.txt) file from an HDFS directory and write the results of your calculations to an HDFS file. To improve your word count, remove any punctuation that might have attached itself to your words. Also transform all words into lower case so that the capitalization does not affect the word count. Run your script using submit-spark tool and demonstrate that it works. Submit working code. Describe steps in your program in the MS Word document.

**Problem 4**. In a Spark API of your choice, write a working BigramCount program which would count occurrences of every pair of consecutive words. You should clean your words just as you did in the previous problem by removing punctuations and cases. However, do not count two words separated by a point at the end of a sentence as a bigram. If you are an experienced programmer add to the bigram count word pairs in which the first word is the last word on the line and the second word is the first word on the subsequent line. If you are not an experienced programmer, than do not do it. Test your program on a small text file, where for comparison, you could identify bigrams manually. Run your program on Ulysis(4300.txt) file and demonstrate that it works. Provide us with the total count of your bigrams, first 20 bigrams and all bigrams containing word “heaven”. Read your file from the local operating system and write results to the local operating system. Include working code in the MS Word Document. Submit the file with the complete working code separately. Describe steps in your program in the MS Word document.

Please, describe every step of your work and present all intermediate and final results in a Word document. Please, copy past text version of your command. We cannot retype text that is in JPG images. Please, always submit a copy of original, working scripts and class files you used as separate files. Sometimes we need to run your code and retyping is too costly. Please, submit to the class drop box. For issues and comments visit the class Discussion Board .