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

Issued on: Sept 24, 2017 Due on Saturday by 11:59 AM EST, Sept 30, 2017

You can do these problems in the language of your choice: Python, Scala, Java or R.

**Problem 1.** Consider two attached text files: bible.txt and 4300.txt. The first contains ASCII text of King James Bible and the other the text of James Joyce’s novel Ulysses. Use Spark transformation and action functions present in RDD API to transform those texts into RDD-s that contain words and numbers of occurrence of those words in respective text. From King James Bible eliminate all verse numbers of the form: 03:019:024. Eliminate from both RDDs so called “stop words”. Please use the list of stop words on Web page: <http://www.lextek.com/manuals/onix/stopwords1.html>. . Create RDD-s that contain only words unique for each of text. Finally create an RDD that contains only the words common to both texts. In latest RDD preserve numbers of occurrences in two texts. In other words a row in your RDD will look like (love 45 32). List for us 30 most frequent words in each RDD (text). Print or store the words and the numbers of occurrences. Create for us the list of 20 most frequently used words common to both texts. In your report, print (store) the words, followed by the number of occurrences in Ulysses and then the Bible. Order your report in descending order starting by the number of occurrences in Ulysses. Present the same data this time ordered by the number of occurrences in the Bible. List for us a random samples containing 5% of words in the final RDD. We are just practicing RDD transformations and actions. You could implement this problem in a command shell or as a standalone program.

**(30%)**

**Problem 2**. Implement problem 1 using DataFrame API. You could implement this problem in a command shell or as a standalone program.

**(20%)**

**Problem 3**. Consider attached files transactions.txt and products.txt. Each line in transactions.txt file contains a transaction date, time, customer id, product id, quantity bought and price paid, delimited with hash (#) sign. Each line in file products.txt contains product id, product name, unit price and quantity available in the store. Bring those data in Spark and organize it as DataFrames with named columns. Using either DataFrame methods or plain SQL statements find 5 customers with the largest spent on the day. Find the names of the products each of those 5 customers bought. Find the names and total number sold of 10 most popular products. Order products once per the number sold and then by the total value (quanity\*price) sold.

**(30%)**

**Problem 4**. Implement problem 3 using RDD APIs.

**(20%)**

Please, describe every step of your work and present all intermediate and final results in a Word document. Please, copy past text version of all essential command and snippets of results into the Word document. We cannot retype text that is in JPG images. Please, always submit a separate copy of the original, working scripts and/or class files you used as separate files. Sometimes we need to run your code and retyping is too costly. Please include in your MS Word document only relevant portion of the console output or output files. Sometime either console output or the result file is too long and including it into the MS Word document makes that document too hard to read. PLEASE DO NOT EMBED files into your MS Word document. Please, submit to the class drop box. For issues and comments visit the class Discussion Board. You can solve these problems using any language of your choice.