DSSP16

Spark Exercises

These exercises aim to guide you towards building your first simple applications in Apache Spark using Python. Definitely, you will encounter more difficult and challenging problems, however, since for many of you these are your first steps in Spark, it is important to understand the most important concepts as gently as possible.

Exercise 01
Given a text document, write a Spark application to perform the following tasks:
- remove punctuation marks,
- keep the words with at least 4 characters,
- find the number of occurrences of every word,
- sort the words in decreasing order based on the number of occurrences,
- print the first 10 words.

Hint. Start from the **wordcount** application and change the code in order to achieve the required functionality. Test your ideas one-by-one and verify that the result is correct, before you move to more complex solutions. You may work with the file **/dssp/data/leonardo/leonardo.txt** to test your application.

(The solution is contained in **sparkx01.py**)

Exercise 02
Given two text files, find the set of common distinct words and print them out. Assume that the set of common words is small enough to fit in the main memory of the driver.

Hint. The two files may be given as parameters in the main function of your application. However, if it is more convenient, you may fix the filenames in your code. For convenience you may use the same text file twice, e.g., ***/dssp/data/leonardo/leonardo.txt***

(The solution is contained in **sparkx02.py**)

Exercise 03
A probabilistic graph is defined as a graph $G(V,E)$, where each edge e exists with probability p(e). This means that the existence of an edge is not certain, but it is based on other events. Given a probabilistic graph, isolate all edges that have a probability at least T, and then compute the average degree of every node. The average degree of a node is defined as the sum of the probabilities of the edges adjacent to this node. The input dataset **/dssp/data/collins** contains a graph represented as an edge-list file, where each line contains the node identifiers and the probability.

Hint. Start from wordcount, and change the code in order to meet the requirements.

(The solution is contained in **sparkx03.py**)

Exercise 04
A purchase transaction is defined as a set of products purchased together by a customer (e.g., in a supermarket). The input is a file that contains one transaction in each line, where products are seperated by comma (‘,’). We are interested in finding groups of 2 products that are purchased together. We need the product groups that have a number of occurrences more than a specified threshold T (e.g., T=20). The input dataset is located at **/dssp/data/groceries**

(The solution is contained in **sparkx04.py**)