Assignment 2

Tools & Techniques for Large-Scale Data Analytics (CT5105),
NUI Galway, Year 2015/2016, Semester 1

- ***Submission deadline (strict): Wednesday, 28th October, 23:59***
- *Put all your code files into a single .zip archive "YourName_Assignment2.zip" and submit via Blackboard*
- *Include all source code files required to compile and run your code*
- *Use Java 8 as programming language*
- *Please note that all submissions will be checked for plagiarism*
- *Use comments to explain your source code. Missing or insufficient comments can lead to mark deductions*
- *You will likely require the Apache Spark documentation: http://spark.apache.org/docs/latest/index.html*

Question 1 [no marks]

Install Apache Spark, as explained on the lecture slides (single-machine installation, without Hadoop). Try out some of the Java-examples included with Spark (folder examples\src\main\java\org\apache\spark\examples) to test your installation.

Questions 2 [50 marks]

Same as Assignment 1 Question 3[1] (temperatures counting using MapReduce), however, now use Spark and RDDs instead of Java-streams/collections. Filtering, mapping, reducing... should now be RDD operations.

Remarks/hints: Remember that MapReduce requires key/value pairs as map results (even though this seems redundant for such a simple task) and that operations should be performed in parallel where possible.
Take a look at JavaWordCount.java in Spark's example folder and Lecture 5/6 to see how another (but similar) counting task is implemented using MapReduce with Spark. For full marks you should use lambda expressions (which make MapReduce and other code significantly simpler and easier to read) instead of anonymous classes.
To create an RDD of objects of a certain class (e.g., Measurement), that class needs to implement interface Serializable.

Question 3 [50 marks]

A typical use case for Data Analytics is *sentiment analysis* (a.k.a. *opinion mining*), i.e., the computational determination of the attitudes of people towards a certain topic or object.
Download the following data archive:
https://archive.ics.uci.edu/ml/machine-learning-databases/00331/sentiment%20labelled%20sentences.zip
Dataset imdb_labelled.txt in this archive contains a number of single-sentence movie reviews, each labelled with "0" (negative sentiment) or "1" (positive sentiment).
Create a program which trains a linear Support Vector Machine (SVM) using any 60% of the labelled sentences in imdb_labelled.txt, using Spark MLlib. Afterwards, use the learned model to predict and print the labels (sentiments) of a few test movie reviews (which you can take from the remaining 40% of the data file).

Remarks/hints: As always, use lambda expressions to pass functions into higher-order Spark operations such as map.
If you like, you can let your program compute the ROC AUC, but this is not required for this assignment.
You don't need to understand how the SVM algorithm actually works to do this questions.
Question 3 can be solved independently from Question 2.

[1] If you didn't complete this question in Assignment 1, the lab tutor can show you non-Spark demo code to get started.