# Assignment 4

Tools & Techniques for Large-Scale Data Analytics (CT5105)

NUI Galway, Academic year 2019/2020, Semester 1

- ***Submission deadline (strict): Sunday, 10th November, 23:59.*** *Late submission only with a medical or counsellor's certificate.*
- *Put all source code files (and other answers, if any) into a single .zip archive with name "YourName_Assignment4.zip" and submit via Blackboard by the deadline.*
- *Include all source code files (that is, files with name ending .java) required to compile and run your code, including all sources for your classes, interfaces, etc.*
- ***Please include the question number in the name of the respective .java file or package*** *(e.g., "MyClass_Q1.java")*
- *Unless specified otherwise in the question, use only plain Java together with Apache Spark for this assignment.*
- *All submissions will be checked for plagiarism.*
- ***Use Java comments to explain your source code. Missing or insufficient comments lead to mark deductions.***

**Question** [max. marks: 100]

Create a program which uses Spark to cluster given Twitter tweets by their geographic origins (coordinates), using the *K-means* clustering algorithm. Use RDDs/JavaRDDs as far as possible.

You are given data file `twitter2D.txt`[1] with Twitter tweets and their attributes. The first two values in each line are the world coordinates from which the respective tweet was posted. The other values are a time stamp, a user id, an optional flag 1=spam/0=no spam, and finally the actual tweet message.

Your program should learn a K-means clustering with four clusters from this file. From each line in the file, only the coordinates are required as features for learning. Use all coordinates in the file to train the model.

Finally, let your program print every tweet (message) in the given file together with its respective cluster index (that is, the number of the cluster which contains that tweet's coordinates, according to the learned model), sorted by the cluster indices. Tweets in the first cluster should be printed first, then those in the second cluster, and so on.
E.g., the output of the program might look like this[2]:

```
Tweet "..." is in cluster 0
Tweet "..." is in cluster 0
Tweet "..." is in cluster 1
Tweet "..." is in cluster 1
Tweet "..." is in cluster 1
Tweet "..." is in cluster 2
Tweet "..." is in cluster 2
...
```

Hint: Studying the example code in a recent lecture might be helpful.

[1] on Blackboard under "Assessment"

[2] The cluster numbers in the example output above are fictitious.