**Teodor Chakarov, Number: 12141198, Data-intensive Computing**

**Exercise 2: Text Processing and Classification using Spark**

# Part 1 RDDs

In this part I made a solution analogically to Apache Hadoop tool, which was used in Exercise 1. With the help of API I managed to output the calculate chi values in output\_rdd.txt. Where I sorted the values for each category, the words sorted by descending chi value. Those chi values are being calculated based on the occurrences of the words in the document. Here the difference will occure with Part 2.

# Part 2 DataFrames: SparkML and Pipelines

Spark provides 2 different APIs for working. The first is with RDD and the second one (analogically to the ML library scikit-learn) working with Data Frame.

In this part I am using just tools that provides smooth and easy way of working with data. Pipeline is data streaming tool, created for stacking steps on top of each other and the end-user not bothering this the details.

The difference in Part 2 in comparison to Part 1 is that before we calculate the chi values, we are using here not only TF (term frequencies) but IDF – (inverse document frequency). This calculates a value how important is each word not only based on the occurrence in one document but in whole corpus of text. After that I used the Normalization technique to scale the numbers between 0 and 1. Based on those numbers we calculated the chi square. And the outcome is more different because the top words are picked by the help of more advanced technique in NLP in comparison to Counting Vectorization (occurrence of each word in the document).

# Part 3 Text Classification

As described in the notebook for parts 2 and 3, I tried to train Support Vector Machines (powerful but very slow algorithm for big data training). With help of Chi values I managed to achieve bad result around 52% F1 score. I tried using also Logistic Regression but I have the same results. Based on that I will conclude that those models are under fitting the data. (Since the data is high dimensional (2000 attributes)

That’s why I tried with only TF-IFD strategy, and L2 Normalization of the values. On that test with SVM again, I ached a really good score on both validation and testing sets (98%). The data was not reduced, high dimensional.

I tried to fit PCA for reduction, but had some problem with the computation of this step. But PCA is also a technique of reduction of dimensions.