# SPAM FILTER

*Big Data Technologies – Master EIT Digital in Data Science*

Spam classification of emails based on ling-spam dataset

Ignacio Uyá, Marcos Bernal, Yolanda de la Hoz

i.uya.lasarte@gmail.com,marcosbernal3@gmail.com,yolanda93h@gmail.com

# Index

# Introduction

In this report it is described the Spam Filter application based on ling-spam dataset. This application has been developed with Spark and Scala in order to demonstrate its capabilities processing large datasets in a distributed manner.

In this report, first it is described the process/algorithm to classify emails and then it is described the methods used from scala and spark to optimize this process in a distributed manner.

# Spam Filter Algorithm

Mutual information factor is computed to know which is the gain of information, considering the word frequency. In order to compute this parameter, it’s needed to process the set of documents from ling-spam dataset. Then it is computed the probabilities of occurrence for each Word considering also each class and finally this probability is used to know exactly the frequency of occurrency for a given word in the whole dataset.

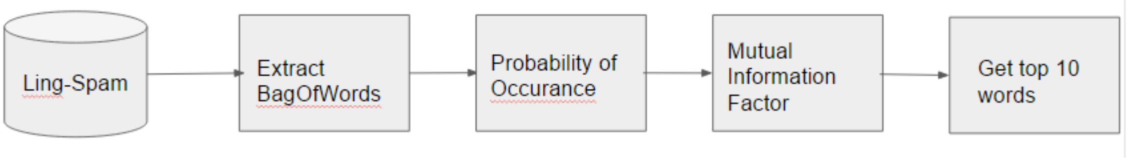

*Figure 1. Spam Filter Process*

# Algorithm implementation

For the implementation of this process, the application made of usage of Spark and Scala framework. This framework allow us to make operations over a fast access data structure called RDD (Resilient Distributed Dataset) distributed in a cluster of machines in a fault-tolerant way.

For the implementation of this algorithm, it is followed a Map and Reduce paradigm which allow us to parallelize the computational operations over different machines. The following list shows the main structures maintened in algorithm process and the spark operators used to optimize this process.

1- ***Bag of Words.*** It is used the map operator for each file to split each file separately and then combine them in a set to remove non informative words with the diff operation.

2- ***probaWord.*** It is used the map operator for each word to set 1 value and it is counted the number of words in each file using the a mapreduce approach combining the results using the reduceByKey operator.

3- ***Partial MutualInformationFactor for each class and occurrency.*** In this function, first it is computed the different probabilities to apply (probaWC, probaW and probaC) the mutual information formula, as it is explained in the report. This function makes usage of leftOuterJoin in order to join probaWC and probaW with the same key and apply the function with the getOrElse function to sustitute the default value in case of probaWC has 0 value which is the value that the join operator assign because that word doesn't occurs in the other class.

4- ***MI factor***. Finally, it is used the union operator and reduceByKey in order to sum all the values  and ordered to obtain the top10 words.

## Algorithm Applications – Question 6

Once that it is obtained these top words, they can be used to classify emails in ham or spam depending on the number of occurences of these top words in that emails. So if an email has a big frequency of these words it can be classify to spam and use also the rest of the words of this email to classify and relate new emails.

The top 10 words that the application spam filter has return:

```
(!,3.441014732541008)
(Subject:,2.8507939401890723)
(free,2.5327935186140924)
(our,2.2996035960092933)
(remove,2.0684180140905113)
($,2.014498552690443)
(mail,1.7444374781160021)
(money,1.7182948153970663)
(com,1.7144612747691124)
(day,1.6971763757396217)
```

These words has the greater mutual information factor, which means that it appers rarely in the ham emails and they are also very frecuent in spam emails, so there is a big probability that emails containing these words include spam messages inside.

This is the result of applying this function over the spam-ling dataset. Nevertheless, the result shows words such as money and $ that can be very frequent in emails related with bank or economics. So maybe, this algorithm could be improved using specific datasets related to that topics that appears to have also this “ham words” and related them with other words that appear to be very frecuent also in such emails using the mutual information factor with the same spam dataset but changing the ham dataset.