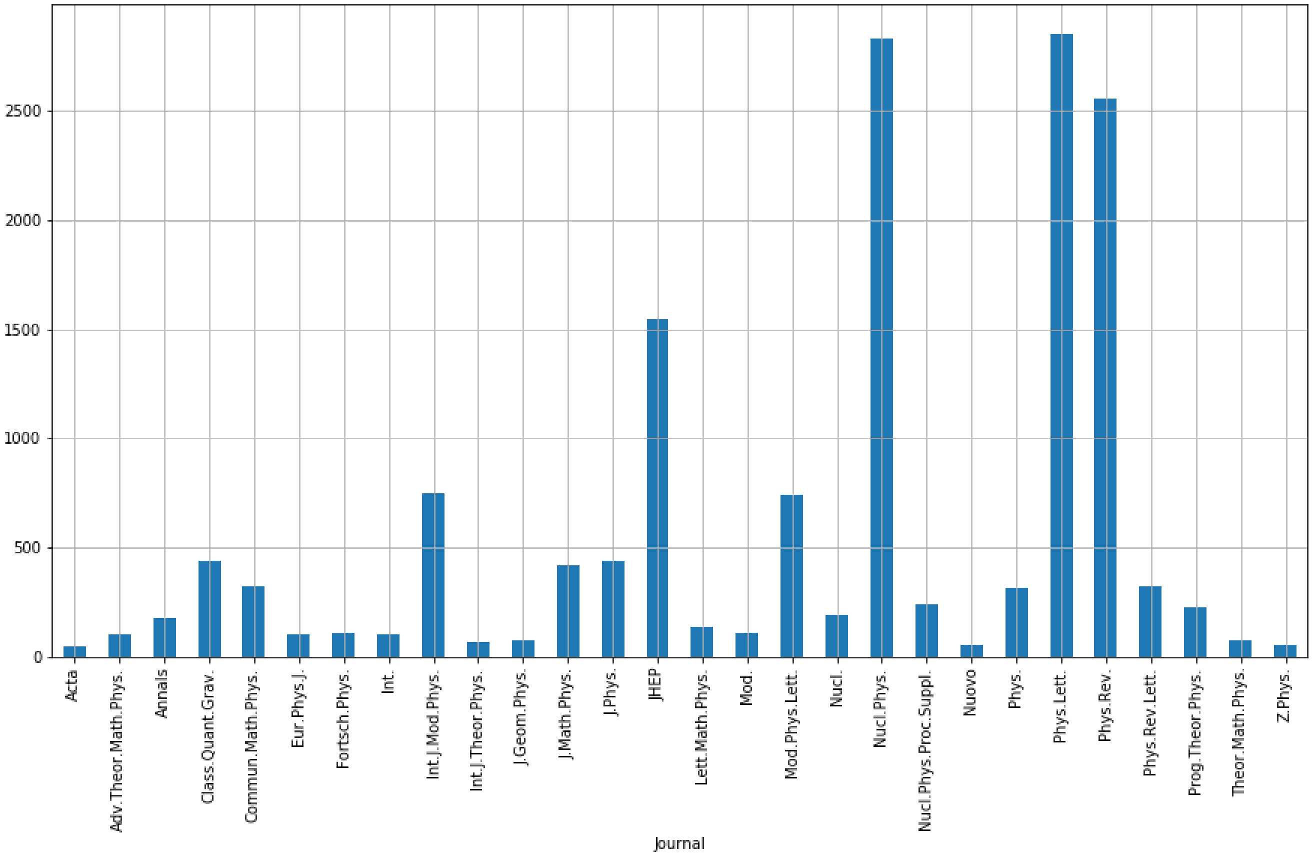
Jesús Bujalance Martín
Antoine Menand

SPEIT - DATA MINING REPORT

THE DATASET

The goal was to classify a dataset of research papers from high energy physics into 28 different categories (the journal where they were published).

We had two different types of information:
- a citation network
- an abstract from each article



We decided to study text (NLP) and graph techniques separately, hoping that a combination of both approaches would give us the best result.

NLP APPROACH

We implemented two different methods:

1. Tf-idf Vectorizer with stemming

Our first idea was to improve the code given as an example.

Features :
Instead of using a CountVectorizer, we used a TfidfVectorizer, because we believe that the tf-idf weightening provides with the most useful information since it balances rarity and frequency of the terms in the collection of articles. The input of the TfidfVectorizer was no longer just the abstract, but also the author and the title of the article, which gave us more information.

Feature extraction :
We tried to reduce the amount of features using PCA, but the results were not as good.

Classification algorithm :
We tried three different algorithms.
We tried two simple models, a naive bayes classifier and a logistic regression. These algorithms gave us good results, but the best results were obtained with linear SVM classifiers. On the Internet we found that SVM classifiers were used very often in NLP problems.

Results:
These method gave us the best result among the 4 different approaches that we tried. We obtained a logloss of **2.14** with the logistic regression and **2.002** with the SVM. The latter was also significantly longer.

2. Doc2Vec

We wanted to implement an embedding approach. Doc2Vec is in the library gensim, and is inspired by the algorithm Word2Vec.
The basic idea behind Word2Vec is to find a vector of features that represent each word, taking into consideration the words that are around. In order to get a single vector for an entire document, we could have made an average of the Word2Vec vectors of each word in the document, but we would have lost a lot of information doing this. Doc2Vec is a more sophisticated algorithm, so we decided to use it.

Features :
When training our Doc2Vec model, we set the size of the features vector to 100

Feature extraction :
We did not apply any feature extraction method. The Doc2Vec embedding allows the user to specify how long the feature vector is.

Classification algorithm :
We used a SVM classifier as well, for it was widely used in the examples we could find on the Internet.

Results:
These method gave us better results than the CountVectorizer basic method, but we obtained a logloss of **2.4,** which was quite disappointing. However, it is true that we are not familiar with the gensim library and it might be possible that we haven't implemented Doc2Vec properly.

GRAPH APPROACH

Our idea was to implement a **spectral clustering** algorithm in two steps : a spectral embedding and a k-means clustering.

We ran into a problem when coding this method. Since our graph is a directed graph, its laplacian matrix L = D – A is not a positive semi-definite matrix, which means that its eigen values aren't all positive so the algorithm doesn't work. Therefore we chose an undirected representation of the graph, loosing a lot of important information.

Results:
We achieved a logloss of **2.67**. However I feel like this technique is more appropiate for unsupervised learning, since in the end we did not use the labels of a training data much. It's still a very fast method, and if we know how to implement it for directed graphs the results might have been much better.

MIXED APPROACH

Lastly we tried to combine the features from the Tf-Idf Vector and the graph properties (the 3 features given in the starting-kit script). However the results were worse than the Tf-Idf Vector alone.

CONCLUSION

Overall, the Tf-Idf Vectorizer with stemming was the best algorithm.
Most of our more sophisticated attempts failed, either because we did not use all the information available (spectral graph clustering) or because they are not as good (Doc2Vec). Our attempt to combine the information coming from both sources also failed, mainly because our graph-based algorithm only had 3 features compared to the huge vector of features used for NLP.