**Project IA Documentation**

1. The purpose of the project is to implement a solution in order to classify the topic of a text: culture, finance, politics, science, sports, tech. The implementation is written in Python
2. The dataset can be found at <https://github.com/butnaruandrei/MOROCO> , where are over 20000 texts labeled.
3. The first step is to make a vectorial representation of the texts:

BOW (bag of words) : implemented in main.py and documented from <https://www.freecodecamp.org/news/an-introduction-to-bag-of-words-and-how-to-code-it-in-python-for-nlp-282e87a9da04/>

TD-IDF Implemented in tdidf.py and documented from <https://towardsdatascience.com/natural-language-processing-feature-engineering-using-tf-idf-e8b9d00e7e76>

Word2vec implemented in main.py

DBOW(distributed Bag of words) <https://towardsdatascience.com/multi-class-text-classification-with-doc2vec-logistic-regression-9da9947b43f4>

1. The second step is to implement a classifier:
   * + - 1. **Logistic Regression:** First, Doc2vec is an adaptation of Word2Vec that allows us to learn document similarity. Doc2vec model by itself is an unsupervised method.

We will use the gensim library for doc2vec.

We transform the text in DBOW, we eliminate the stop words and preprocess the text and then we classify our train datas(70% from all datas).

DBOW is the Doc2Vec model analogous to Skip-gram model in Word2Vec. The paragraph vectors are obtained by training a neural network on the task of predicting a probability distribution of words in a paragraph given a randomly-sampled word from the paragraph.

Accuracy is: 0.7980356046654389 for 21.000 instances time of running: 15 minutes

0.8063426986615218 for 27.000 instances, time of running 20 minutes

<https://towardsdatascience.com/multi-class-text-classification-with-doc2vec-logistic-regression-9da9947b43f4>

<https://github.com/susanli2016/NLP-with-Python/blob/master/Doc2Vec%20Consumer%20Complaint.ipynb>

\*\*Put the implementation in MOROCO-master in order to have the dataset.