PRI Report

Group 4

Nelson Nunes  
 DEI  
Instituto Superior Tecnico  
 [nelson.filipe@tecnico.ulisboa.pt](mailto:nelson.filipe@tecnico.ulisboa.pt)

André Fernandes  
 DEI  
Instituto Superior Tecnico  
[andre.diegues@tecnico.ulisboa.pt](mailto:andre.diegues@tecnico.ulisboa.pt)

Miguel Reis  
 DEI  
Instituto Superior Tecnico  
 [TODO](mailto:nelson.filipe@tecnico.ulisboa.pt)

ABSTRACT

TODO

Introduction:

In this paper we describe our work for the Process and Retrieval Information course. Three tasks were addressed: Ad hoc Search, Text Classification and Named Entity Recognition with statistical analysis. The Datasets used were extracted from the Manifest Project dataset. For both task one and three, the English version was used to better use the existent English vocabulary NLP tools. For the second task, the Portuguese version was used, in this way as we know more about Portuguese Politics, we could better analyze our results.

1 Ad-Hoc Search:

2 Text Classification:

We can divide this task into two subtasks: text preprocessing and method implementation.

For text preprocessing we consider the following strategies: the NLTK Portuguese stemmer, the NLTK Portuguese stopwords and the use of CounterVectorizer or Td-idf Vectorizer from the sklearn library, the use of different min and max frequency, and the number of ngrams extracted.

For method selection we tried a variety of approaches: three different Naïve Bayes implementations (BernoulliNB, MultinomialNB, GaussianNB), SGDClassifier, LinearSVM, a Deep Neural Network (Multi-Layer Perceptron), a Convolutional Neural Network, and a Voting Emsemble Method build with a NaïveBayes, SVM, and RandomForest classifiers).

For some of the previous methods, the RAM requirements (specially for bigger ngram max values) were enormous, so we implemented a mini-batch approach using the partial-fit function of the sklearn.

We performed a grid search for these parameters and the results can be view under the folder Results in the Ex2 folder. More explanation of these results in section 4. The script we coded for this testing was classificationTesting.py

After analyzing the results the architecture selected doesn’t use stopwords or stemming, uses td-idf normalized with min frequency of 0.001 and max frequency of 0.5, with n-grams:1,2,3, and the model is a SVM using linear kernel.

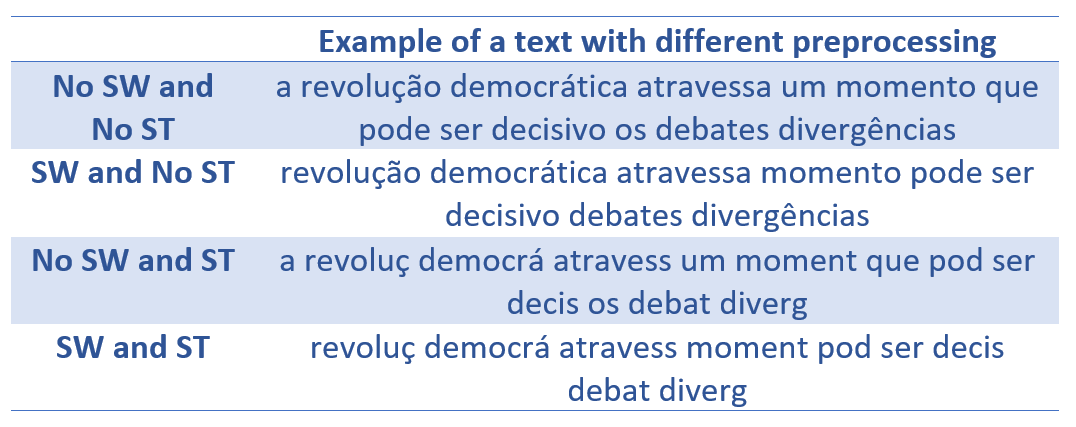
3 Named Entity Recognition:

4 Results:

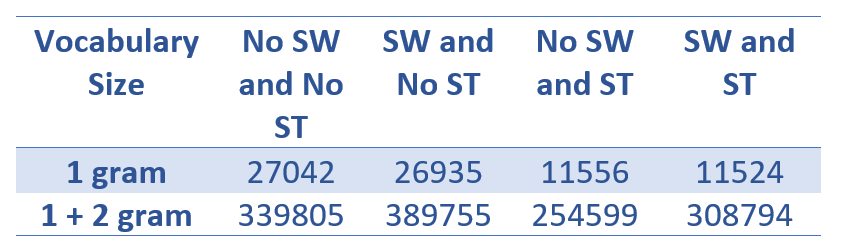
4.1 Ad-Hoc Search:

4.2 Text Classification:

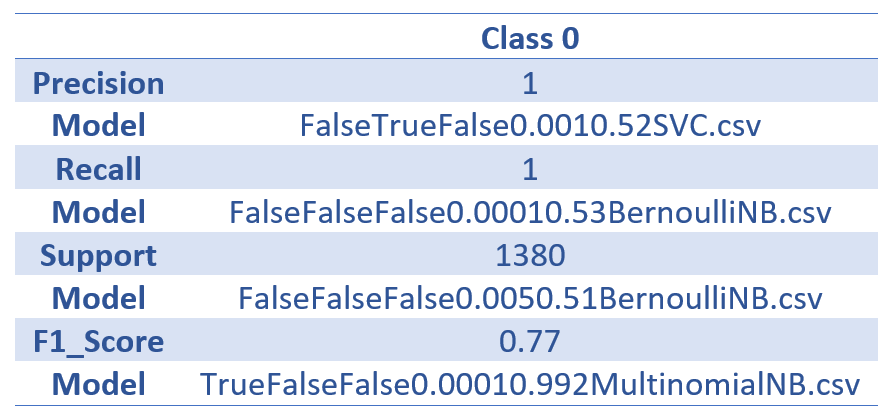
In the following table we show an example of the results of our tested preprocessing, SW means StopWords and ST means Stemming.

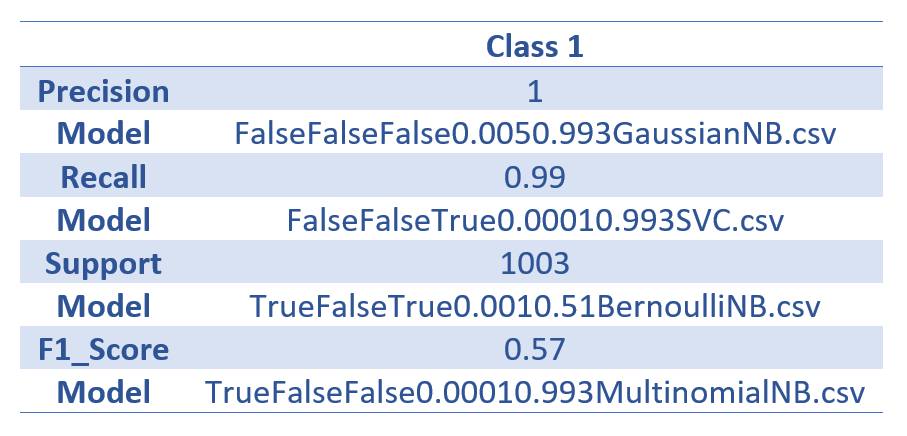


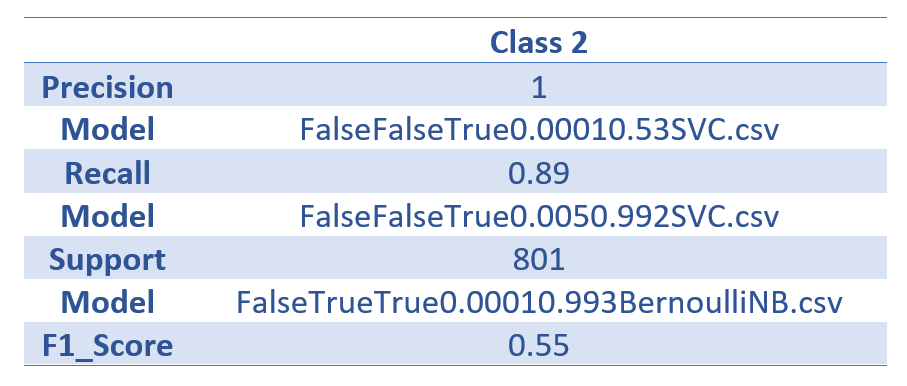
In the following table we show the number of parameters growing with number of ngrams chosen.

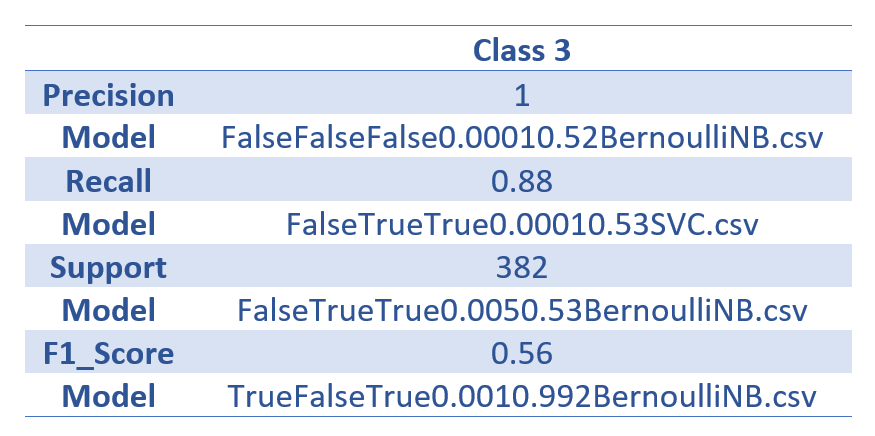


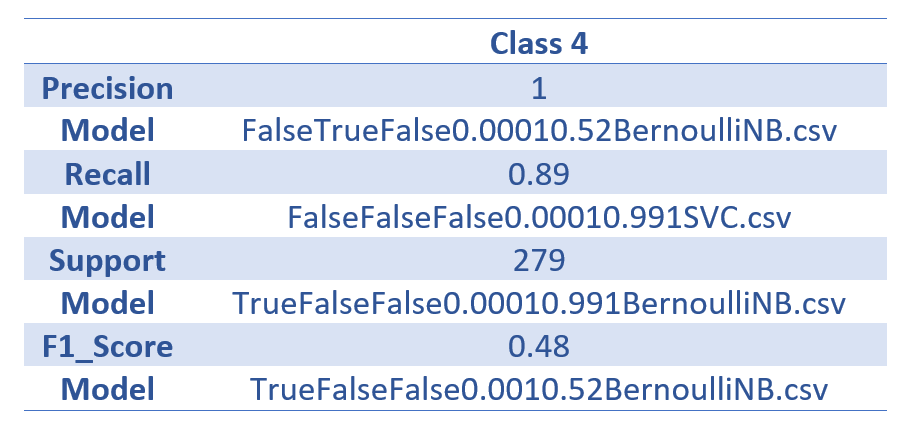
The following tables show some information we extracted from the results of our grid search (576 models tested), grouped by class. The name structure follows the following rule:

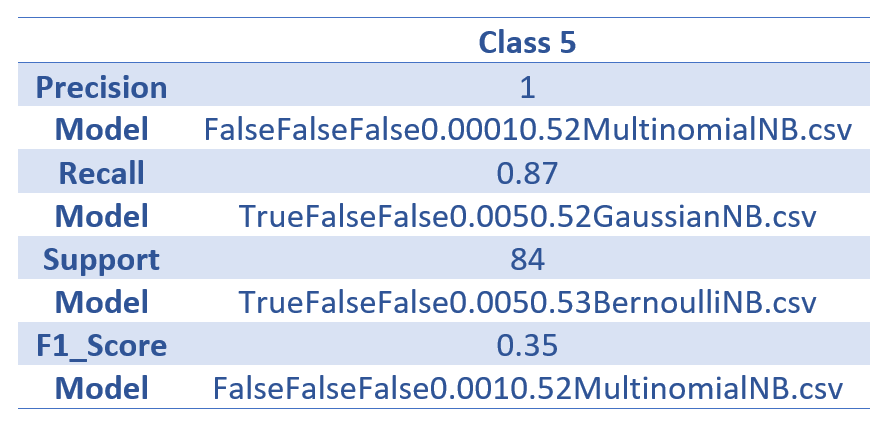
[StopWords:Steeming:tfIdf;minFreq;maxFreq:Ngrams:model.csv] 







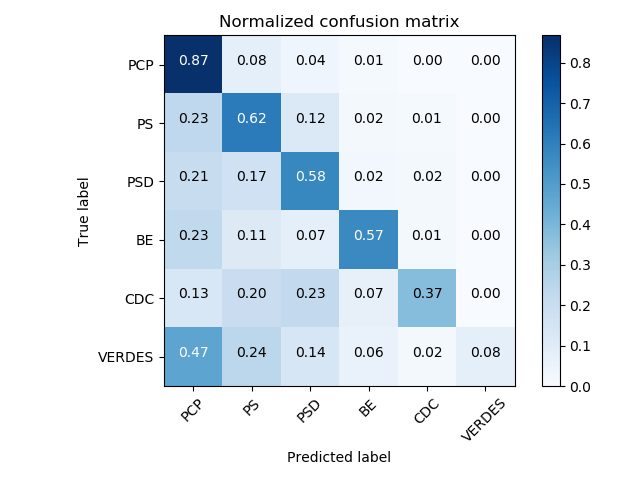




The architecture who achieved better overall results doesn’t use stop words or stemming, uses td-idf normalized with min frequency of 0.001 and max frequency of 0.5, with n-grams:1,2,3, and the model is a SVM using linear kernel.

The following results show its performance.





Extra Results: Testing with Neural Networks showed clear overfitting that we weren’t able to fix (always predicted the same class). Voting ensembles with both hard and soft voting achieved less than 0.5 of precision. More than 3 NGRAMS lead to worst results. Not defining a min and max frequency lead to worst results. Stop words and stemming didn’t improve the results.

The performance of SVM seems logic as we think that: text is often linear separable, and text has a lot of features. The Linear SVM is also a good choice because it is computationally cheap.

We understand that better results were possible it the dataset was more balanced, Verdes instances are highly infrequent, while PCP are highly frequent instances.

4.3 Named Entity Recognition: