Practice IV

Text classification

Specifications

- Individually do the following
  - Load the raw version of corpus generated in practice II
    - *Title* and *content* columns must be concatenated and will be used as features
    - *Section* column will be used as target (class)
  - Split the corpus in train an test sets using 80% for training and 20% for testing
  - Apply text normalization
  - Create different text representations of the corpus using unigrams
  - Use different machine learning methods to train a model and predict test instances
  - Evaluate predictions of models

Specifications

- Text normalization
  - For this processes you can use:
    - Tokenization
    - Text cleaning
    - Stop words
    - Lemmatization
  - You should try different step combinations or versions in order to improve the classifier performance

Specifications

- Text representation
  - For this processes you can use:
    - Binarized
    - Frequency
    - TF-IDF
    - Embeddings
  - You could try SVD to generate an alternative version of text representation

Specifications

- Machine learning methods
  - For this processes you can use any classifier that supports multi-class classification
  - It would help if you tuned the algorithm parameters to improve the results

Evidence

- Source code
- A report in PDF format describing the following:
  - Task to be solved
  - Text normalization process
  - Text representations
  - Machine learning methods

Evidence

A table describing the experiments performed showing the best configuration of each ML method

Machine learning method	ML method parameters	Text normalization	Text representation	Average f-score
Logistic regression	max_iter = 200	Tokenization + stopwords + lemmatization	binarized	0.85
Naïve Bayes	default	Tokenization + stopwords + lemmatization	frequency	0.88
...	...	...	...	...
Multilayer perceptron	hidden_layer_sizes = (200, 100)	Tokenization + text_cleaning + stopwords + lemmatization	Tf-idf + svd	0.9