University of Turin
Department of Computer Science

Natural Language Technologies Project

Text categorization by VSM model

Authors:
Christian Quaranta
Giulia Coucourde
Leonardo Magliolo

Academic Year 2022/2023

Introduction:

The project consists of the realization of software capable of classifying documents by supervised machine learning that exploits Rocchio's method with Vector Space Model representation formalism.
The classifier, when learning, will need to take into account the proximity of the documents with respect to the positive reference centroid but also the relative distance to the centroid of the negative examples with respect to the class of documents being considered.

Dataset:

The dataset on which the algorithm was tested consists of 200 documents written in the Italian language and partitioned into 10 different classes: 20 documents for each class.

Model:

Define the following sets:

- D: Set that encompasses all the documents that it is useful to consider.
- T: Set enclosing all terms mentioned at least within some document belonging to D
- C: Set that encloses all classes related to documents that it is useful to consider
- POS_i: Set that encloses all documents belonging to D that are classified with the i-th class.
- $NPOS_i$: Set that encloses the N most positive documents that are not classified with the i-th class.
- W: set of all possible weights w_{kj} associated with the k-th term belonging to T and the j-th document belonging to D (determined as the product between the frequency of the term in the document and its IDF)

For each class belonging to C a vector c_i is learned : $<f_{1i}, \dots, f_{|T|i}>$.

For component f_{ki} is determined by the following formula:

$$f_{ki} = \beta \cdot \sum_{d_j \in POS_i} \frac{w_{kj}}{|POS_i|} - \gamma \cdot \sum_{d_j \in NPOS_i} \frac{w_{kj}}{|NPOS_i|}$$

Learning:

During the learning phase, it was useful to employ the following techniques to extract the values of the vectors$< f_{1i}, \ldots, f_{|T|i} >$:

- Bag of words: Useful for representing a document by words contained, as the model considered does not need to examine sequences of words.
- Removal of stopwords: Removal of words that have poor selectivity relative to the classes of the documents in which they are cited.
- Lemmatization: Different words with common lemmas express similar concepts, considering them as different components within class vectors would lower the accuracy of the model.
- Inverse Document Frequency: At same frequency it allows more selective terms to be weighted in favor of a particular subset of classes.
- Parameters$\beta\gamma\beta$=16 and γ=4 (as per the design request) and N = 10 were set.

Decoding:

The class of a document is predicted on the basis of maximizing the similarity between the document vector itself (d) and the of the vectors associated with the possible classes:

If $\hat{c} = argmax_{c_i} \left(\frac{d \cdot c_i}{|d| \cdot |c_i|}\right)$ then the predicted class of document d will be the itself associated with the vector $\hat{c}$.

Testing:

It was chosen to test the performance of the model on the dataset using the cross-validation technique (as suggested in the project requirements).

Specifically, 10 different partitions of the dataset consisting of 90% of the instances (18 per class) for the training sets and the remaining 10% (2 per class) for the test set were made, respectively, and the confusion matrix and model accuracy were calculated for each of them.

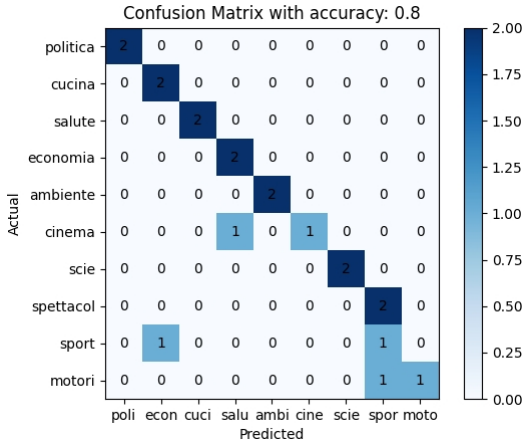
Libraries used:

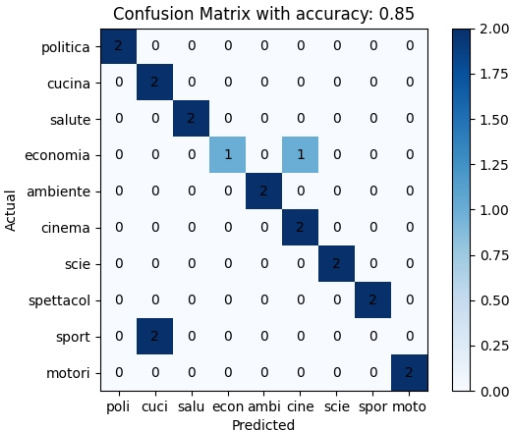
In addition to the native functionality provided by Python, the following libraries were chosen to support the implementation of the project:

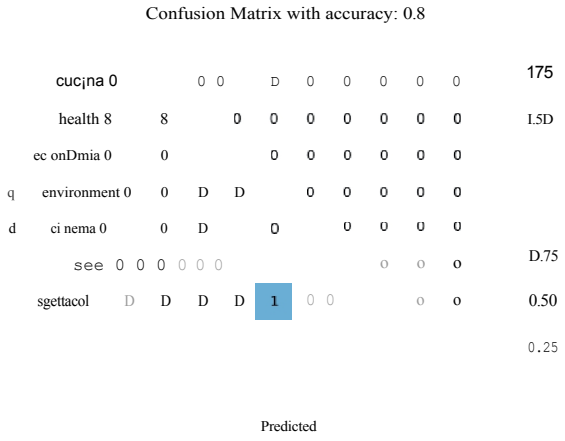
- Spacy: Used for removing stopwords and performing lemmatization operations. The template that is used for Spacy is : "it_core_news_lg".
- Scikit-learn: Used to partition the dataset, calculate cosine- similarity metrics, and represent confusion matrices.
- Pandas and Numpy: Used for representation of vectors and matrices useful for model implementation.

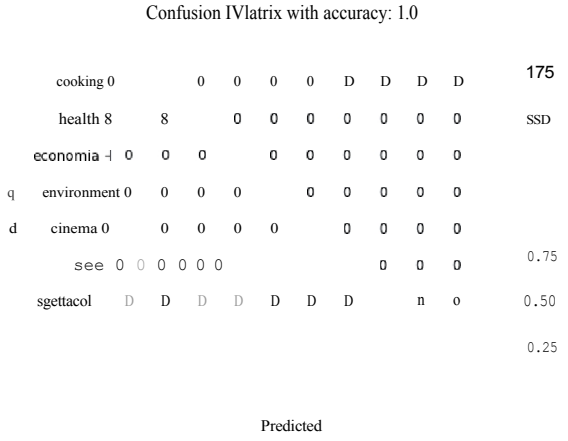
Achievements:

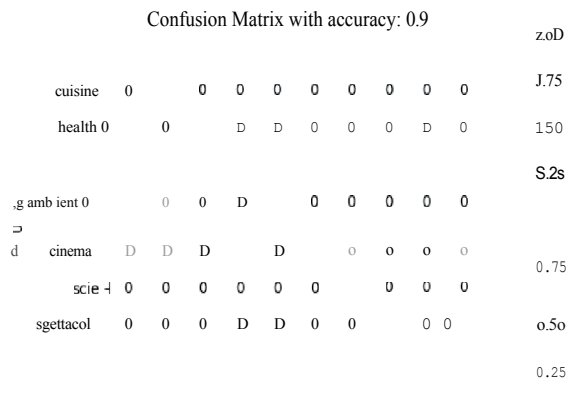
The confusion matrices associated with each of the 10 tests performed for cross-validation with their accuracy values are shown below:

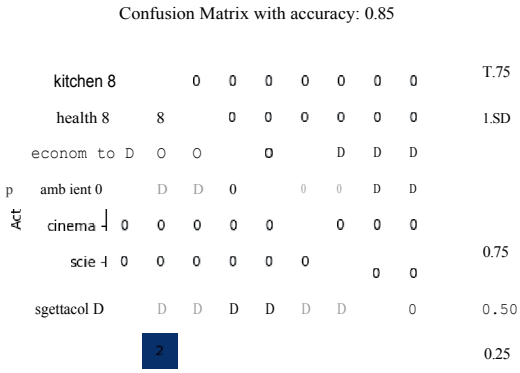


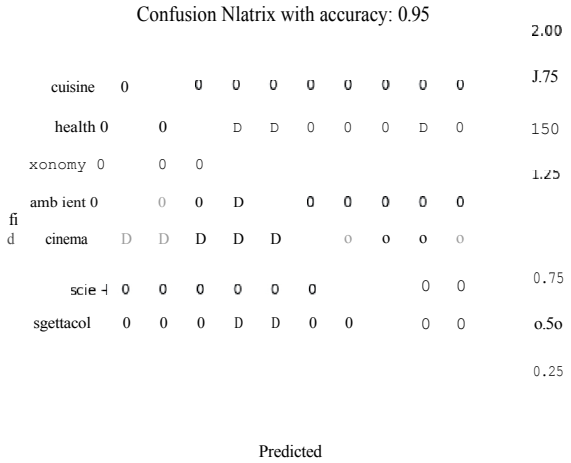


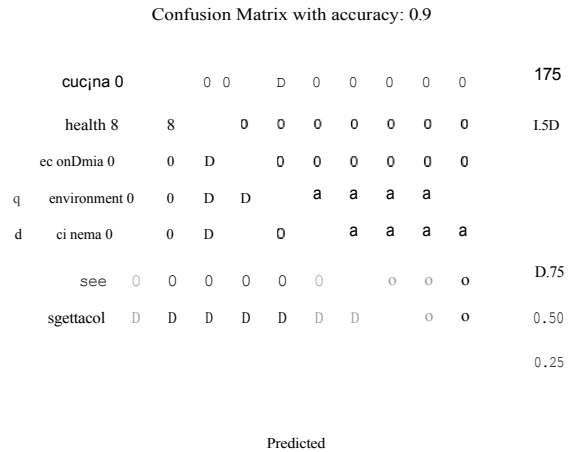

Confusion Matrix with accuracy: 0.8
cucina
health
ecnomia
environment
cinema
see
sgettacol
Predicted

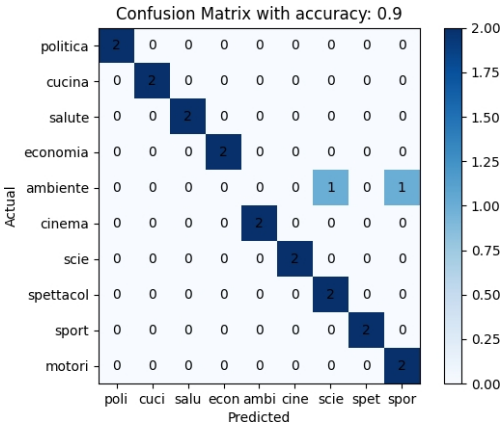

Confusion IVlatrix with accuracy: 1.0
cooking
health
economia
environment
cinema
see
sgettacol
Predicted

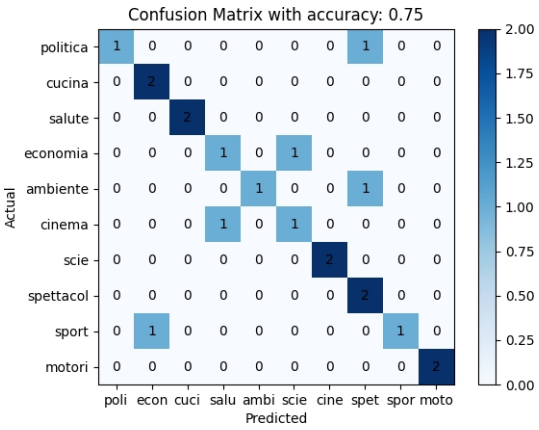

Confusion Matrix with accuracy: 0.9
cuisine
health
ambient
cinema
science
sgettacol
Predicted


Confusion Matrix with accuracy: 0.85
kitchen
health
economic
ambient
cinema
science
sgettacol
Act
poli cuci satu econ ambi ci ne scie sgor rroto
Predicted


Confusion Nlatrix with accuracy: 0.95
cuisine
health
economy
ambient
cinema
science
sgettacol
Predicted


Confusion Matrix with accuracy: 0.9
cucina
health
ecnomia
environment
cinema
see
sgettacol
Predicted





The average accuracy reported amounts to 0.87 and the variance of accuracy is

0.0057. The average number of errors per class, on the other hand, is quite

variable:

- Shows: 0.8
- Policy: 0.5
- Economy: 0.4
- Engines:0.4
- Environment: 0.2
- Health: 0.1
- Scientific: 0.1
- Sports: 0.1
- Kitchen: 0

The hypothesis is that categories lend themselves better to describing scenarios involving others:

- The shows often describe situations that concern areas outside the category of the category itself.
- Policy needs to regulate aspects of human life outside the category itself, so specific terms frequently appearing in documents belonging other classes may be used.
- Similar reasoning can be applied in economics.