

# Struttura del progetto

All'interno del repository sono presenti quattro entità principali:

- file `classifier.py`, contenente il classificatore, questo è il file da eseguire per verificare il funzionamento dell'assignment
- file `document_retriever.ipynb`, un notebook dentro il quale sono contenute le funzioni utilizzate per il download dei vari documenti tramite api di wikipedia
- directory `Corpora`, contenente il training set, che consiste di due cartelle, `Medical` e `Non Medical` ciascuna contenente i documenti necessari al training del classificatore
- directory `Test`, contenente il test set precedentemente estratto dal training set, le label assegnate a ciascun documento sono contenute nel file `test_labels.txt`, mentre i file relativi ai documenti su cui testare il classificatore sono nella directory `TestSet`

È inoltre presente un'ultima directory `CorporaBackup` che contiene tutti i documenti scaricati tramite utility `document_retriever.ipynb`, prima che questi vengano suddivisi in training e test set.

## `document_retriever.ipynb`

Come precedentemente accennato, questo file contiene le utility per la creazione di training e test set. In particolare in primo luogo viene creato il corpora iniziale, costituito di tutti i documenti (visibile nella directory `CorporaBackup`), in seguito viene diviso secondo la proporzione 80% training, 20% test.

Per i documenti medicali e non sono stati considerati documenti appartenenti a varie categorie:

```
medical_categories = [  
    "Category:Bacteriology",  
    "Category:Virology",  
    "Category:Cancer",  
    "Category:Anatomy",  
    "Category:Genetics",  
    "Category:Pediatrics",  
]  
  
non_medical_categories = [  
    "Category:Geometry",  
    "Category:Literature",  
    "Category:Hunting",  
    "Category:Politics",  
    "Category:Education",
```

```
"Category:Fashion",  
]
```

per ognuna di queste categorie è vengono ritornati al massimo 100 elementi attenenti alla particolare categoria. Si può osservare che in `CorporaBackup` i file in ogni directory prima che vengano divisi in training e test set non sono 600 e 600, come ci si aspetterebbe, questo perché per alcune categorie vengono trovati meno di 100 documenti.

Per dividere i documenti in training e test set viene utilizzata la funzione `populate_test_set()` del notebook `document_retriever.ipynb`.

## `classifier.py`

Si tratta di una classe che include tutte le utility principali del classificatore. Alla generazione del classificatore vengono svolti i seguenti passi:

- costruite le due *bag of words* relative alle due categorie **medical** e **non-medical**
- costruito il *vocabolario*
- compute le *priors* per ciascuna delle due classi

Al fine di costruire le due bag of words per ogni categoria, come anche spiegato all'interno del codice, viene generata la bag of words di ogni testo appartenente alla categoria data. A tale scopo, prima di raggiungere la sua rappresentazione come BoW, ogni documento viene normalizzato tramite i seguenti passi:

- **tokenization** tramite `nltk.wordpunct`, tramite la quale il testo viene diviso in vari token, rimuovendo inoltre i segni di punteggiatura
- **stopword elimination**, in modo tale da andare ad eliminare le parole più comuni del linguaggio
- per ogni token ne viene effettuato lo **stemming**, in modo da andare a ridurre la dimensione della BoW (riducendo ogni token al proprio stem, si più token diventano un solo stem)
- rimozione degli stem rimasti contenenti il simbolo `=`, questo viene fatto in quanto le sezioni dei documenti del Corpora sono tra loro separate tramite `== Nome Sezione ==`, non è rilevante dunque tenere conto di quel simbolo nella maggioranza dei casis

A questo punto la fase di training si può ritenere completa, e si può procedere alla vera e propria classificazione: `Classifier.classify()`.

In particolare la classificazione viene effettuata tramite il metodo **Naive Bayes** secondo la formula (si veda pdf per visualizzare la formula in modo corretto):

$$\hat{y} = \operatorname{argmax} \log(P(\text{class})) + \sum_{i \in \text{pos}} \log(p(w_i | \text{class}))$$

Dopo aver eseguito la classificazione, verranno mostrati i risultati, che nello specifico caso di un'istanza di training e test set sono i seguenti:

```
True medicals: 110
True non medicals: 88
False negatives: 7
False positives: 0

The average precision is: 0.9743172816343548
The average recall is: 0.9401709401709402
The accuracy is: 0.9658536585365853
```