Datamining : TP3

M2 Informatique, Universite Paris 7

Anne-Claire Haury

Exercice 1

1. Ecrire un script qui prend en entree le chier `SMSSpamCollection.txt`, lit les messages et leurs labels ('spam' ou 'ham') et les range dans une liste `messages` et une liste `labels`.
2. Creer une liste nommee `spams` contenant tous les spams et une liste nommee `hams` contenant tous les messages legitimes.
3. Ecrire une fonction `frequence_mots` qui prend en entree une liste de messages et renvoie un dictionnaire permettant, pour chaque mot, d'acceder au nombre de fois qu'il apparaît dans l'ensemble des messages.
4. Explorer ces donnees en regardant par exemple quels mots apparaissent le plus souvent et en dessinant un histogramme. Faire de même pour les spams d'une part et les messages legitimes de l'autre.

Exercice 2

Ecrire une fonction `nettoyage_texte` qui prend en entree une chaîne de caracteres et renvoie le texte nettoye de la ponctuation, des chi res, des majuscules, etc. Si vous avez le temps, nettoyez egalement les urls, les accents, les mots trop courts, etc.

Pour cela, vous aurez sans doute besoin de la librairie `re` de python (expressions regulieres).

Exercice 3

Ecrire une classe `TfIdf` qui traite un corpus de documents et calcule les valeurs tf-idf de chacun des mots. Un objet de cette classe doit être capable d'updater ces valeurs lorsqu'un nouveau document est lu.

Exercice 4

En utilisant la fonction `scipy.stats.ttest_ind` qui e ectue un test de di erence de deux moyennes comme vu en cours (cours 2), dire si la frequence d'apparition du mot 'love' est signi cativement di erente entre les spams et les non-spams. Idem pour le mot 'call'. Faites ce test pour chacun des mots et faites apparaître les mots dont la frequence d'apparition est signi cativement di erente entre les deux types de messages.