Projet–Fouille de Données
Thème : Classification des Tweets

Objectifs :

- Maitriser l'API de twitter pour l'extraction des tweets
- Maitriser la partie NLP (natural language processing) avec NLTK en Python
- Appliquer les principes de nettoyage des données
- Classer les tweets : regrouper ensemble les tweets qui sont similaires. C'est une étape qui peut être considérée comme une étape

Spécifications

Imaginons que vous avez un compte Twitter, et que vous lez suivre les tweets (texte très court) sur ce réseau social. Vu le nombre colossal de Tweets, et faute de temps, vous n'avez pas la possibilité de les lire tous. Pour cela, vous avez besoin d'une application qui va jouer le rôle d'assistantet qui va vous effectuer un résumé de toutes ces informations. Une des approches qu'on peut utiliser estde le classer sous former de groupes de sorte à ce qu'on présente à l'utilisateur un seul Tweet de chaque groupe.
Pour cela, on doit procéder en trois grandes étapes :

i. Prétraitement des tweets

Dans cette étape, l'objectif est d'éliminer le texte inutile des tweets tels que les #, les noms des utilisateurs, les url, …

ii. Traitement des tweets : NLP (Natural LanguageProcessing)

On doit procéder à l'analyse du tweet en respectant les différentes étapes du NLP (*Natural LanguageProcessing*). La bibliothèque à utiliser est NLTK en Python.

iii. Classification des tweets

Etant donné un ensemble de tweets, l'objectif est de les résumer sous formes de groupes de sorte à ce que les Tweets qui sont dans le même groupe soient similaires. Ainsi, l'utilisateur pourra par la suite lire juste un Tweet de chaque groupe (le Tweet qui est le centroïde de groupes).

Travail à faire

1. Télécharger les Tweets à partir de Twitter en utilisant l'API de twitter. Pour cela, vous devriez un compte « Twitter Developer ». Pour cela, vous devriez télécharger au moins 10 mille twwets. Pour la documentation de l'API de twitter, vous pouvez consulter les liens suivants :
   - https://developer.twitter.com/en/docs/twitter-api
   - https://developer.twitter.com/en
2. Utiliser la bibliothèque NLTK pour effectuer une analyse de chaque tweet et le transformer en un ensemble de mots en suivant les différentes étapes de base du processus NLP (Natural

LanguageProcessing). Pour la documentation sur NLTK et les différentes phases du NLP, vous pouvez consulter l'un des liens suivants :

- https://pythonspot.com/category/nltk/
- https://www.youtube.com/watch?v=WYge0KZBhe0
- https://www.tutorialspoint.com/natural_language_processing/natural_language_processing_python.htm

3. Utiliser l'algorithme *K-Means* pour classer les Tweets en *k* classes (vous pouvez essayer plusieurs valeurs de k allant de 3 à 30 par exemple).
   Notez bien que pour calculer la distance entre deux textes (Tweets transformés sous forme d'un ensemble de mots), on peut utiliser la distance de Jaccard définie comme suite :

$$dist((tweet1, tweet2) = \frac{|tweet1 \cap tweet2|}{|tweet1 \cup tweet2|}$$

4. Après avoir récupéré les classes, choisissez un Tweet par classe comme représentant. Les Tweets choisis seront les résumés de toutes les informations contenues dans les Tweets.

Livrables

Lien Binder du fichier jupyter notebook contenant le travail pour cela :

- Installer la bibliothèque twitter ,(n'oublier pas de mettre à jour pip) vous pouvez utiliser aussi la bibliothèque tweepy :
- Consulter la documentation https://python-twitter.readthedocs.io/en/latest/ ou http://docs.tweepy.org/en/latest/
- Utiliser la bibliothèque NLTK : https://nltk.readthedocs.io/en/latest/
- Utiliser Sklearn :https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html
- Générer le fichier requirements.txt indispensable pour la création du binder.
- Vous pouvez ajouter un lien vers une vidéo de démonstration (Youtube) dans votre jupyter notebook exemple :

```
# import IPython.display.YouTubeVideo class.
from IPython.display import YouTubeVideo
# create an instance of YouTubeVideo class with provided youtube video id.
youtube_video = YouTubeVideo('nC3QfNiudQ4')
# display youtube video
display(youtube_video)
```

Remarque : La qualité du rapport ainsi que la bonne organisation du fichier jupyter notebook seront pris en compte dans l'évaluation.

Référence utiles: https://github.com/PacktPublishing/Learning-Data-Mining-with-Python