project Data Mining

Réalisé par : Nidhal Hazbri 3DNI2

Objectifs :

```
• Maitriser l’API de twitter pour l’extraction des tweets
• Maitriser la partie NLP (natural language processing) avec NLTK en Pyt
hon
• Appliquer les principes de nettoyage des données
• Classer les tweets : regrouper ensemble les tweets qui sont similaire
s. C’est une étape qui peutêtre considérée comme une étape
```

Specifications

Imaginons que vous avez un compte Twitter, et que vous lez suivre les tweets sur ce reseau social. Vu le nombre colossal de Tweets, et faute de temps, vous n'avez pas la possibilite de les lire tous. Pour cela, vous avez besoin d'une application qui va jouer le role d'assistantet qui va vous effectuer un resume de toutes ces informations. Une des approches qu'on peut utiliser estde le classer sous former de groupes de sorte a ce qu'on presente a l'utilisateur un seul Tweet de chaque groupe. Pour cela, on doit proceder en trois grandes etapes :

Travail faire

On a Telecharger les tweets a partir de Twitter en utilisant l'API de twitter. Pour cela, vous devriez un compte « Twitter Developper ». Pour cela, vous devriez telecharger au moins 10 mille twwets. Pour la documentation de l'API de twitter, vous pouvez consulter les liens suivants :

In [1]:
```python
import pandas as pd
import tweepy
consumer_key="LHZVzcEN30hfmN2cPBqkoB3wq"
consumer_secret="DGZ7gQFDlqXoPfmAUWHOsY2eMTA0qhgKVb3rbExcx8Vhav3x3a"
access_token="1325046107437752325-a2zNm36NnzJqTFBFkIagjzpkdCadjs"
access_token_secret="7ohQJ7WTf2DuHsr9NNwPkOPXq5zUkaycrzo2nPhPUoGLL"
auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_token_secret)
api = tweepy.API(auth)
```

In [2]:
```
twitter_data_analysis = pd.DataFrame(columns = ['text'])
tweets = tweepy.Cursor(api. user_timeline , id="twitter").items( 15000)

# Iterate and print tweets
i=0
for tweet in tweets:
    twitter_data_analysis.loc[i,"text"] = tweet.text
    i+=1
```

In [3]:
```
import csv
twitter_data_analysis.to_csv('twitter_data_analysis.csv',index = False)
twitter_data_analysis.head(10)
```

Out[3]:

	text
0	RT @shesooosaddity: if you had a twitter befor...
1	@CloudNaii 40404
2	@issahairplug drink water replaced good morning
3	@Ne_ThatGuy we're taking oomf to the Fleets
4	@_JusJust_ remember "I dedicate my 500th Tweet...
5	@ambr_ncole they're tourists
6	@PhallonXOXO proof you're doing it right 😌
7	some of you hating...\n\nbut we see you Fleeti...
8	That thing you didn’t Tweet but wanted to but ...
9	@quakerraina this is art

In [4]:
```
twitter_data_analysis.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 3218 entries, 0 to 3217
Data columns (total 1 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   text    3218 non-null   object
dtypes: object(1)
memory usage: 210.3+ KB
```

Pretraitement des tweets

Dans cette etape, l'objectif est d'eliminer le texte inutile des tweets tels que les #, les noms des utilisateurs, les url,emoji ...

In [5]:
```
import re
for index, row in twitter_data_analysis.iterrows():
    err = row['text']
    new0 = re.sub(r"http\S+", "", err)
    new1 = re.sub(r"#\S+", "", new0)
    new2 = re.sub(r"@\S+", "", new1)
    new3 = re.sub(r"\n+", "", new2)
    new4 = re.sub(r"RT+", "", new3)
    new5 = re.sub("\d+", '', new4)
    emoji_pattern = re.compile("["
                               u"\U0001F600-\U0001F64F"  # emoticons
                               u"\U0001F300-\U0001F5FF"  # symbols & pictographs
                               u"\U0001F680-\U0001F6FF"  # transport & map symbol
                               u"\U0001F1E0-\U0001F1FF"  # flags (iOS)
                               u"\U00002500-\U00002BEF"  # chinese char
                               u"\U00002702-\U000027B0"
                               u"\U00002702-\U000027B0"
                               u"\U000024C2-\U0001F251"
                               u"\U0001f926-\U0001f937"
                               u"\U00010000-\U0010ffff"
                               u"\u2640-\u2642"
                               u"\u2600-\u2B55"
                               u"\u200d"
                               u"\u23cf"
                               u"\u23e9"
                               u"\u231a"
                               u"\ufe0f"  # dingbats
                               u"\u3030"
                               "]+", flags=re.UNICODE)
    new6 = re.sub(emoji_pattern, "", new5)
    twitter_data_analysis.loc[index,'text'] =new6
```

In [6]:
```
twitter_data_analysis.head(40)
```

Out[6]:

	text
0	if you had a twitter before rt this
1	
2	drink water replaced good morning
3	we're taking oomf to the Fleets
4	remember "I dedicate my th Tweet to:____"
5	they're tourists
6	proof you're doing it right
7	some of you hating...but we see you Fleeting
8	That thing you didn’t Tweet but wanted to but ...
9	this is art
10	aren't we all six feet
11	this Tweet just graduated with honors
12	saw it, love it, can't wait for the wedding p...
13	
14	breathe
15	apology accepted
16	H
17	THIRSTY
18	looking hydrated
19	the moon will share
20	bark among the stars
21	rubber ducky knew all along
22	If the moon can hydrate so can you
23	Reading an article before Retweeting it? That’...
24	Hey everyone, we made a temporary change to th...
25	Me seeing my Twitter friends I’ve never met ...
26	
27	dedication
28	not a single person on this app
29	but was it a good Tweet?
30	checks out
31	you forgot one:
32	mutual acknowledgment of good Tweets is frien...
33	

	text
34	how it started it never ended we get it
35	just make sure your DMs are open
36	that was a classic
37	no don't stop
38	strangers to bffs on Twitter real quick
39	cool cool cool cool cool

In [7]:
```
twitter_data_analysis.to_csv('clining_twitter_data_analysis.csv',index = False)
```

In [8]:
```
import nltk
nltk.download('stopwords' )
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\nidhal\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

Out[8]:
```
True
```

Traitement des tweets: NLP (Natural LanguageProcessing)

On doit proceder a !'analyse du tweet en respectant les differentes etapes du NLP (Natural LanguageProcessing). La bibliotheque a utiliser est NLTK en Python.

In [9]:
```
from nltk.stem.porter import PorterStemmer
from nltk.corpus import stopwords
ps = PorterStemmer()
stemed_dataset=[]
for i in range(0,twitter_data_analysis.shape[0]):
    stemmed_array=twitter_data_analysis['text'][i].split()
    stemmed=[ps.stem(word) for word in stemmed_array if not word in set(stopwords
    stemmed=' '.join(stemmed)
    stemed_dataset.append(stemmed)
print(stemed_dataset[0:10])
```

```
['twitter rt', '', 'drink water replac good morn', "we'r take oomf fleet", 'rem
emb "I dedic th tweet to:____"', "they'r tourist", 'proof right', 'hating...but
see fleet', 'that thing didn't tweet want didn't got close like nah. We place f
or…', 'art']
```

In [10]:
```python
from sklearn.feature_extraction.text import CountVectorizer
cv = CountVectorizer()
X=cv.fit_transform(stemed_dataset)
print(X)
```

```
  (0, 2505)	1
  (0, 1977)	1
  (2, 683)	1
  (2, 2609)	1
  (2, 1915)	1
  (2, 985)	1
  (2, 1517)	1
  (3, 2619)	1
  (3, 2312)	1
  (3, 1638)	1
  (3, 875)	1
  (4, 1908)	1
  (4, 589)	1
  (4, 2352)	1
  (4, 2495)	1
  (4, 2413)	1
  (4, 0)	1
  (5, 2368)	1
  (5, 2442)	1
  (6, 1820)	1
  (6, 1947)	1
  (7, 875)	1
  (7, 1061)	1
  (7, 321)	1
  (7, 2035)	1
  :	:
  (3212, 1362)	1
  (3212, 159)	1
  (3212, 1626)	1
  (3212, 453)	1
  (3213, 286)	1
  (3213, 2374)	1
  (3213, 1263)	1
  (3213, 118)	1
  (3214, 2619)	1
  (3214, 896)	1
  (3214, 1362)	1
  (3214, 2507)	1
  (3214, 59)	1
  (3214, 2657)	1
  (3214, 452)	1
  (3215, 992)	1
  (3215, 1097)	1
  (3215, 821)	1
  (3216, 366)	1
  (3217, 985)	2
  (3217, 1517)	1
  (3217, 2495)	1
  (3217, 1577)	1
  (3217, 80)	1
  (3217, 1026)	1
```

Classification des tweets

Etant donne un ensemble de tweets, l'objectif est de les resumer sous formes de groupes de sorte a ce que les Tweets qui sont dans le meme groupe soient similaires. Ainsi, l'utilisateur pourra par la suite lire juste un Tweet de chaque groupe (le Tweet qui est le centro"ide de groupes). on a Utiliser l'algorithme K-Means pour classer les Tweets en k classes ,valeurs de k allant de 1 a30 par exemple).

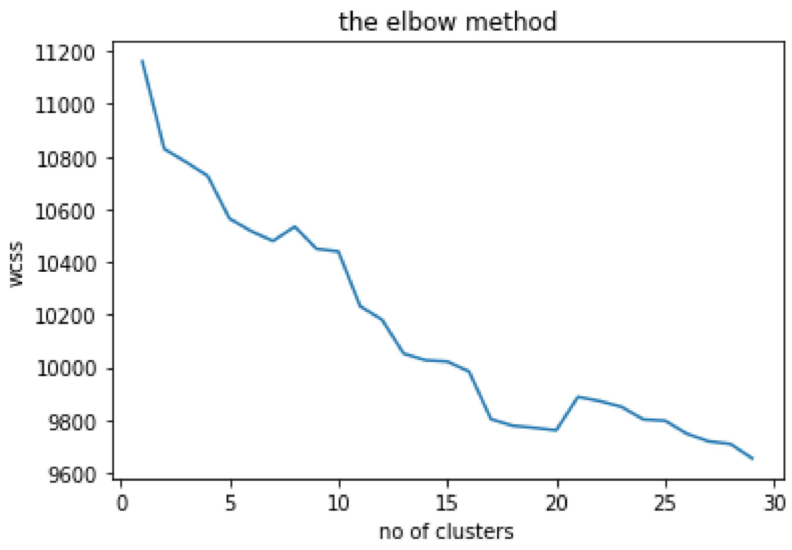
In [11]:
```
from sklearn.cluster import KMeans
wcss=[]
```

In [12]:
```
for i in range(1,30):
    Kmeans=KMeans(n_clusters=i,init='k-means++',max_iter=300,n_init=10,random_sta
    Kmeans.fit(X)
    wcss.append(Kmeans.inertia_)
```

```
Iteration  0, inertia 20498.000
Iteration  1, inertia 11076.509
Iteration  2, inertia 11063.863
Iteration  3, inertia 10972.384
Iteration  4, inertia 10807.074
Iteration  5, inertia 10796.986
Converged at iteration 5: center shift 0.000000e+00 within tolerance 1.263624
e-07
Initialization complete
Iteration  0, inertia 11271.000
Iteration  1, inertia 11145.480
Iteration  2, inertia 11105.142
Iteration  3, inertia 11037.026
Iteration  4, inertia 10979.415
Converged at iteration 4: center shift 0.000000e+00 within tolerance 1.263624
e-07
Initialization complete
Iteration  0, inertia 11257.000
Iteration  1, inertia 10937.616
```

In [13]:
```python

import matplotlib.pyplot as plt
plt.plot(range(1,30),wcss)
plt.title('the elbow method')
plt.xlabel('no of clusters')
plt.ylabel('wcss')
plt.show()
```



In [14]:
```python
true_k=30
Kmeans=KMeans(n_clusters=true_k,init='k-means++',n_init=1)
Kmeans.fit(X)
```

Out[14]:
```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
       n_clusters=30, n_init=1, n_jobs=None, precompute_distances='auto',
       random_state=None, tol=0.0001, verbose=0)
```

La cellule suivante contient les mots cles de chaque cluster

In [15]:

```
print("Top terms per cluster:")
order_centroids = Kmeans.cluster_centers_.argsort()[:, ::-1]
terms = cv.get_feature_names()
for i in range(true_k):
    print("Cluster %d:" % i)
    for ind in order_centroids[i, :10]:
        print(' %s' % terms[ind])
    print()
print("\n")
```

```
 freeeezi
 freedom

Cluster 27:
 we
 tIny
 four
 fren
 freestylin
 freeeezi
 freedom
 free
 freak
 frank

Cluster 28:
 pa
 sco
 tu
```

On a choisir un Tweet par classe comme representant. Les tweets choisis seront les resumes de toutes les informations contenues dans les tweets.

In [16]:
```
result_final_twitter_data_analysis = pd.DataFrame(columns = [ 'text'])
i=0
j=0
while i<30:
    while True:
        Y=cv.transform([stemed_dataset[j]])
        prediction=Kmeans.predict(Y)
        if i == prediction:
            print("tweet of cluster "+str(prediction)+twitter_data_analysis.loc[i
            result_final_twitter_data_analysis.loc[i,"text"] =twitter_data_analys
            j=0
            break
        j+=1
    i+=1
result_final_twitter_data_analysis.to_csv('result_final_twitter_data_analysis.csv
```

```
tweet of cluster [0]  if you had a twitter before  rt this
tweet of cluster [1]
tweet of cluster [2] drink water replaced good morning
tweet of cluster [3] we're taking oomf to the Fleets
tweet of cluster [4] remember "I dedicate my th Tweet to:____"
tweet of cluster [5] they're tourists
tweet of cluster [6] proof you're doing it right
tweet of cluster [7]some of you hating...but we see you Fleeting
tweet of cluster [8]That thing you didn’t Tweet but wanted to but didn’t but go
t so close but then were like nah. We have a place for…
tweet of cluster [9] this is art
tweet of cluster [10] aren't we all six feet
tweet of cluster [11] this Tweet just graduated with honors
tweet of cluster [12] saw it, love it, can't wait for the wedding pics
tweet of cluster [13]
tweet of cluster [14]breathe
tweet of cluster [15] apology accepted
tweet of cluster [16] H
tweet of cluster [17] THIRSTY
tweet of cluster [18] looking hydrated
tweet of cluster [19] the moon will share
tweet of cluster [20] bark among the stars
tweet of cluster [21] rubber ducky knew all along
tweet of cluster [22]If the moon can hydrate so can you
tweet of cluster [23]Reading an article before Retweeting it? That’s growth.Bef
ore you Retweet an article, we’ll remind you to read it…
tweet of cluster [24]Hey everyone, we made a temporary change to the Retweet fu
nction. When you hit the Retweet button, you can either…
tweet of cluster [25]  Me seeing my Twitter friends I’ve never met in person su
cceed.
tweet of cluster [26]
tweet of cluster [27] dedication
tweet of cluster [28] not a single person on this app
tweet of cluster [29] but was it a good Tweet?
```

conclusion : ¶

on a charge les tweets d'apres l'api de twitter, on les a mis dans le fichier csv twitter_data_analysis. puis on a fait le data cleaning et on a mis le resultat dans le fichier

cleaning_twitter_data_analysis.csv.Et enfin on a mis un tweet de chaque cluster dans le fichier result_final_twitter_data_analysis.

lien ghithub (https://github.com/hazbri/projectDataMining/)