MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITÉ DE SOUSSE

المعهد العالي للإعلامية وتقنيات الاتصال بحمام سوسة

Institut supérieur de l'informatique et des technologies de la communication - HAMMAM SOUSSE

Rapport de projet Data Mining

Classification des Tweets

Réalisé par :

Amina Ladhari 3DNI2

Encadrée par :

Mr. Lotfi Ben Romdhane

Mr. Khemais Abdallah

Introduction

Le *Data Mining* est en fait un terme générique englobant toute une famille d'outils facilitant l'exploration et l'analyse des données contenues au sein d'une base décisionnelle de type Data Warehouse ou DataMart. Les techniques mises en action lors de l'utilisation de cet instrument d'analyse et de prospection sont particulièrement efficaces pour extraire des informations significatives depuis de grandes quantités de données.

Processus de data Mining :

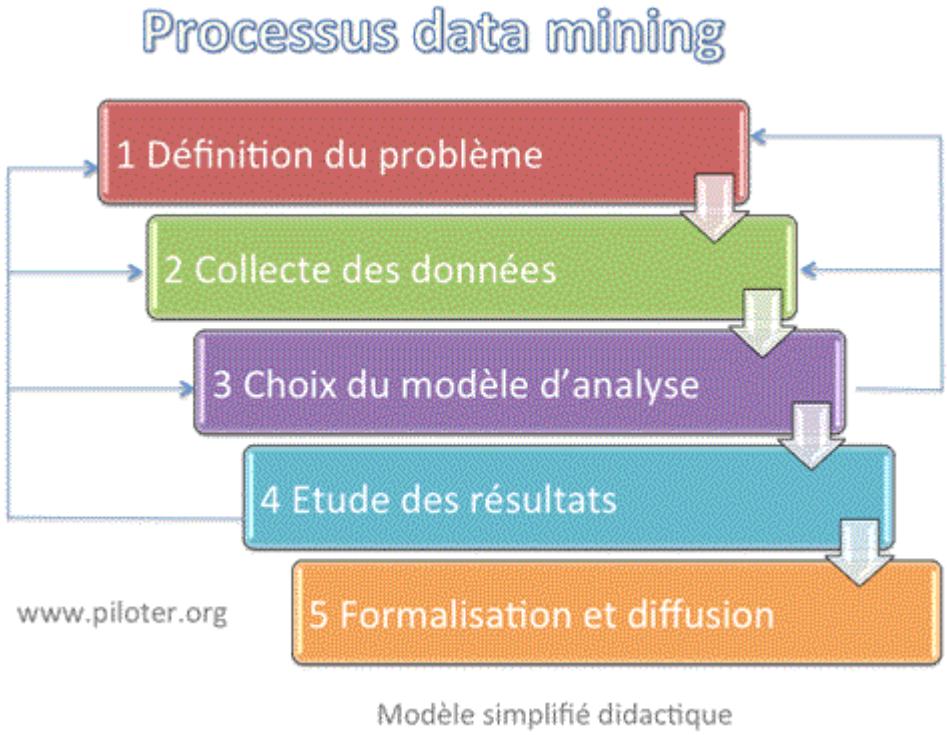

Figure 1 : Processus de data mining

Préparation datasets

On a utilisé ces deux bibliothèques pour collecter les tweets

- Tweepy

```
In [1]: !pip install tweepy
```

```
Collecting tweepy
  Downloading tweepy-3.9.0-py2.py3-none-any.whl (30 kB)
Requirement already satisfied: six>=1.10.0 in /opt/conda/lib/python3.7/site-packages (from tweepy) (1.14.0)
Requirement already satisfied: requests-oauthlib>=0.7.0 in /opt/conda/lib/python3.7/site-packages (from tweepy)
(1.2.0)
Requirement already satisfied: requests[socks]>=2.11.1 in /opt/conda/lib/python3.7/site-packages (from tweepy) (2.
23.0)
Requirement already satisfied: requests>=2.0.0 in /opt/conda/lib/python3.7/site-packages (from requests-oauthlib>=
0.7.0->tweepy) (2.23.0)
```

Figure 2 : Installation de Tweepy

- GetOldTweets3

```
In [2]: !pip install GetOldTweets3
```

```
Collecting GetOldTweets3
  Downloading GetOldTweets3-0.0.11-py3-none-any.whl (13 kB)
Requirement already satisfied: lxml>=3.5.0 in /opt/conda/lib/python3.7/site-packages (from GetOldTweets3) (4.5.0)
Collecting pyquery>=1.2.10
  Downloading pyquery-1.4.3-py3-none-any.whl (22 kB)
Requirement already satisfied: lxml>=3.5.0 in /opt/conda/lib/python3.7/site-packages (from GetOldTweets3) (4.5.0)
Collecting cssselect>0.7.9
  Downloading cssselect-1.1.0-py2.py3-none-any.whl (16 kB)
Installing collected packages: cssselect, pyquery, GetOldTweets3
Successfully installed GetOldTweets3-0.0.11 cssselect-1.1.0 pyquery-1.4.3
```

Figure 3 : Installation de GetOldTweets3

Récupérer les tweets

On constate les 3000 tweets les plus récents qui étaient pertinents pour le thème <Education>.

```
# Input search query to scrape tweets and name csv file
# Max recent tweets pulls x amount of most recent tweets from that user
text_query = ' Education '
count = 3000
#screen_name = screen_name
# Calling function to query X amount of relevant tweets and create a CSV file
text_query_to_csv(text_query, count)
```

Figure *4* : Récupérer les tweets par text_query

Save file csv

```
Education= pd.read_csv('./ Education -tweets.csv' )
Education.head(3000)
```

Figure 5 : enregistrer le fichier en csv

Concaténation de datasets

On fait la concaténation après la concaténation des tweets

```
import os
import glob
import pandas as pd
datasets = pd.concat([education, mechanical,health, sport],ignore_index=True)

datasets.head(12000)
```

	user	Text
0	Education1939	This past week The Staff Recognition Committee...
1	alexanderrusso	RT @UNICEFmedia: "Evidence shows that schools ...
2	TaleamSystems	Taleam Systems' CEO implemented the Student e-...
3	javrsda	Ya quiero que salga la nueva temporada de Sex ...
4	Calpe19	RT @LizzyJPrice: @JonnyGeller A great loss. Wh...
...	...	...
11995	Aaliyah_ys2	RT @donachena: Mon hygiène de vie est horrible...
11996	ALEXANDREDECAS6	RT @anderbatist: COM O VASCO NO Z-4, SPORT x C...
11997	witho68	@wisey_9 No less a sport than gymnastics, sync...
11998	sportscardex	@SGCFinests @darrenrovell @Sothebys @GoldinAuc...

Figure 6 : Concaténation de datasets

Import libraires python

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

import tweepy
import csv
import os
import pandas as pd

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import re
import spacy
from sklearn.model_selection import train_test_split
import nltk
nltk.download('stopwords')
nltk.download('wordnet')
from nltk.tokenize import RegexpTokenizer, WhitespaceTokenizer
from nltk.stem import WordNetLemmatizer
from nltk.corpus import stopwords
import string
from string import punctuation
import collections
from collections import Counter
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
import en_core_web_sm
```

Figure 7 : Importation des libraires python

Distance de Jaccard

```
def jaccard_similarity(query, document):
    intersection = set(query).intersection(set(document))
    union = set(query).union(set(document))
    return len(intersection)/len(union)
# jaccard_score(socialvector, economic_vector)

#for similarity of 1 and 2 of column1
# jaccard_similarity('dog lion a dog','dog is cat')


def get_scores(group,tweets):
    scores = []
    for tweet in tweets:
        s = jaccard_similarity(group, tweet)
        scores.append(s)
    return scores
```

Figure 8: calculer la Distance de Jaccard entre les tweets

On fait le calcul pour tous les catégories.

KMeans Clustering

Pour traiter les données d'apprentissage, l'algorithme K-means dans l'exploration de données.

```
# fitting kmeans to dataset
kmeans = KMeans(n_clusters=3, init='k-means++', n_init=10, max_iter=300, random_state=0)
Y_kmeans = kmeans.fit_predict(X)

# Visualising the clusters
plt.scatter(X[Y_kmeans==0, 0], X[Y_kmeans==0, 1], s=100, c='violet', label= 'Cluster 1')
plt.scatter(X[Y_kmeans==1, 0], X[Y_kmeans==1, 1], s=100, c='cyan', label= 'Cluster 2')
plt.scatter(X[Y_kmeans==2, 0], X[Y_kmeans==2, 1], s=100, c='green', label= 'Cluster 3')
#plt.scatter(X[Y_kmeans==3, 0], X[Y_kmeans==3, 1], s=100, c='blue', label= 'Cluster 4')
#plt.scatter(X[Y_kmeans==4, 0], X[Y_kmeans==4, 1], s=100, c='magenta', label= 'Cluster 5')
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], s=100, c='black', label='Centroids' )
plt.title('Clusters of tweets in mechanical and sport groups')
plt.xlabel('mechanical tweets')
plt.ylabel('sport tweets')
plt.legend()
plt.show()
```

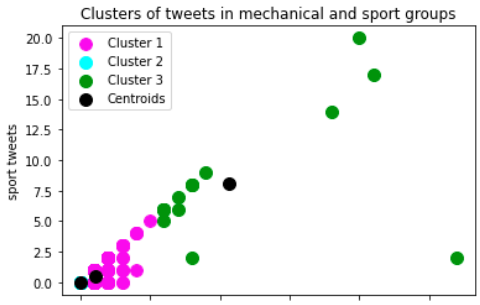

Figure 9 : Appliquer l'algorithme de clustering

Clustered Datasets :

Ce graphe à secteurs pour afficher le nombre total de tweets dans chaque catégorie.

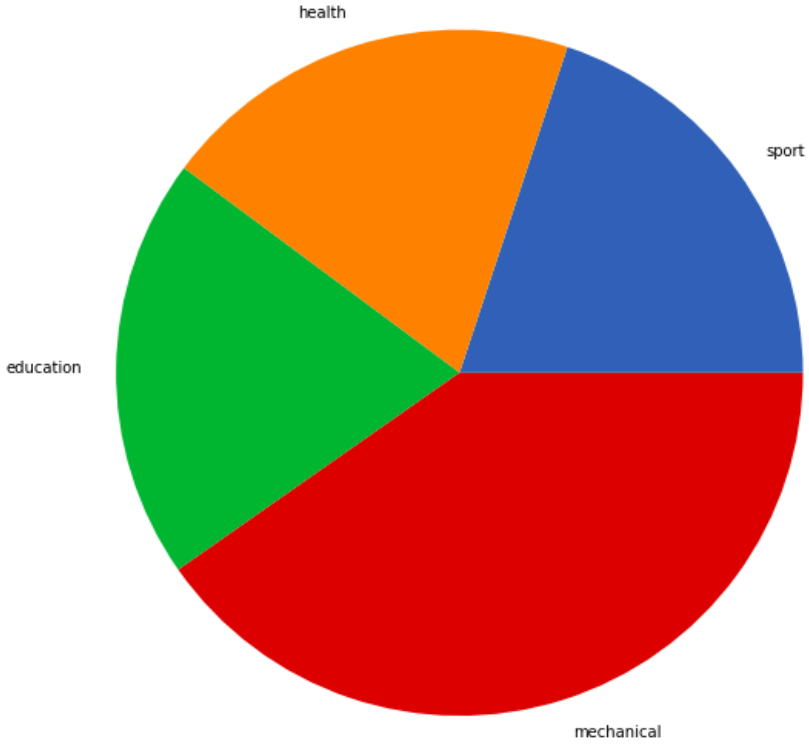

Figure 10: Représentation du volumes des tweets

Les tweets représentatifs de datasets total

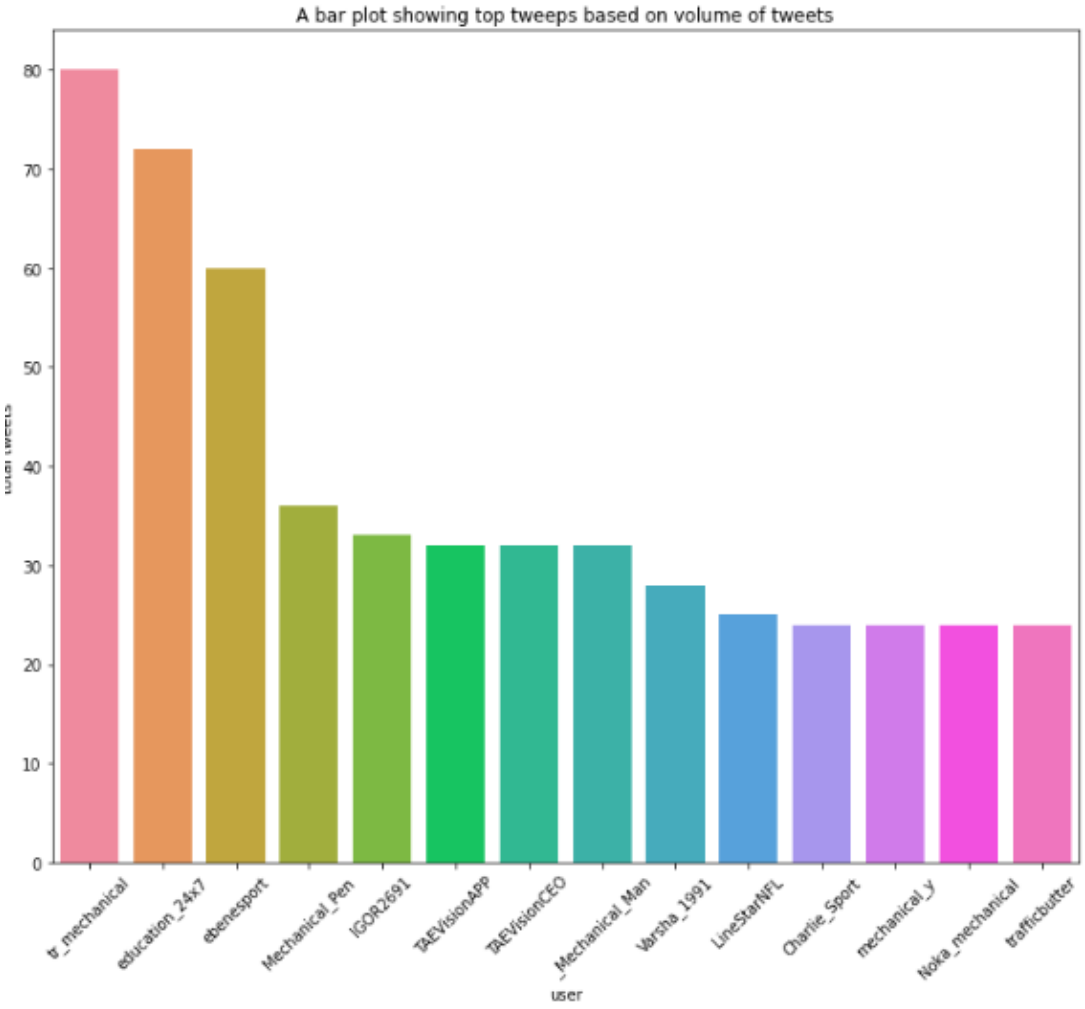

Figure 11 : Les tweets représentatifs de catégorie technology