FOUILLE DE DONNEES

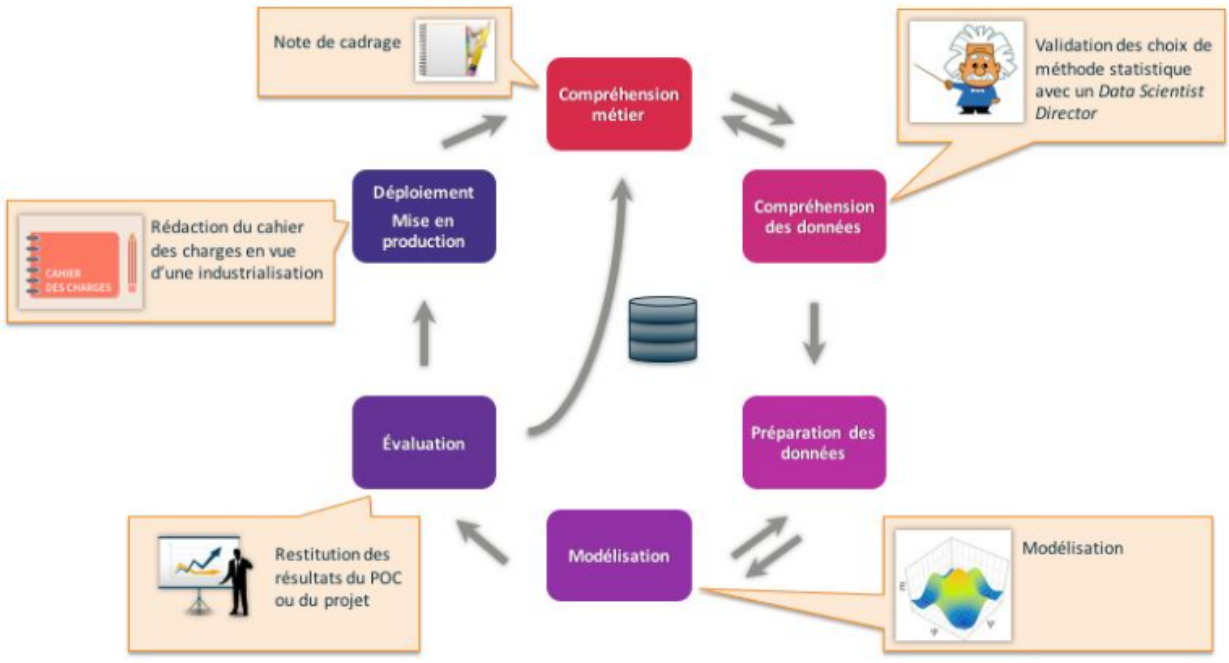
CLASSIFICATION DES TWEETS

Définition de la fouille de données

La fouille de données est un domaine qui est apparu avec l'explosion des quantités d'informations stockées, avec le progrès important des vitesses de traitement et des supports de stockage. La fouille de données vise à découvrir, dans les grandes quantités de données,les informations précieuses qui peuvent aider à comprendre les données ou à prédire le comportement des données futures. Le datamining utilise depuis sont apparition plusieurs outils de statistiques et d'intelligence artificielle pour atteindre ses objectifs.

Processus du data mining

Il est très important de comprendre que le data mining n'est pas seulement le problème de découverte de modèles dans un ensemble de donnée. Ce n'est qu'une seule étape dans tout un processus suivi par les scientifiques, les ingénieurs ou toute autre personne qui cherche à extraire les connaissances à partir des données. En 1996 un groupe d'analystes définit le data mining comme étant un processus composé de cinq étapes sous le standard **CRISP-DM (Cross-Industry Standard Process for Data Mining)** comme schématisé cidessous :



1. La compréhension du problème métier

La première étape consiste à bien comprendre les éléments métiers et problématiques que la Data Science vise à résoudre ou à améliorer.

L'objectif de notre projet consiste à

• 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.

1.1 Collecte des données

Dans cette étape, on s'intéresse à la manière dont les données sont générées et collectées. D'après la définition du problème et des objectifs du data mining, on peut avoir une idée sur les données qui doivent être utilisées. Ces données n'ont pas toujours le même format et la même structure.

Les données utilisées sont extraites de Twitter à l'aide de Tweepy, une bibliothèque python permettant d'accéder à l'API Twitter.

In [70]:

```
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
import tweepy as tw
import pandas as pd
import numpy as np
import nltk
import string
import re
import string

# better viewing of tweet text
pd.set_option('display.max_colwidth', 150)
# reproducible rng
seed = 42
plt.style.use("bmh")
#%matplotlib inline

consumer_key = "Tippg04Ns8bfIZtE3uMprKEym"
consumer_secret = "qHjBkD4CBX67rHPenKu96c7W5Tsv59t9BGBVsm0pXM0oLatOS8"
access_token= "1328070400136400905-ZOCtds4L0DD3LqMJ2r9rZHSyAi75Xu"
access_token_secret = "qX3pBBD42A1dAZDPXx2DGcH0T0hcr4e5Oxw40YwR9wTnc"

auth = tw.OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_token_secret)
api = tw.API(auth,wait_on_rate_limit=True,wait_on_rate_limit_notify=True, compressi
```

```
"""search_words ='#'
date = "2020-12-10" # choose any date with the format "yyyy-mm-dd"
```

```
posts =[]
tweets = tw.Cursor(api.search,q=search_words,lang="en",since=date).items(10000)
for tweet in tweets:

#print(tweet.id,tweet.user.screen_name,tweet.created_at,tweet.text,len(tweet.text)
)
    posts.append([tweet.id,tweet.user.screen_name,tweet.created_at,tweet.text])
    df_tweets = pd.DataFrame(posts).to_csv('data.csv',index=False,header=
['ID','User','Date','Tweet'])
"""
"""search_words ='#Corona'
date = "2019-05-05" # choose any date with the format "yyyy-mm-dd"
posts =[]
tweets = tw.Cursor(api.search,q=search_words,lang="en",since=date).items(2000)
for tweet in tweets:

#print(tweet.id,tweet.user.screen_name,tweet.created_at,tweet.text,len(tweet.text)
)

posts.append([tweet.id,tweet.user.screen_name,tweet.created_at,tweet.text,len(twee
t.text)])
    df_tweets = pd.DataFrame(posts).to_csv('dataa.csv',index=False,header=
['ID','User','Date','Tweet','Len'])
    """
"""search_words ='#élection'
date = "2018-04-15" # choose any date with the format "yyyy-mm-dd"
posts =[]
tweets = tw.Cursor(api.search,q=search_words,lang="en",since=date).items(1000)
for tweet in tweets:

#print(tweet.id,tweet.user.screen_name,tweet.created_at,tweet.text,len(tweet.text)
)

posts.append([tweet.id,tweet.user.screen_name,tweet.created_at,tweet.text,len(twee
t.text)])
    df_tweets = pd.DataFrame(posts).to_csv('dataaa.csv',index=False,header=
['ID','User','Date','Tweet','Len'])"""
```

In [71]:

```
dataSet_tweets1= pd.read_csv('data.csv')
dataSet_tweets2= pd.read_csv('dataa.csv')
dataSet_tweets3= pd.read_csv('dataaa.csv')
dataSet_tweets1.columns, dataSet_tweets2.columns, dataSet_tweets3.columns = ['ID','
dataSet_tweets = pd.concat([dataSet_tweets1, dataSet_tweets2, dataSet_tweets3], ign
```

Après avoir collecté les données a partir de l'API twitter on a concatiné les 3 fichiers CSV pour avoir un nombre de tweet égale à 10000

2. La compréhension des données

Cette phase vise à déterminer précisément les données à analyser, à identifier la qualité des données disponibles et à faire le lien entre les données et leur signification d'un point de vue métier. La Data Science étant basée sur les données seules, les problèmes métiers relatifs à des données existantes, qu'elles soient internes ou externes, peuvent ainsi être résolus par la Data Science.

In [72]:

```
style_dict = {'background-color':'#31C7FA',
              'color':'black',
              'border-color': 'white',
              'border-width':'2px',
              'font-family':'Roboto'}
# Affichage des 4 premières lignes
dataSet_tweets.head().style.set_properties(**style_dict)
```

Out[72]:

	ID	User	Date	Tweet	Len
0	1336858921605197824	AlanHerringtonn	2020-12-10 02:22:59	I wish every band would find success :(	39
1	1336858921592639488	tina29073	2020-12-10 02:22:59	RT @Franklin_Graham: Concerns with the voting machines, software, mail-in ballots, counting procedures, and more seem to be growing every d…	140
2	1336858921563250688	JPHUPE	2020-12-10 02:22:59	That's more like it https://t.co/vMgy7wuOnl	43
3	1336858921563205634	fellawhomstdve	2020-12-10 02:22:59	three hours into ambien and tweeting and she gives you this look, wyd https://t.co/YwzAbalKM4	93
4	1336858921529651206	jjgisjjg	2020-12-10 02:22:59	RT @BBCWorld: New Christmas campaign for Canadians held in China https://t.co/ke7qlriuGc	88

In [73]:

```
print("DataFrame's shape : ", dataSet_tweets.shape)
```

```
DataFrame's shape :  (10965, 5)
```

In [74]:

```
# affichage d'informations sur les données
dataSet_tweets.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10965 entries, 0 to 10964
Data columns (total 5 columns):
ID       10965 non-null int64
User     10965 non-null object
Date     10965 non-null object
Tweet    10965 non-null object
Len      10965 non-null int64
dtypes: int64(2), object(3)
memory usage: 428.4+ KB
```

In [75]:

```
missing = dataSet_tweets.isna().sum()
print(missing)
```

```
ID       0
User     0
Date     0
Tweet    0
Len      0
dtype: int64
```

==> Notre DataSet de 10965 échantillons et de 5 attributs
==> Nous avons 2 attributs numériques et 3 du type catégorique
==> Nous n'avons pas des valeur manquantes

---

3. Préparation des données

Les données collectées doivent être "préparées". Avant tout, ellesd oivent être nettoyées puisqu'elles peuvent contenir plusieurs types d'anomalies : des données peuvent être omises à cause des erreurs de frappe ou à causes des erreursdues au système lui-même, dans ce cas il faut remplacer ces données ou éliminer complètement leurs enregistrements. Des données peuvent être incohérentes c-à-d qui sortent des intervalles permis, on doit les écarter où les normaliser. Parfois on est obligé à faire des transformations sur les données pour unifier leur poids.

In [76]:

```
#Supprimer les lignes dupliquées
dataSet_tweets.drop_duplicates('Tweet')
dataSet_tweets.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10965 entries, 0 to 10964
Data columns (total 5 columns):
ID       10965 non-null int64
User     10965 non-null object
Date     10965 non-null object
Tweet    10965 non-null object
Len      10965 non-null int64
dtypes: int64(2), object(3)
memory usage: 428.4+ KB
```

==> Dans notre projet on est besoin de La colonne Tweets seulement pour faire l'analyse

In [77]:

```
dataSet_tweets=dataSet_tweets.drop(columns = ['ID','User','Len'])
```

Nettoyage des textes

faut supprimer les ponctuations, les liens, hashtags, RT, les émojis, les caractères spéciaux, les chiffres

In [78]:

```
string.punctuation
```

Out[78]:

```
'!"#$%&\'()*+,-./:;<=>?@[\\]^_`{|}~'
```

In [79]:

```
def cleanText(text):
    text  = "".join([char for char in text if char not in string.punctuation])
    text = text.lower() # Makes text lowercase
    text = re.sub("http\S+", "", text)
    text = re.sub('[0-9]+', '', text)
    text = re.sub(r'https?:\/\/\S+', '', text) # Removes hyperlinks
    text = re.sub('#', '', text) # Removes hashtags
    text = re.sub('@[A-Za-z0–9]+', '', text) # Removes mentions (@)
    text = re.sub('RT[\s]+', '', text) # Removes "RT"
    text = re.sub(r"@\S+", '', text)
    text = re.sub(r"\n+", '', text)
    text = re.sub("RT+", '', text)
    text = re.sub("rt[\s]+", '', text)
    text = re.sub("hhh+", '', text)

    emoji_pattern = re.compile("["
                               u"\U0001F600-\U0001F64F"  # emoticons
                               u"\U0001F300-\U0001F5FF"  # symbols & pictographs
                               u"\U0001F680-\U0001F6FF"  # transport & map symbols
                               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)
    text = re.sub(emoji_pattern, '', text)

    return text
```

In [80]:

```
dataSet_tweets['Tweet_punct'] = dataSet_tweets['Tweet'].apply(lambda x: cleanText(x
```

In [81]:

```
style_dict2 = {'background-color':'#4ACDF9',
               'color':'black',
               'border-color': 'white',
               'border-width':'2px',
               'font-family':'Roboto'}

dataSet_tweets.head(7).style.set_properties(**style_dict2)
```

Out[81]:

	Date	Tweet	Tweet_punct
0	2020-12-10 02:22:59	I wish every band would find success :(	i wish every band would find success
1	2020-12-10 02:22:59	RT @Franklin_Graham: Concerns with the voting machines, software, mail-in ballots, counting procedures, and more seem to be growing every d…	franklingraham concerns with the voting machines software mailin ballots counting procedures and more seem to be growing every d…
2	2020-12-10 02:22:59	That’s more like it https://t.co/vMgy7wuOnl	that’s more like it
3	2020-12-10 02:22:59	three hours into ambien and tweeting and she gives you this look, wyd https://t.co/YwzAbalKM4	three hours into ambien and tweeting and she gives you this look wyd
4	2020-12-10 02:22:59	RT @BBCWorld: New Christmas campaign for Canadians held in China https://t.co/ke7qlriuGc	bbcworld new christmas campaign for canadians held in china

NLP:Natural LanguageProcessing

Le NLP pour Natural Language Processing ou Traitement Numérique du Langage est une discipline qui porte essentiellement sur la compréhension, la manipulation et la génération du langage naturel par les machines. Ainsi, le NLP est réellement à l’interface entre la science informatique et la linguistique. Il porte donc sur la capacité de la machine à interagir directement avec l’humain.

les différentes étapes du NLP

1.Tokenisation

Passons à la Tokénization ! C’est un procédé très simple qui divise une chaîne de caractère en tokens, c’est-à-dire des éléments atomiques de la chaîne. Un token n’est pas forcément un mot, ce peut être par exemple un signe de ponctuation. NLTK fournit plusieurs types de tokénization, comme la tokénization par mot ou par

phrase.

==>On doit procéder à l'analyse du tweet en respectant les différentes étapes du NLP

In [82]:

```
from nltk.tokenize import word_tokenize

dataSet_tweets['Tweet_tokenized'] = dataSet_tweets['Tweet_punct'].apply(lambda x: w

style_dict3 = {'background-color':'#6BD6FA',
               'color':'black',
               'border-color': 'white',
               'border-width':'2px',
               'font-family':'Roboto'}

dataSet_tweets.head(7).style.set_properties(**style_dict3)
```

Out[82]:

	Date	Tweet	Tweet_punct	Tweet_tokenized
0	2020-12-10 02:22:59	I wish every band would find success :(	i wish every band would find success	['i', 'wish', 'every', 'band', 'would', 'find', 'success']
1	2020-12-10 02:22:59	RT @Franklin_Graham: Concerns with the voting machines, software, mail-in ballots, counting procedures, and more seem to be growing every d…	franklingraham concerns with the voting machines software mailin ballots counting procedures and more seem to be growing every d…	['franklingraham', 'concerns', 'with', 'the', 'voting', 'machines', 'software', 'mailin', 'ballots', 'counting', 'procedures', 'and', 'more', 'seem', 'to', 'be', 'growing', 'every', 'd…']
2	2020-12-10 02:22:59	That's more like it https://t.co/vMgy7wuOnl	that's more like it	['that', "'", 's', 'more', 'like', 'it']
3	2020-12-10 02:22:59	three hours into ambien and tweeting and she gives you this look, wyd https://t.co/YwzAbalKM4	three hours into ambien and tweeting and she gives you this look wyd	['three', 'hours', 'into', 'ambien', 'and', 'tweeting', 'and', 'she', 'gives', 'you', 'this', 'look', 'wyd']
4	2020-12-10 02:22:59	RT @BBCWorld: New Christmas campaign for Canadians held in China https://t.co/ke7qlriuGc	bbcworld new christmas campaign for canadians held in china	['bbcworld', 'new', 'christmas', 'campaign', 'for', 'canadians', 'held', 'in', 'china']
5	2020-12-10 02:22:59	RT @susie_dent: Word of the day is 'ultracrepidarian' (19th century): a presumptuous critic; one who gives opinions and advice on subjects…	susiedent word of the day is 'ultracrepidarian' th century a presumptuous critic one who gives opinions and advice on subjects…	['susiedent', 'word', 'of', 'the', 'day', 'is', "'", 'ultracrepidarian', "'", 'th', 'century', 'a', 'presumptuous', 'critic', 'one', 'who', 'gives', 'opinions', 'and', 'advice', 'on', 'subjects…']
6	2020-12-10 02:22:59	@Roland_sanchez2 @Feet1100 um really is I can advertise if I want to. fuck off	rolandsanchez feet um really isi can advertise if i want to fuck off	['rolandsanchez', 'feet', 'um', 'really', 'isi', 'can', 'advertise', 'if', 'i', 'want', 'to', 'fuck', 'off']

2.Remove stopwords

Vient ensuite l'étape de suppression des stopwords qui est cruciale, car elle va enlever dans le texte tous les mots qui n'ont que peu d'intérêt sémantique. Les stopwords sont en effet tous les mots les plus courants d'une langue (déterminants, pronoms, etc..). NLTK dispose d'une liste de stopwords en anglais (ou dans d'autres

langues).

In [83]:

```
stopword = nltk.corpus.stopwords.words('english')

def remove_stopwords(text):
    text = [word for word in text if word not in stopword]
    return text
```

In [84]:

```
dataSet_tweets['Tweet_nonstop'] = dataSet_tweets['Tweet_tokenized'].apply(lambda x:
style_dict3 = {'background-color':'#6BD6FA',
               'color':'black',
               'border-color': 'white',
               'border-width':'2px',
               'font-family':'Roboto'}

dataSet_tweets.head(7).style.set_properties(**style_dict3)
```

Out[84]:

	Date	Tweet	Tweet_punct	Tweet_tokenized	Tweet_nonstop
0	2020-12-10 02:22:59	I wish every band would find success :(	i wish every band would find success	['i', 'wish', 'every', 'band', 'would', 'find', 'success']	['wish', 'every', 'band', 'would', 'find', 'success']
1	2020-12-10 02:22:59	RT @Franklin_Graham: Concerns with the voting machines, software, mail-in ballots, counting procedures, and more seem to be growing every d…	franklingraham concerns with the voting machines software mailin ballots counting procedures and more seem to be growing every d…	['franklingraham', 'concerns', 'with', 'the', 'voting', 'machines', 'software', 'mailin', 'ballots', 'counting', 'procedures', 'and', 'more', 'seem', 'to', 'be', 'growing', 'every', 'd…']	['franklingraham', 'concerns', 'voting', 'machines', 'software', 'mailin', 'ballots', 'counting', 'procedures', 'seem', 'growing', 'every', 'd…']
2	2020-12-10 02:22:59	That’s more like it https://t.co/vMgy7wuOnl	that’s more like it	['that', '’', 's', 'more', 'like', 'it']	['’', 'like']
3	2020-12-10 02:22:59	three hours into ambien and tweeting and she gives you this look, wyd https://t.co/YwzAbalKM4	three hours into ambien and tweeting and she gives you this look wyd	['three', 'hours', 'into', 'ambien', 'and', 'tweeting', 'and', 'she', 'gives', 'you', 'this', 'look', 'wyd']	['three', 'hours', 'ambien', 'tweeting', 'gives', 'look', 'wyd']
4	2020-12-10 02:22:59	RT @BBCWorld: New Christmas campaign for Canadians held in China https://t.co/ke7qlriuGc	bbcworld new christmas campaign for canadians held in china	['bbcworld', 'new', 'christmas', 'campaign', 'for', 'canadians', 'held', 'in', 'china']	['bbcworld', 'new', 'christmas', 'campaign', 'canadians', 'held', 'china']
5	2020-12-10 02:22:59	RT @susie_dent: Word of the day is ‘ultracrepidarian’ (19th century): a presumptuous critic; one who gives opinions and advice on subjects…	susiedent word of the day is ‘ultracrepidarian’ th century a presumptuous critic one who gives opinions and advice on subjects…	['susiedent', 'word', 'of', 'the', 'day', 'is', '’', 'ultracrepidarian', '’', 'th', 'century', 'a', 'presumptuous', 'critic', 'one', 'who', 'gives', 'opinions', 'and', 'advice', 'on', 'subjects…']	['susiedent', 'word', 'day', '’', 'ultracrepidarian', '’', 'th', 'century', 'presumptuous', 'critic', 'one', 'gives', 'opinions', 'advice', 'subjects…']
6	2020-12-10 02:22:59	@Roland_sanchez2 @Feet1100 um really is I can advertise if I want to. fuck off	rolandsanchez feet um really isi can advertise if i want to fuck off	['rolandsanchez', 'feet', 'um', 'really', 'isi', 'can', 'advertise', 'if', 'i', 'want', 'to', 'fuck', 'off']	['rolandsanchez', 'feet', 'um', 'really', 'isi', 'advertise', 'want', 'fuck']

3.Stemming and Lemmitization

Ces deux méthodes sont très couramment utilisées dans le traitement du langage naturel car permettent de représenter sous un même mot plusieurs dérivées du mot. Dans le cas du Stemming, nous allons uniquement garder le radical du mot (ex : dormir, dortoir et dors deviendront dor). La lemmatization, moins radicale , va laisser au mot un sens sémantique mais va éliminer le genre ou le pluriel par exemple.

In [85]:

```
ps = nltk.PorterStemmer()

def stemming(text):
    text = [ps.stem(word) for word in text]
    return text
```

In [86]:

```
dataSet_tweets['Tweet_stemmed'] = dataSet_tweets['Tweet_nonstop'].apply(lambda x: s

style_dict5 = {'background-color':'#A9E5F9',
               'color':'black',
               'border-color': 'white',
               'border-width':'2px',
               'font-family':'Roboto'}
#dataSet_tweets.head(7)
dataSet_tweets.head(7).style.set_properties(**style_dict5)
```

Out[86]:

	Date	Tweet	Tweet_punct	Tweet_tokenized	Tweet_nonstop	Tweet_ste
0	2020-12-10 02:22:59	I wish every band would find success :(	i wish every band would find success	['i', 'wish', 'every', 'band', 'would', 'find', 'success']	['wish', 'every', 'band', 'would', 'find', 'success']	['wish', 'band', ' 'find', 'su
1	2020-12-10 02:22:59	RT @Franklin_Graham: Concerns with the voting machines, software, mail-in ballots, counting procedures, and more seem to be growing every d…	franklingraham concerns with the voting machines software mailin ballots counting procedures and more seem to be growing every d…	['franklingraham', 'concerns', 'with', 'the', 'voting', 'machines', 'software', 'mailin', 'ballots', 'counting', 'procedures', 'and', 'more', 'seem', 'to', 'be', 'growing', 'every', 'd…']	['franklingraham', 'concerns', 'voting', 'machines', 'software', 'mailin', 'ballots', 'counting', 'procedures', 'seem', 'growing', 'every', 'd…']	['franklingr 'concern' 'm 'softwar', ' 'ballot', ' 'pro 'seem', 'everi
2	2020-12-10 02:22:59	That’s more like it https://t.co/vMgy7wuOnl	that’s more like it	['that', '’', 's', 'more', 'like', 'it']	['’', 'like']	['
3	2020-12-10 02:22:59	three hours into ambien and tweeting and she gives you this look, wyd https://t.co/YwzAbalKM4	three hours into ambien and tweeting and she gives you this look wyd	['three', 'hours', 'into', 'ambien', 'and', 'tweeting', 'and', 'she', 'gives', 'you', 'this', 'look', 'wyd']	['three', 'hours', 'ambien', 'tweeting', 'gives', 'look', 'wyd']	['three', 'ambien', 'give'
4	2020-12-10 02:22:59	RT @BBCWorld: New Christmas campaign for Canadians held in China https://t.co/ke7qlriuGc	bbcworld new christmas campaign for canadians held in china	['bbcworld', 'new', 'christmas', 'campaign', 'for', 'canadians', 'held', 'in', 'china']	['bbcworld', 'new', 'christmas', 'campaign', 'canadians', 'held', 'china']	['bbcworld' 'chi 'cam 'canadian'
5	2020-12-10 02:22:59	RT @susie_dent: Word of the day is ‘ultracrepidarian’ (19th century): a presumptuous critic; one who gives opinions and advice on subjects…	susiedent word of the day is ‘ultracrepidarian’ th century a presumptuous critic one who gives opinions and advice on subjects…	['susiedent', 'word', 'of', 'the', 'day', 'is', '’', 'ultracrepidarian', '’', 'th', 'century', 'a', 'presumptuous', 'critic', 'one', 'who', 'gives', 'opinions', 'and', 'advice', 'on', 'subjects…']	['susiedent', 'word', 'day', '’', 'ultracrepidarian', '’', 'th', 'century', 'presumptuous', 'critic', 'one', 'gives', 'opinions', 'advice', 'subjects…']	['susied', ' 'ultracrepi '’', 'th', 'c 'presu 'critic 'give', 'o 'subje
6	2020-12-10 02:22:59	@Roland_sanchez2 @Feet1100 um really is I can advertise if I want to. fuck off	rolandsanchez feet um really isi can advertise if i want to fuck off	['rolandsanchez', 'feet', 'um', 'really', 'isi', 'can', 'advertise', 'if', 'i', 'want', 'to', 'fuck', 'off']	['rolandsanchez', 'feet', 'um', 'really', 'isi', 'advertise', 'want', 'fuck']	['rolandsa 'feet', 'um', 'isi', 'ad 'want'

In [87]:

```
wn = nltk.WordNetLemmatizer()

def lemmatizer(text):
    text = [wn.lemmatize(word) for word in text]
    return text
```

In [88]:

```
dataSet_tweets['Tweet_lemmatized'] = dataSet_tweets['Tweet_nonstop'].apply(lambda x

style_dict6 = {'background-color':'#D4ECF4',
               'color':'black',
               'border-color': 'white',
               'border-width':'2px',
               'font-family':'Roboto'}

dataSet_tweets.head(7).style.set_properties(**style_dict6)
```

Out[88]:

	Date	Tweet	Tweet_punct	Tweet_tokenized	Tweet_nonstop	Tweet_ste
0	2020-12-10 02:22:59	I wish every band would find success :(	i wish every band would find success	['i', 'wish', 'every', 'band', 'would', 'find', 'success']	['wish', 'every', 'band', 'would', 'find', 'success']	['wish', 'band', ' 'find', 'su
1	2020-12-10 02:22:59	RT @Franklin_Graham: Concerns with the voting machines, software, mail-in ballots, counting procedures, and more seem to be growing every d…	franklingraham concerns with the voting machines software mailin ballots counting procedures and more seem to be growing every d…	['franklingraham', 'concerns', 'with', 'the', 'voting', 'machines', 'software', 'mailin', 'ballots', 'counting', 'procedures', 'and', 'more', 'seem', 'to', 'be', 'growing', 'every', 'd…']	['franklingraham', 'concerns', 'voting', 'machines', 'software', 'mailin', 'ballots', 'counting', 'procedures', 'seem', 'growing', 'every', 'd…']	['franklingr 'concern' 'm 'softwar', ' 'ballot', ' 'pro 'seem', 'everi
2	2020-12-10 02:22:59	That’s more like it https://t.co/vMgy7wuOnl	that’s more like it	['that', "'", 's', 'more', 'like', 'it']	["'", 'like']	["'
3	2020-12-10 02:22:59	three hours into ambien and tweeting and she gives you this look, wyd https://t.co/YwzAbalKM4	three hours into ambien and tweeting and she gives you this look wyd	['three', 'hours', 'into', 'ambien', 'and', 'tweeting', 'and', 'she', 'gives', 'you', 'this', 'look', 'wyd']	['three', 'hours', 'ambien', 'tweeting', 'gives', 'look', 'wyd']	['three', 'ambien', 'give'
4	2020-12-10 02:22:59	RT @BBCWorld: New Christmas campaign for Canadians held in China https://t.co/ke7qlriuGc	bbcworld new christmas campaign for canadians held in china	['bbcworld', 'new', 'christmas', 'campaign', 'for', 'canadians', 'held', 'in', 'china']	['bbcworld', 'new', 'christmas', 'campaign', 'canadians', 'held', 'china']	['bbcworld' 'chi 'cam 'canadian'
5	2020-12-10 02:22:59	RT @susie_dent: Word of the day is ‘ultracrepidarian’ (19th century): a presumptuous critic; one who gives opinions and advice on subjects…	susiedent word of the day is ‘ultracrepidarian’ th century a presumptuous critic one who gives opinions and advice on subjects…	['susiedent', 'word', 'of', 'the', 'day', 'is', "'", 'ultracrepidarian', "'", 'th', 'century', 'a', 'presumptuous', 'critic', 'one', 'who', 'gives', 'opinions', 'and', 'advice', 'on', 'subjects…']	['susiedent', 'word', 'day', "'", 'ultracrepidarian', "'", 'th', 'century', 'presumptuous', 'critic', 'one', 'gives', 'opinions', 'advice', 'subjects…']	['susied', ' 'ultracrepi "', 'th', 'c 'presu 'critic 'give', 'o 'subje
6	2020-12-10 02:22:59	@Roland_sanchez2 @Feet1100 um really is I can advertise if I want to. fuck off	rolandsanchez feet um really isi can advertise if i want to fuck off	['rolandsanchez', 'feet', 'um', 'really', 'isi', 'can', 'advertise', 'if', 'i', 'want', 'to', 'fuck', 'off']	['rolandsanchez', 'feet', 'um', 'really', 'isi', 'advertise', 'want', 'fuck']	['rolandsa 'feet', 'um', 'isi', 'ad 'want'

WordCloud

In [89]:

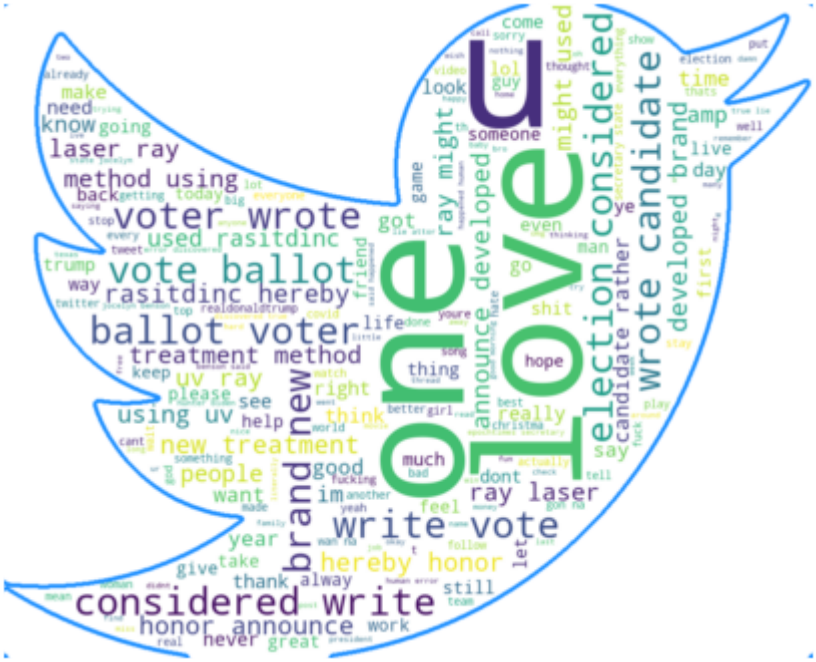
```
from wordcloud import WordCloud, STOPWORDS, ImageColorGenerator
import matplotlib.pyplot as plt
import numpy as np
from PIL import Image
import matplotlib
import matplotlib.pyplot as plt

#%matplotlib inline

def word_cloud(wd_list):
    twitter_mask = np.array(Image.open('extras/twitter_mask.png'))
    stopwords = set(STOPWORDS)
    all_words = ' '.join([text for text in wd_list])
    wordcloud = WordCloud(
        mask=twitter_mask,
        background_color="white",
        width=1000,
        height=600,
        contour_width=5,
        contour_color='dodgerblue',
        #colormap=matplotlib.cm.inferno,
        max_font_size=200).generate(all_words)
    plt.figure(figsize=(10, 6))
    plt.axis('off')
    plt.imshow(wordcloud, interpolation="bilinear");
```

In [90]:

```
lemma_list_of_words=[]
for i in range(0,dataSet_tweets.shape[0]):
    lemma_array=dataSet_tweets['Tweet_lemmatized'][i]
    lemma_array=' '.join(lemma_array)
    lemma_list_of_words.append(lemma_array)
word_cloud(lemma_list_of_words)
```



Vous le savez peut-être mais les algorithmes n'aiment pas les mots… Heureusement pour nous, il existe des méthodes simples permettant de convertir un document en une matrice de mot.

Classification des tweets

La véctorisation

Elle permet de convertir une collection de documents texte en une matrice de nombres de jetons

In [91]:

```
from sklearn.feature_extraction.text import CountVectorizer
countVectorizer = CountVectorizer()
countVector = countVectorizer.fit_transform(lemma_list_of_words)
words = countVectorizer.get_feature_names()
print('{} Number of tweets has {} words'.format(countVector.shape[0], countVector.s
```

```
10965 Number of tweets has 22564 words
```

Convert a collection of text documents to a matrix of token counts

In [92]:

```
count_vect_df = pd.DataFrame(countVector.toarray(), columns=countVectorizer.get_fea
count_vect_df.head(7)
```

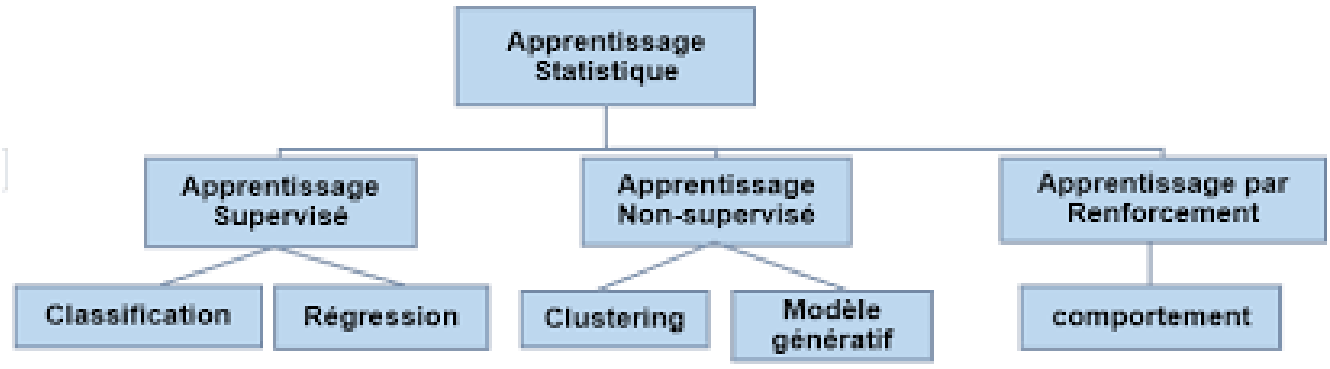
Out[92]:

	aa	aaa	aaaaaaaa	aaaaaaaaa	aaaaaaaaaand	aaaandi	aadandi	aaereign	aamirali	aand	.
0	0	0	0	0	0	0	0	0	0	0	.
1	0	0	0	0	0	0	0	0	0	0	.
2	0	0	0	0	0	0	0	0	0	0	.
3	0	0	0	0	0	0	0	0	0	0	.
4	0	0	0	0	0	0	0	0	0	0	.
5	0	0	0	0	0	0	0	0	0	0	.
6	0	0	0	0	0	0	0	0	0	0	.

7 rows × 22564 columns

3. Modélisation

Dans cette étape, on doit choisir la bonne technique pour extraire les connaissances (exploration) des données. Des techniques telles que les réseaux de neurones, les arbres de décision, les réseaux bayésiens, le clustering, ... sont utilisées. Généralement, l'implémentation se base sur plusieurs de ces techniques, puison choisit le bon résultat. Dans le reste de ce rapport on va détailler les différentes techniques utilisées dans l'exploration des données et l'estimation du modèle.



Dans notre projet on a fait le regroupement des tweets similaires donc il s'agit d'un apprentissage non supervisé car les données ne sont pas étiquetées et pour cela on a utilisé l'algorithme de clustering KMEANS

L'algorithme KMEANS

K-means (ou K-moyennes) : C'est l'un des algorithmes de clustering les plus répandus. Il permet d'analyser un jeu de données caractérisées par un ensemble de descripteurs, afin de regrouper les données "similaires" en groupes (ou clusters).

In [93]:

```
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
import logging
array = [3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,
inertias = []
sil_scores = []
#fit the models, save the evaluation metrics from each run
for i in array:
    #logging.warning('fitting model for {} clusters'.format(i))
    model = KMeans(n_clusters=i,init='k-means++',max_iter=300,n_init=10,random_stat
    model.fit(countVector)
    labels = model.labels_
    sil_scores.append(silhouette_score(countVector, labels))
    inertias.append(model.inertia_)
```

In [94]:

```
common_words = model.cluster_centers_.argsort()[:,-1:-26:-1]
for num, centroid in enumerate(common_words):
    print('Cluster' ,str(num) + ' : ' + ', '.join(words[word] for word in centroid)
```

```
Cluster 0 : swung, voter, major, leave, according, remainers, since,
politicspollss, tory, election, labour, last, pol, geetv, geezlais,
gelo, gelles, gem, gemchaserz, gemesss, gemini, geminiblm, geminifee
d, gemsofbollywood, gen
Cluster 1 : stop, making, deal, fraud, everything, give, big, menta
l, trump, virus, hel, donnie, jenki, get, stopping, slytherjennie, l
ie, realdonaldtrump, love, shipper, paigecams, challenge, gemini, ge
m, gemchaserz
Cluster 2 : epochtimes, human, said, benson, secretary, happened, di
scovered, lie, true, state, jocelyn, error, attor, gelles, gelo, ge
m, ℝℕℕ, gemchaserz, gemesss, geminiblm, geminifeed, gemsofbollywood,
gen, gender, gene
Cluster 3 : want, dont, like, body, people, know, nicer, get, see, r
eally, look, would, life, christmas, men, girl, im, live, one, woma
n, say, someone, go, good, give
Cluster 4 : believe, supremacy, historia, reiss, ℝℕℕ, gen, gem, gemc
haserz, gemesss, gemini, geminiblm, geminifeed, gemsofbollywood, gen
e, gender, gelles, geneenvironment, geneprincipe, general, generalbr
novich, generalhospital, generalize, gelo, geezlais, generate
```

In [95]:

```
from sklearn.cluster import KMeans
i=3
j=0
while i<30:
    while True:
        Y=countVectorizer.transform([lemma_list_of_words[j]])
        Y_pred=model.predict(Y)
        if i == Y_pred:
            print("Tweet"+str(Y_pred)+" : "+dataSet_tweets.Tweet[i])
            j=0
            break
        j+=1
    i+=1
```

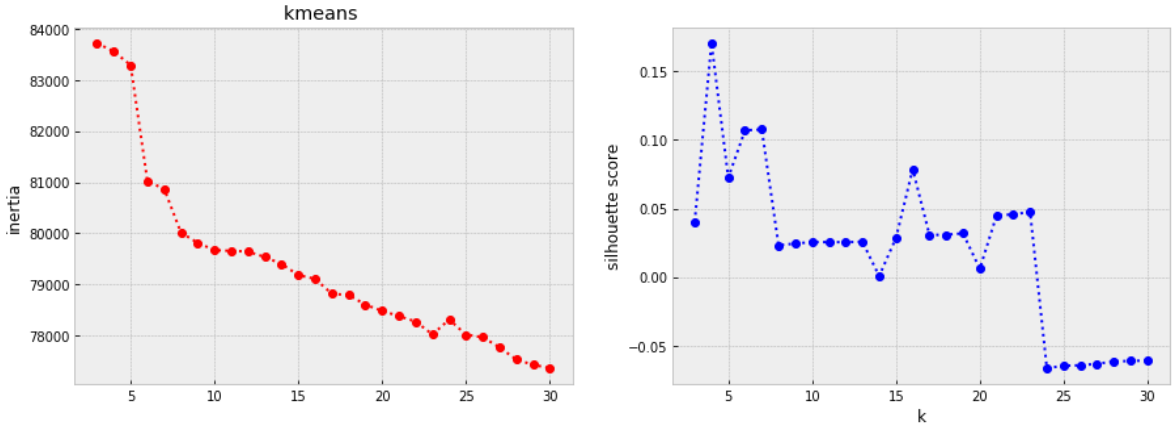
```
Tweet[3] : three hours into ambien and tweeting and she gives you th
is look, wyd https://t.co/YwzAbalKM4 (https://t.co/YwzAbalKM4)
Tweet[4] : RT @BBCWorld: New Christmas campaign for Canadians held i
n China https://t.co/ke7qlriuGc (https://t.co/ke7qlriuGc)
Tweet[5] : RT @susie_dent: Word of the day is ‘ultracrepidarian’ (19
th century): a presumptuous critic; one who gives opinions and advic
e on subjects…
Tweet[6] : @Roland_sanchez2 @Feet1100 um really is
I can advertise if I want to. fuck off
Tweet[7] : @ungodlyjoon ik its okay im kidding😭
Tweet[8] : RT @ChantelJeffries: It me https://t.co/PmfpbyeAVq (http
s://t.co/PmfpbyeAVq)
Tweet[9] : @sunsetamidala THESE VIDEOS SENDING ME INTO CARDIAC ARRES
T
Tweet[10] : @zane @ImNotScottySire @natalinanoel @jasonnash @MariahA
mato  @todderic_ @CorinnaKopf @carlyincontro @eringilfoy… https://t.
co/rWOKDj2Oi9 (https://t.co/rWOKDj2Oi9)
Tweet[11] : RT @Fandango: Rosemary (Emily Blunt) refuses to listen t
o Tony’s (Christopher Walken) advice on love in this exclusive clip
 from Wild Mount
```

In [98]:

```
%matplotlib inline
# plot the quality metrics for inspection
fig, ax = plt.subplots(1, 2, sharex=True,figsize=(15,5))

plt.subplot(121)
plt.plot(array, inertias, 'r:o')
plt.ylabel('inertia')
plt.title('kmeans ')

plt.subplot(122)
plt.plot(array, sil_scores, 'b:o')
plt.ylabel('silhouette score')
plt.xlabel('k');
```



Pour l'instant, partons avec notre meilleure valeur k=16, formons un nouveau modèle sur toutes nos données et continuons notre analyse.

In [99]:

```
best_k = 16
best_model = KMeans(n_clusters=best_k,init='k-means++',n_jobs=-1,max_iter=300,n_ini
best_model.fit(countVector)
```

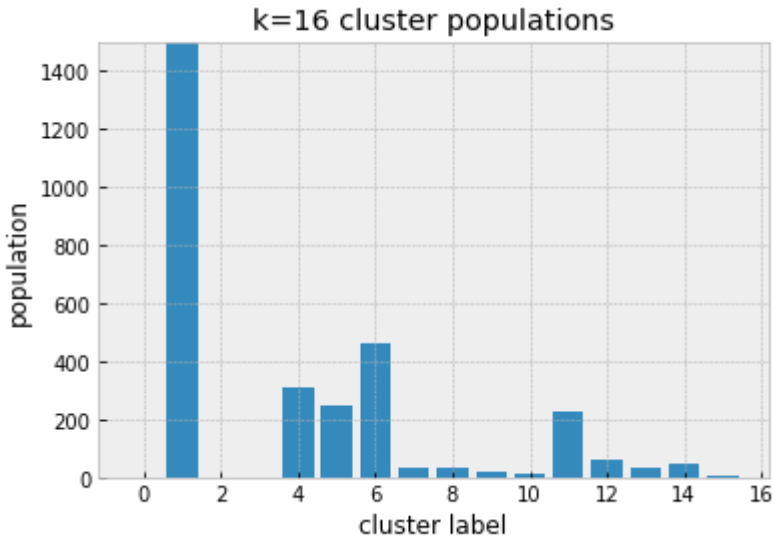
```
C:\Users\Nourhene\AppData\Roaming\Python\Python37\site-packages\sklear
n\cluster\_kmeans.py:939: FutureWarning: 'n_jobs' was deprecated in ve
rsion 0.23 and will be removed in 0.25.
  " removed in 0.25.", FutureWarning)
```

Out[99]:

```
KMeans(n_clusters=16, n_jobs=-1, random_state=42)
```

In [100]:

```
plt.bar(range(len(set(best_model.labels_))), np.bincount(best_model.labels_))
plt.ylabel('population')
plt.xlabel('cluster label')
plt.title('k={} cluster populations'.format(best_k));
# truncating the axis again!
plt.ylim(0,1500);
```



Conclusion

Twitter est une source précieuse de données sur ce qui se passe dans le monde. Les données riches disponibles via la suite d'API fournissent une vue détaillée des personnes et du contenu de la plate-forme. Dans ce didacticiel, nous avons travaillé sur un exemple de flux de travail de bout en bout - de la collecte de données à partir de l'API Twitter à la création et à l'inspection d'un modèle d'utilisateurs de Twitter. En cours de route, nous avons montré comment identifier et utiliser les éléments pertinents de la charge utile de données, comment convertir ces données dans un format compatible avec de nombreuses bibliothèques d'apprentissage automatique et comment inspecter les modèles résultants pour leur interprétabilité. Plus précisément, nous avons **collecté des données csv** correspondantes, **analysé ces données** pour **extraire des informations** spécifiques à l'utilisateur, **appliqué des algorithmes de clustering** aux données texte,et **évalué le modéle choisi**

Réalisé par:

Boulares Nourhéne

In [ ]:

```
```