Mariam Mchirgui 2 MRIAAD

Projet–Fouille de Données Thème : Classification des Tweets .

PROJET CLUSTERING

1. Introduction

Ce projet a été réalisé dans le cadre du cours de Data Mining , avec pour objectif la classification automatique des tweets pour fournir des résumés simplifiés à un utilisateur. Les étapes incluent le prétraitement des tweets, l'analyse NLP, et le regroupement en clusters .

2. Objectifs

- Extraire et nettoyer un ensemble de tweets.
- Appliquer des techniques de traitement du langage naturel (NLP) pour analyser les données.
- Regrouper les tweets en clusters similaires à l'aide des algorithmes K-Means et Fuzzy-C-Means.

3. Outils Utilisés

- **Python :** Langage principal.
- **Bibliothèques :** NLTK, scikit-learn, matplotlib, pandas, numpy.
- **Environnement :** Jupyter Notebook via Anaconda.

4. Méthodologie (**Étapes principales**) :

1 . Le développeur X de Twitter m'a accordé une limite d'accès à 100 tweets par mois via l'API. Étant donné que cette quantité est insuffisante pour une analyse approfondie, j'ai décidé d'utiliser un dataset préexistant disponible sur Kaggle, Ce dataset porte sur les tweets liés à la Coupe du Monde (World Cup) et contient un volume plus important de données .

REF TWEETS.csv

2 . Code Python :

- ## Importation des bibliothèques nécessaires

Description

Cette commande importe les bibliothèques nécessaires pour la manipulation et l'analyse des données. Voici le rôle de chaque bibliothèque :

- pandas : Gestion et manipulation des données structurées.
- numpy : Calculs numériques et manipulation de tableaux.
- re : Traitement des expressions régulières pour le nettoyage des textes.
- nltk : Bibliothèque de traitement du langage naturel (NLP) pour la tokenization et la gestion des mots vides.

```
import pandas as pd
import numpy as np
import re
#import spacy
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
```

- ## Chargement des données

Description

Chargement des données des tweets depuis un fichier CSV.

```
data = pd.read_csv(r"C:\Users\ichig\Downloads\REF TWEETS.csv")
```

- ## Analyse initiale des données

Description

Cette commande affiche les premières lignes du DataFrame pour une vue d'ensemble rapide des données. Elle permet de vérifier que le nettoyage et les transformations ont été appliqués correctement.

```
data.head()
```

	ID	Date Created	Number of Likes	Tweet	Sentiment
0	1	2022-11-21 22:37:33+00:00	5	The first female referee at a Men's World Cup ...	Neutral
1	2	2022-11-21 22:12:57+00:00	1	North Korea will host the World Cup in 2030. W...	Neutral
2	3	2022-11-21 22:07:05+00:00	7	If you're wondering why there has been so much...	Neutral
3	4	2022-11-21 21:48:16+00:00	1	NFL referee: "We will scrutinize every angle o...	Negative
4	5	2022-11-21 21:38:01+00:00	0	Possibly imagining it, but feel like there are...	Neutral

- ## Informations sur les données

Description

Cette commande fournit des informations générales sur le DataFrame, y compris le nombre de lignes, le nombre de colonnes, le type de données pour chaque colonne, et s'il y a des valeurs manquantes (NaN). Cela permet d'obtenir une vue d'ensemble sur la structure et la qualité des données.

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4000 entries, 0 to 3999
Data columns (total 5 columns):
 #   Column           Non-Null Count  Dtype 
---  ------           --------------  ----- 
 0   ID               4000 non-null   int64 
 1   Date Created     4000 non-null   object
 2   Number of Likes  4000 non-null   int64 
 3   Tweet            4000 non-null   object
 4   Sentiment        4000 non-null   object
dtypes: int64(2), object(3)
memory usage: 156.4+ KB
```

Deleting dupliate TWEETS(if any)

- ## Détection des doublons dans les Tweets

Description

Cette commande identifie les doublons dans la colonne Tweet du jeu de données. Les doublons sont triés et affichés pour permettre une vérification ou une suppression ultérieure. Le paramètre keep=False garantit que tous les doublons sont marqués (pas uniquement le premier ou le dernier).

```
data[data['Tweet'].duplicated(keep=False)].sort_values('Tweet').head(8)
```

	ID	Date Created	Number of Likes	Tweet	Sentiment
2994	2995	2022-12-10 21:35:09+00:00	1	#ENGFRA\nGary Neville delivers brutal referee ...	Negative
3067	3068	2022-12-11 10:56:45+00:00	1	#ENGFRA\nGary Neville delivers brutal referee ...	Negative
2144	2145	2022-12-05 22:10:59+00:00	4	Bro I would really like to see croatia or braz...	Negative
2041	2042	2022-12-05 22:10:59+00:00	3	Bro I would really like to see croatia or braz...	Neutral
2115	2116	2022-12-05 21:03:37+00:00	7	By far the worst referee in this World Cup the...	Negative
2071	2072	2022-12-05 21:03:37+00:00	6	By far the worst referee in this World Cup the...	Negative
2116	2117	2022-12-05 20:43:17+00:00	7	Clement turpin is another absolutely terrible ...	Negative
2073	2074	2022-12-05 20:43:17+00:00	4	Clement turpin is another absolutely terrible ...	Negative

- ## Suppression des doublons dans les Tweets

Description

Cette commande supprime les doublons dans la colonne Tweet du DataFrame. Seul le premier tweet unique est conservé, les autres occurrences identiques sont supprimées.

```
data = data.drop_duplicates('Tweet')
```

NLP

- ## Importation du parseur HTML

Description

Cette commande importe le module html ainsi que la classe HTMLParser pour le traitement et le nettoyage des entités HTML. Le parseur est utilisé pour convertir les caractères HTML spéciaux en texte lisible.

```
import html
from html.parser import HTMLParser
html_parser = HTMLParser()
```

- ## Nettoyage des entités HTML dans les Tweets

Description

Cette commande nettoie les entités HTML dans la colonne Tweet en utilisant la fonction html.unescape, qui convertit les entités HTML (comme &) en caractères lisibles (comme &). Le résultat est stocké dans une nouvelle colonne clean_tweet.

```
data['clean_tweet'] = data['Tweet'].apply(lambda x: html.unescape(x))

data.head(10)
```

	ID	Date Created	Number of Likes	Tweet	Sentiment	clean_tweet
0	1	2022-11-21 22:37:33+00:00	5	The first female referee at a Men's World Cup ...	Neutral	The first female referee at a Men's World Cup ...
1	2	2022-11-21 22:12:57+00:00	1	North Korea will host the World Cup in 2030. W...	Neutral	North Korea will host the World Cup in 2030. W...
2	3	2022-11-21 22:07:05+00:00	7	If you're wondering why there has been so much...	Neutral	If you're wondering why there has been so much...
3	4	2022-11-21 21:48:16+00:00	1	NFL referee: "We will scrutinize every angle o...	Negative	NFL referee: "We will scrutinize every angle o...
4	5	2022-11-21 21:38:01+00:00	0	Possibly imagining it, but feel like there are...	Neutral	Possibly imagining it, but feel like there are...
5	6	2022-11-21 21:30:59+00:00	16	I'd love it if one of the footballers at the W...	Neutral	I'd love it if one of the footballers at the W...
6	7	2022-11-21 21:27:15+00:00	0	Fox Sports spent $500M to have US broadcasting...	Negative	Fox Sports spent $500M to have US broadcasting...
7	8	2022-11-21 21:21:17+00:00	32	World Cup matches are set to last up to 100 MI...	Negative	World Cup matches are set to last up to 100 MI...
8	9	2022-11-21 21:14:52+00:00	1172	They always experiment with these changes in t...	Neutral	They always experiment with these changes in t...
9	10	2022-11-21 21:14:19+00:00	0	Full-time: \n\n#USMNT 1\n\n#WAL 1\n\nA tale of t...	Negative	Full-time: \n\n#USMNT 1\n\n#WAL 1\n\nA tale of t...

- ## Fonction de suppression de motif dans un texte

Description

Cette fonction, remove_pattern, prend un texte en entrée (input_txt) et un motif (pattern) sous forme d'expression régulière. Elle recherche toutes les occurrences du motif dans le texte et les supprime en utilisant re.sub. Le texte nettoyé est ensuite renvoyé.

```python
def remove_pattern(input_txt, pattern):
    r = re.findall(pattern, input_txt)
    for i in r:
        input_txt = re.sub(i, '', input_txt)
    return input_txt
```

- ## Nettoyage supplémentaire des Tweets

Description

Cette série de commandes effectue plusieurs étapes de nettoyage sur la colonne clean_tweet :

1. **Suppression des caractères non alphanumériques** : La commande utilise une expression régulière pour supprimer tous les caractères qui ne sont pas des lettres, des chiffres ou des apostrophes.
2. **Suppression des caractères non alphabétiques et non espacés** : La fonction remove_pattern est utilisée pour supprimer tous les autres caractères non alphabétiques et non espacés (comme les signes de ponctuation).
3. **Conversion en minuscules** : Tous les tweets sont convertis en minuscules pour standardiser le texte et éviter les doublons dus à la casse.

Enfin, la commande data['clean_tweet'][9] affiche le 10e tweet nettoyé.

Suppression des caractères non alphanumériques

```python
data['clean_tweet'] = data['clean_tweet'].apply(lambda x: re.sub(r"[^a-zA-Z0-9']",' ',x))
data.head(10)
```

	ID	Date Created	Number of Likes	Tweet	Sentiment	clean_tweet
0	1	2022-11-21 22:37:33+00:00	5	The first female referee at a Men's World Cup ...	Neutral	The first female referee at a Men s World Cup ...
1	2	2022-11-21 22:12:57+00:00	1	North Korea will host the World Cup in 2030. W...	Neutral	North Korea will host the World Cup in 2030 W...
2	3	2022-11-21 22:07:05+00:00	7	If you're wondering why there has been so much...	Neutral	If you're wondering why there has been so much...
3	4	2022-11-21 21:48:16+00:00	1	NFL referee: "We will scrutinize every angle o...	Negative	NFL referee We will scrutinize every angle o...
4	5	2022-11-21 21:38:01+00:00	0	Possibly imagining it, but feel like there are...	Neutral	Possibly imagining it but feel like there are...
5	6	2022-11-21 21:30:59+00:00	16	I'd love it if one of the footballers at the W...	Neutral	I d love it if one of the footballers at the W...
6	7	2022-11-21 21:27:15+00:00	0	Fox Sports spent $500M to have US broadcasting...	Negative	Fox Sports spent 500M to have US broadcasting...
7	8	2022-11-21 21:21:17+00:00	32	World Cup matches are set to last up to 100 MI...	Negative	World Cup matches are set to last up to 100 MI...
8	9	2022-11-21 21:14:52+00:00	1172	They always experiment with these changes in t...	Neutral	They always experiment with these changes in t...
9	10	2022-11-21 21:14:19+00:00	0	Full-time: \n\n#USMNT 1\n#WAL 1\n\nA tale of t...	Negative	Full time USMNT 1 WAL 1 A tale of two ha...

Suppression des caractères non alphabétiques et non espacés

```
data['clean_tweet'] = np.vectorize(remove_pattern)(data['clean_tweet'], r"[^\w\s]")
data.head(10)
```

	ID	Date Created	Number of Likes	Tweet	Sentiment	clean_tweet
0	1	2022-11-21 22:37:33+00:00	5	The first female referee at a Men's World Cup ...	Neutral	The first female referee at a Men s World Cup ...
1	2	2022-11-21 22:12:57+00:00	1	North Korea will host the World Cup in 2030. W...	Neutral	North Korea will host the World Cup in 2030 W...
2	3	2022-11-21 22:07:05+00:00	7	If you're wondering why there has been so much...	Neutral	If youre wondering why there has been so much ...
3	4	2022-11-21 21:48:16+00:00	1	NFL referee: "We will scrutinize every angle o...	Negative	NFL referee We will scrutinize every angle o...
4	5	2022-11-21 21:38:01+00:00	0	Possibly imagining it, but feel like there are...	Neutral	Possibly imagining it but feel like there are...
5	6	2022-11-21 21:30:59+00:00	16	I'd love it if one of the footballers at the W...	Neutral	I d love it if one of the footballers at the W...
6	7	2022-11-21 21:27:15+00:00	0	Fox Sports spent $500M to have US broadcasting...	Negative	Fox Sports spent 500M to have US broadcasting...
7	8	2022-11-21 21:21:17+00:00	32	World Cup matches are set to last up to 100 MI...	Negative	World Cup matches are set to last up to 100 MI...
8	9	2022-11-21 21:14:52+00:00	1172	They always experiment with these changes in t...	Neutral	They always experiment with these changes in t...
9	10	2022-11-21 21:14:19+00:00	0	Full-time: \n\n#USMNT 1\n#WAL 1\n\nA tale of t...	Negative	Full time USMNT 1 WAL 1 A tale of two ha...

Conversion en minuscules

```
data['clean_tweet'] = data['clean_tweet'].apply(lambda x: x.lower())
data.head(10)
```

	ID	Date Created	Number of Likes	Tweet	Sentiment	clean_tweet
0	1	2022-11-21 22:37:33+00:00	5	The first female referee at a Men's World Cup ...	Neutral	the first female referee at a men s world cup ...
1	2	2022-11-21 22:12:57+00:00	1	North Korea will host the World Cup in 2030. W...	Neutral	north korea will host the world cup in 2030 w...
2	3	2022-11-21 22:07:05+00:00	7	If you're wondering why there has been so much...	Neutral	if youre wondering why there has been so much ...
3	4	2022-11-21 21:48:16+00:00	1	NFL referee: "We will scrutinize every angle o...	Negative	nfl referee we will scrutinize every angle o...
4	5	2022-11-21 21:38:01+00:00	0	Possibly imagining it, but feel like there are...	Neutral	possibly imagining it but feel like there are...
5	6	2022-11-21 21:30:59+00:00	16	I'd love it if one of the footballers at the W...	Neutral	i d love it if one of the footballers at the w...
6	7	2022-11-21 21:27:15+00:00	0	Fox Sports spent $500M to have US broadcasting...	Negative	fox sports spent 500m to have us broadcasting...
7	8	2022-11-21 21:21:17+00:00	32	World Cup matches are set to last up to 100 MI...	Negative	world cup matches are set to last up to 100 mi...
8	9	2022-11-21 21:14:52+00:00	1172	They always experiment with these changes in t...	Neutral	they always experiment with these changes in t...
9	10	2022-11-21 21:14:19+00:00	0	Full-time: \n\n#USMNT 1\n#WAL 1\n\nA tale of t...	Negative	full time usmnt 1 wal 1 a tale of two ha...

Affichage le 10e tweet nettoyé

```
data['clean_tweet'][9]
```

```
'full time     usmnt 1  wal 1  a tale of two halfs were the usmnt was more dominant in the first and wales dominated in the second  also this referee cre
w put on the most concacfiest referring performance in the history of the world cup it was bad     fifaworldcup  qatar2022'
```

- ## Dictionnaire des Contractions

Description

Ce dictionnaire apostrophe_dict contient des paires de mots, où chaque clé est une contraction courante en anglais (par exemple, "ain't", "can't", "don't") et chaque valeur est sa forme complète (par exemple, "am not / are not", "cannot", "do not"). Il peut être utilisé pour étendre ou remplacer les contractions par leurs équivalents complets dans le texte.

```
# Apostrophe Dictionary
apostrophe_dict = {
"ain't": "am not / are not",
"aren't": "are not / am not",
"can't": "cannot",
"can't've": "cannot have",
"'cause": "because",
"could've": "could have",
"couldn't": "could not",
"couldn't've": "could not have",
"didn't": "did not",
"doesn't": "does not",
"don't": "do not",
"hadn't": "had not",
"hadn't've": "had not have",
"hasn't": "has not",
"haven't": "have not",
"he'd": "he had / he would",
"he'd've": "he would have",
"he'll": "he shall / he will",
"he'll've": "he shall have / he will have",
"he's": "he has / he is",
"how'd": "how did",
"how'd'y": "how do you",
"how'll": "how will",
"how's": "how has / how is",
"i'd": "I had / I would",
```

- ## Fonction de remplacement des contractions par leurs équivalents complets

Description

La fonction lookup_dict prend deux arguments : un texte (text) et un dictionnaire (dictionary). Elle commence par remplacer les sauts de ligne par des espaces pour assurer que le texte est bien formaté. Ensuite, pour chaque mot du texte, elle vérifie si ce mot (en minuscules) se trouve dans le dictionnaire. Si c'est le cas, le mot est remplacé par son équivalent complet dans le dictionnaire. Cette méthode permet de traiter les contractions et de rendre le texte plus standardisé.

```
def lookup_dict(text, dictionary):
    text = text.replace("\n", " ")
    for word in text.split():
        if word.lower() in dictionary:
            if word.lower() in text.split():
                text = text.replace(word, dictionary[word.lower()])
    return text
```

- ## Remplacement des contractions par leurs équivalents complets dans les Tweets

Description

La commande applique la fonction lookup_dict sur chaque tweet de la colonne clean_tweet afin de remplacer les contractions par leurs équivalents complets. La fonction lookup_dict parcourt chaque tweet et remplace les mots présents dans le dictionnaire apostrophe_dict. Cela permet de normaliser les tweets en remplaçant des contractions comme "ain't" par "am not / are not", ce qui peut améliorer la précision des analyses ultérieures telles que le clustering ou la classification des tweets. Le résultat nettoyé est stocké dans la même colonne clean_tweet.

```
data['clean_tweet'] = data['clean_tweet'].apply(lambda x: lookup_dict(x,apostrophe_dict))
data.head(10)
```

	ID	Date Created	Number of Likes	Tweet	Sentiment	clean_tweet
0	1	2022-11-21 22:37:33+00:00	5	The first female referee at a Men's World Cup ...	Neutral	the first female referee at a men s world cup ...
1	2	2022-11-21 22:12:57+00:00	1	North Korea will host the World Cup in 2030. W...	Neutral	north korea will host the world cup in 2030 w...
2	3	2022-11-21 22:07:05+00:00	7	If you're wondering why there has been so much...	Neutral	if youre wondering why there has been so much ...
3	4	2022-11-21 21:48:16+00:00	1	NFL referee: "We will scrutinize every angle o...	Negative	nfl referee we will scrutinize every angle o...
4	5	2022-11-21 21:38:01+00:00	0	Possibly imagining it, but feel like there are...	Neutral	possibly imagining it but feel like there are...
5	6	2022-11-21 21:30:59+00:00	16	I'd love it if one of the footballers at the W...	Neutral	i d love it if one of the footballers at the w...
6	7	2022-11-21 21:27:15+00:00	0	Fox Sports spent $500M to have US broadcasting...	Negative	fox sports spent 500m to have us broadcasting...
7	8	2022-11-21 21:21:17+00:00	32	World Cup matches are set to last up to 100 Ml...	Negative	world cup matches are set to last up to 100 mi...
8	9	2022-11-21 21:14:52+00:00	1172	They always experiment with these changes in t...	Neutral	they always experiment with these changes in t...
9	10	2022-11-21 21:14:19+00:00	0	Full-time: \n\n#USMNT 1\n#WAL 1\n\nA tale of t...	Negative	full time usmnt 1 wal 1 a tale of two ha...

- ## Dictionnaire des abréviations courantes et de leurs significations

Description

Le dictionnaire short_word_dict contient une liste d'abréviations populaires et leurs significations complètes. Par exemple, "afaik" est remplacé par "as far as I know" et "omg" par "oh my god". Ce dictionnaire est utile dans le cadre de l'analyse de textes, en particulier pour les tweets ou autres messages informels où les abréviations sont courantes. En développant ces abréviations, on peut obtenir un texte plus compréhensible et plus standardisé pour l'analyse, ce qui est essentiel pour une extraction d'information efficace

```
short_word_dict = {
"121": "one to one",
"a/s/l": "age, sex, location",
"adn": "any day now",
"afaik": "as far as I know",
"afk": "away from keyboard",
"aight": "alright",
"alol": "actually laughing out loud",
"b4": "before",
"b4n": "bye for now",
"bak": "back at the keyboard",
"bf": "boyfriend",
"bff": "best friends forever",
"bfn": "bye for now",
"bg": "big grin",
"bta": "but then again",
"btw": "by the way",
"cid": "crying in disgrace",
"cnp": "continued in my next post",
"cp": "chat post",
"cu": "see you",
"cul": "see you later",
"cul8r": "see you later",
"cya": "bye",
"cyo": "see you online",
"dbau": "doing business as usual",
"fud": "fear, uncertainty, and doubt",
```

- ## Remplacement des abréviations par leurs significations complètes dans les Tweets

Description

La commande applique la fonction lookup_dict à chaque tweet de la colonne clean_tweet afin de remplacer les abréviations par leurs significations complètes en utilisant le dictionnaire short_word_dict. Par exemple, "afaik" sera remplacé par "as far as I know" et "lol" par "laugh out loud". Cette opération permet de clarifier le texte en remplaçant les abréviations informelles.

```
data['clean_tweet'] = data['clean_tweet'].apply(lambda x: lookup_dict(x,short_word_dict))
data.head(10)
```

	ID	Date Created	Number of Likes	Tweet	Sentiment	clean_tweet
0	1	2022-11-21 22:37:33+00:00	5	The first female referee at a Men's World Cup ...	Neutral	the first female referee at a men s world cup ...
1	2	2022-11-21 22:12:57+00:00	1	North Korea will host the World Cup in 2030. W...	Neutral	north korea will host the world cup in 2030 w...
2	3	2022-11-21 22:07:05+00:00	7	If you're wondering why there has been so much...	Neutral	if youre wondering why there has been so much ...
3	4	2022-11-21 21:48:16+00:00	1	NFL referee: "We will scrutinize every angle o...	Negative	National Football League referee we will scr...
4	5	2022-11-21 21:38:01+00:00	0	Possibly imagining it, but feel like there are...	Neutral	possibly imagrininingrin it but feel like the...
5	6	2022-11-21 21:30:59+00:00	16	I'd love it if one of the footballers at the W...	Neutral	i d love it if one of the footballers at the w...
6	7	2022-11-21 21:27:15+00:00	0	Fox Sports spent $500M to have US broadcasting...	Negative	fox sports spent 500m to have us broadcasting...
7	8	2022-11-21 21:21:17+00:00	32	World Cup matches are set to last up to 100 MI...	Negative	world cup matches are set to last up to 100 mi...
8	9	2022-11-21 21:14:52+00:00	1172	They always experiment with these changes in t...	Neutral	they always experiment with these changes in t...
9	10	2022-11-21 21:14:19+00:00	0	Full-time: \n\n#USMNT 1\n#WAL 1\n\nA tale of t...	Negative	full time United States Men's National Tea...

- ## Dictionnaire des émoticônes

Description

Le dictionnaire emoticon_dict contient une liste d'émoticônes courantes, comme ":)" ou ":-(", et les associe à un espace vide. Ce dictionnaire peut être utilisé pour supprimer ces émoticônes dans un texte, ce qui est souvent nécessaire dans les prétraitements de texte, car elles n'ajoutent pas d'information utile pour les analyses textuelles comme le clustering ou la classification. En remplaçant les émoticônes par un espace, cette opération nettoie le texte sans altérer sa structure.

```
emoticon_dict = {
    ":)": " ",
    ":-)": " ",
    ":-]": " ",
    ":-3": " ",
    ":->": " ",
    "8-)": " ",
    ":-}": " ",
    ":o)": " ",
    ":c)": " ",
    ":^)": " ",
    "=]": " ",
    "=)": " ",
    "<3": " ",
    ":-(": " ",
    ":(": " ",
    ":c": " ",
    ":<": " ",
    ":[": " ",
    ">:(": " ",
    ":{": " ",
    ":-c": " ",
    ":-< ": " ",
    ":-[": " ",
    ":-||": " "
}
```

- ## Suppression des émoticônes dans les Tweets

Description

La commande applique la fonction lookup_dict à chaque tweet de la colonne clean_tweet, permettant de supprimer toutes les émoticônes en se basant sur le dictionnaire emoticon_dict. Par exemple, ":)" et ":-(" sont supprimées et remplacées par des espaces vides. Cette étape est importante dans le traitement des données textuelles, car les émoticônes peuvent ne pas ajouter de valeur significative pour l'analyse et peuvent interférer avec le modèle. Le texte nettoyé est ensuite stocké dans la colonne clean_tweet.

```
data['clean_tweet'] = data['clean_tweet'].apply(lambda x: lookup_dict(x,emoticon_dict))
data.head(50)
```

	ID	Date Created	Number of Likes	Tweet	Sentiment	clean_tweet
0	1	2022-11-21 22:37:33+00:00	5	The first female referee at a Men's World Cup ...	Neutral	the first female referee at a men s world cup ...
1	2	2022-11-21 22:12:57+00:00	1	North Korea will host the World Cup in 2030. W...	Neutral	north korea will host the world cup in 2030 w...
2	3	2022-11-21 22:07:05+00:00	7	If you're wondering why there has been so much...	Neutral	if youre wondering why there has been so much ...
3	4	2022-11-21 21:48:16+00:00	1	NFL referee: "We will scrutinize every angle o...	Negative	National Football League referee we will scr...
4	5	2022-11-21 21:38:01+00:00	0	Possibly imagining it, but feel like there are...	Neutral	possibly imagrininingrin it but feel like the...
5	6	2022-11-21 21:30:59+00:00	16	I'd love it if one of the footballers at the W...	Neutral	i d love it if one of the footballers at the w...
6	7	2022-11-21 21:27:15+00:00	0	Fox Sports spent $500M to have US broadcasting...	Negative	fox sports spent 500m to have us broadcasting...
7	8	2022-11-21 21:21:17+00:00	32	World Cup matches are set to last up to 100 MI...	Negative	world cup matches are set to last up to 100 mi...
8	9	2022-11-21 21:14:52+00:00	1172	They always experiment with these changes in t...	Neutral	they always experiment with these changes in t...
9	10	2022-11-21 21:14:19+00:00	0	Full-time: \n\n#USMNT 1\n#WAL 1\n\nA tale of t...	Negative	full time United States Men's National Tea...

- ## Téléchargement du modèle de tokenization de NLTK

Description

La commande utilise la bibliothèque NLTK (Natural Language Toolkit) pour télécharger le modèle punkt, qui est un modèle pré-entraîné pour la tokenization. Ce modèle permet de diviser un texte en unités plus petites, comme des mots ou des phrases. La tokenization est une étape clé dans le traitement du langage naturel, car elle permet de manipuler plus facilement les textes pour des analyses ultérieures telles que le comptage de mots, la classification ou le clustering.

```
import nltk
nltk.download('punkt')
```

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

- ## Création de tokens pour les tweets nettoyés

Description

La commande applique la fonction word_tokenize de la bibliothèque NLTK à chaque tweet dans la colonne clean_tweet pour diviser chaque tweet en tokens (unités de texte, généralement des mots). Ces tokens sont ensuite stockés dans une nouvelle colonne tweet_token. La tokenization est une étape importante dans le prétraitement des textes, car elle permet de séparer les mots individuels, ce qui facilite l'analyse des données textuelles, comme le comptage des mots, l'extraction de features ou l'entraînement de modèles.

```
# Creating token for the clean tweets
data['tweet_token'] = data['clean_tweet'].apply(lambda x: word_tokenize(x))

## Fully formated tweets & there tokens
data.head(10)
```

	ID	Date Created	Number of Likes	Tweet	Sentiment	clean_tweet	tweet_token
0	1	2022-11-21 22:37:33+00:00	5	The first female referee at a Men's World Cup ...	Neutral	the first female referee at a men s world cup ...	[the, first, female, referee, at, a, men, s, w...
1	2	2022-11-21 22:12:57+00:00	1	North Korea will host the World Cup in 2030. W...	Neutral	north korea will host the world cup in 2030 w...	[north, korea, will, host, the, world, cup, in...
2	3	2022-11-21 22:07:05+00:00	7	If you're wondering why there has been so much...	Neutral	if youre wondering why there has been so much ...	[if, youre, wondering, why, there, has, been, ...
3	4	2022-11-21 21:48:16+00:00	1	NFL referee: "We will scrutinize every angle o...	Negative	National Football League referee we will scr...	[National, Football, League, referee, we, will...
4	5	2022-11-21 21:38:01+00:00	0	Possibly imagining it, but feel like there are...	Neutral	possibly imagrininingrin it but feel like the...	[possibly, imagrininingrin, it, but, feel, lik...
5	6	2022-11-21 21:30:59+00:00	16	I'd love it if one of the footballers at the W...	Neutral	i d love it if one of the footballers at the w...	[i, d, love, it, if, one, of, the, footballers...
6	7	2022-11-21 21:27:15+00:00	0	Fox Sports spent $500M to have US broadcasting...	Negative	fox sports spent 500m to have us broadcasting...	[fox, sports, spent, 500m, to, have, us, broad...

- ## Téléchargement et préparation des mots vides (stopwords)

Description

La commande télécharge le corpus de mots vides stopwords de la bibliothèque NLTK, puis crée un ensemble (set) de mots vides en anglais. Les mots vides sont des mots courants, comme "is", "the", "and", qui n'apportent généralement pas d'informations significatives pour l'analyse de texte. En les supprimant des tweets, on peut se concentrer sur les mots plus pertinents.

```
nltk.download('stopwords')
stop_words = set(stopwords.words('english'))
```

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

- ## Filtrage des mots vides (stop words) dans les tokens

Description

La commande applique un filtrage sur les tokens de chaque tweet stocké dans la colonne tweet_token. Elle élimine les mots vides (stop words) en utilisant l'ensemble stop_words téléchargé précédemment. Le résultat, qui est une version des tokens sans les mots vides, est stocké dans une nouvelle colonne tweet_token_filtered.

```
data['tweet_token_filtered'] = data['tweet_token'].apply(lambda x: [word for word in x if not word in stop_words])

## Tokens columns with stop words and without stop words
data[['tweet_token', 'tweet_token_filtered']].head(10)
```

	tweet_token	tweet_token_filtered
0	[the, first, female, referee, at, a, men, s, w...	[first, female, referee, men, world, cup, phil...
1	[north, korea, will, host, the, world, cup, in...	[north, korea, host, world, cup, 2030, winning...
2	[if, youre, wondering, why, there, has, been, ...	[youre, wondering, much, stoppage, time, world...
3	[National, Football, League, referee, we, will...	[National, Football, League, referee, scrutini...
4	[possibly, imagrininingrin, it, but, feel, lik...	[possibly, imagrininingrin, feel, like, fewer,...
5	[i, d, love, it, if, one, of, the, footballers...	[love, one, footballers, world, cup, actually,...
6	[fox, sports, spent, 500m, to, have, us, broad...	[fox, sports, spent, 500m, us, broadcasting, r...
7	[world, cup, matches, are, set, to, last, up, ...	[world, cup, matches, set, last, 100, minutes,...
8	[they, always, experiment, with, these, change...	[always, experiment, changes, world, cup, roll...
9	[full, time, United, States, Men, 's, National...	[full, time, United, States, Men, 's, National...

- ## Affichage des tokens filtrés pour un tweet spécifique

Description

La commande extrait et affiche les tokens filtrés du 67e tweet (index 66) dans la colonne tweet_token_filtered. Ces tokens représentent les mots significatifs du tweet après avoir supprimé les mots vides (comme "the", "and", etc.), permettant ainsi de mieux analyser le contenu textuel en se concentrant sur les mots clés.

```
data['tweet_token_filtered'][66]
```

```
['referee', 'world', 'cup', 'seems', 'like', 'stressful', 'occupation']
```

- ## Stemming des tokens pour chaque tweet

Description

La commande utilise le PorterStemmer de NLTK pour effectuer un **stemming** sur chaque token filtré dans la colonne tweet_token_filtered. Le stemming est un processus qui réduit les mots à leur racine (par exemple, "running" devient "run"). Cela permet de traiter des variantes de mots comme une seule unité, ce qui est utile pour des analyses telles que le comptage de mots ou le clustering. Les tweets stemmés sont ensuite stockés dans la nouvelle colonne tweet_stemmed.

```
from nltk.stem import PorterStemmer
stemming = PorterStemmer()
```

```
# Created one more columns tweet_stemmed it shows tweets' stemmed version
data['tweet_stemmed'] = data['tweet_token_filtered'].apply(lambda x: ' '.join([stemming.stem(i) for i in x]))
data['tweet_stemmed'].head(10)
```

```
0       first femal refere men world cup philli tough
1    north korea host world cup 2030 win north kore...
2    your wonder much stoppag time world cup check ...
3    nation footbal leagu refere scrutin everi angl...
4    possibl imagrininingrin feel like fewer causti...
5    love one footbal world cup actual wear one lov...
6    fox sport spent 500m us broadcast right 2022 f...
7    world cup match set last 100 minut cut time wa...
8    alway experi chang world cup roll leagu first ...
9    full time unit state men 's nation team 1 wal ...
Name: tweet_stemmed, dtype: object
```

- ## Téléchargement du corpus WordNet et initialisation du Lemmatizer

Description

La commande télécharge le corpus WordNet de NLTK, qui contient une grande base de données lexicale pour l'anglais, et initialise le **WordNet Lemmatizer**. Contrairement au stemming, qui coupe les mots pour en extraire la racine, la lemmatisation réduit un mot à sa forme canonique ou de base (par exemple, "better" devient "good"). Cela permet d'améliorer la précision des analyses textuelles en prenant en compte les formes correctes des mots. Le téléchargement de WordNet est nécessaire pour que le lemmatiseur fonctionne correctement.

```
from nltk.stem.wordnet import WordNetLemmatizer
lemmatizing = WordNetLemmatizer()
```

```
nltk.download('wordnet')
```

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

```
True
```

- ## Lemmatisation des tokens filtrés dans les tweets

Description

La commande utilise le **WordNetLemmatizer** de NLTK pour effectuer une lemmatisation sur chaque token filtré dans la colonne tweet_token_filtered. La lemmatisation est une technique qui permet de réduire les mots à leur forme canonique, par exemple, "better" devient "good". Contrairement au stemming, qui peut produire des racines qui ne sont pas des mots réels, la lemmatisation donne des résultats plus précis et lisibles. Le texte transformé est ensuite stocké dans la colonne tweet_lemmatized.

```
data['tweet_lemmatized'] = data['tweet_token_filtered'].apply(lambda x: ' '.join([lemmatizing.lemmatize(i) for i in x]))
data['tweet_lemmatized'].head(10)
```

```
0      first female referee men world cup philly tough
1    north korea host world cup 2030 winning north ...
2    youre wondering much stoppage time world cup c...
3    National Football League referee scrutinize ev...
4    possibly imagrininingrin feel like fewer caust...
5    love one footballer world cup actually wear on...
6    fox sport spent 500m u broadcasting right 2022...
7    world cup match set last 100 minute cut time w...
8    always experiment change world cup rolling lea...
9    full time United States Men 's National Team 1...
Name: tweet_lemmatized, dtype: object
```

- ## Comptage des éléments (tokens) dans chaque tweet

Description

La commande crée une nouvelle colonne num_elementos dans le DataFrame df_copy qui contient le nombre de tokens dans chaque tweet. La fonction count_elements prend un vecteur (ici un tweet sous forme de liste de tokens) et retourne le nombre d'éléments dans ce vecteur. Cela permet d'avoir une métrique simple pour chaque tweet, ce qui peut être utile pour analyser la longueur des tweets ou pour des tâches comme la normalisation de la taille des entrées dans un modèle d'analyse de texte.

```
df_copy = data
def count_elements(vector):
    return pd.Series([len(vector)])
```

```
df_copy['num_elementos'] = df_copy['tweet_token'].apply(count_elements)

df_copy
```

3962 rows × 11 columns

- ## Transformation des tweets en matrices de termes

Description

La commande effectue plusieurs étapes de transformation des tweets en matrices de termes à l'aide de deux méthodes différentes :

1. **CountVectorizer** : Cette méthode crée une matrice de termes en comptant la fréquence d'apparition de chaque mot dans les tweets après stemming. La somme de tous les mots présents dans la matrice donne le nombre total de mots.
2. **TfidfVectorizer** : Cette méthode transforme les tweets en une matrice de termes en utilisant la pondération TF-IDF (Term Frequency - Inverse Document Frequency), qui reflète l'importance d'un mot dans un document par rapport à l'ensemble des documents. La somme de tous les termes dans cette matrice donne également le nombre total de mots pondérés.

Ces transformations permettent de préparer les données pour des modèles d'apprentissage automatique et de mesurer l'importance des mots dans les tweets.

```
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
```

```
count_vectorizer = CountVectorizer()
dtm = count_vectorizer.fit_transform(data['tweet_stemmed'])
```

```
total_words = dtm.sum()
total_words
```

```
60392
```

```
count_vectorizer = CountVectorizer()
dtm = count_vectorizer.fit_transform(data['tweet_lemmatized'])
total_words = dtm.sum()
total_words
```

```
60071
```

```
tfidf_vectorizer = TfidfVectorizer()
tfidf = tfidf_vectorizer.fit_transform(data['tweet_lemmatized'])
total_words = tfidf.sum()
total_words
```

```
13261.684259530111
```

- ## Création d'un sous-ensemble aléatoire des tweets lemmatisés

Description

La commande sélectionne un sous-ensemble aléatoire de 10% des tweets lemmatisés dans la colonne tweet_lemmatized du DataFrame data. L'échantillonnage est effectué sans remplacement, ce qui signifie qu'un tweet ne sera sélectionné qu'une seule fois. La taille du sous-ensemble est calculée en fonction du nombre total de lignes dans le DataFrame (num_rows). Un générateur de nombres aléatoires est initialisé avec une graine fixe (np.random.seed(42)) pour garantir la reproductibilité des résultats. Le sous-ensemble ainsi généré est stocké dans subset_df.

```
2]: num_rows = data.tweet_lemmatized.shape[0]


    subset_size = int(num_rows * 0.1)

    np.random.seed(42)

    random_indices = np.random.choice(num_rows, subset_size, replace=False)

    subset_df = data.tweet_lemmatized.iloc[random_indices]

    subset_df
```

```
2]: 149     argentina many star player get penalty voucher...
    1026                      crew ref another world cup game
    1289    watching world cup favor follow referee middle...
    720     current fav moment worldcup alywagner announci...
    325     yo fifacom fifaworldcup accept stupidity refer...
                                ...
    1075                         fuck ref italy world cup
    1457    victor gomes best ref south africa surprise he...
    2302    new interview poland goalie szcz sny said half...
    69      sure ref qatar got referee game world cup cant...
    3746                          ref world cup final please
    Name: tweet_lemmatized, Length: 396, dtype: object
```

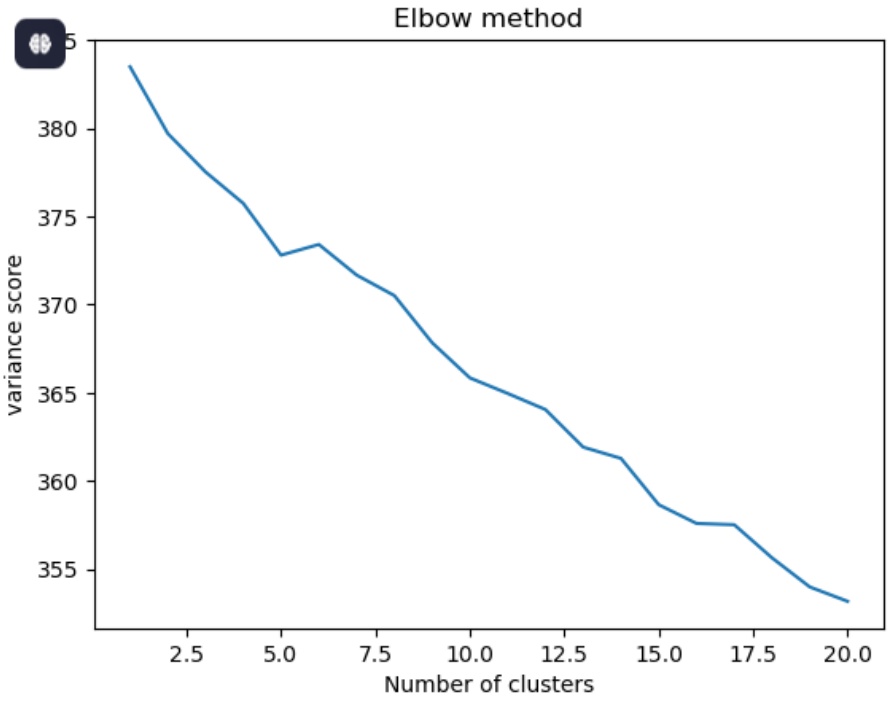
- ## Application de K-Means pour déterminer le nombre optimal de clusters

Description

La commande applique l'algorithme **K-Means** à un sous-ensemble de 10% des tweets lemmatisés, transformés en une matrice TF-IDF. Elle exécute le clustering pour un nombre de clusters allant de 1 à 20, et calcule la **variance intra-cluster** (ou inertie) pour chaque valeur de n_clusters. Cette mesure est utilisée pour déterminer le nombre optimal de clusters en appliquant la **méthode du coude** (Elbow method). Cette méthode consiste à observer où l'inertie commence à diminuer à un rythme plus lent, indiquant ainsi un nombre approprié de clusters. Enfin, un graphique est tracé pour visualiser les scores d'inertie en fonction du nombre de clusters.

```
4]: from sklearn.cluster import KMeans
X = tfidf_vectorizer.fit_transform(subset_df)
max_clusters = 20
scores = []

for n_clusters in range(1, max_clusters + 1):
    kmeans = KMeans(n_clusters=n_clusters)
    kmeans.fit(X)
    scores.append(kmeans.inertia_)
```

```
6]: # Elbow method
import matplotlib.pyplot as plt
plt.plot(range(1, max_clusters + 1), scores)
plt.xlabel('Number of clusters')
plt.ylabel(' variance score ')
plt.title('Elbow method')
plt.show()
```



- ## Application de K-Means avec 7 clusters

Description

La commande applique **K-Means** avec un nombre de clusters fixé à 7 sur la matrice TF-IDF des tweets lemmatisés. L'initialisation k-means++ est utilisée pour sélectionner intelligemment les centres des clusters au début, ce qui améliore la convergence de l'algorithme. La variable random_state=42 permet de garantir que les résultats sont reproductibles. La fonction fit_predict est utilisée pour ajuster le modèle aux données et prédire les étiquettes de cluster pour chaque tweet. Le modèle résultant est stocké dans kmeansmodel, et les étiquettes des clusters sont stockées dans y_kmeans.

```
: kmeansmodel = KMeans(n_clusters=7,init='k-means++', random_state = 42)
y_kmeans= kmeansmodel.fit_predict(X)
```

```
: kmeansmodel
```

```
: KMeans
KMeans(n_clusters=7, random_state=42)
```

- ## Transformation des tweets lemmatisés en features binaires

Description

La commande applique le **CountVectorizer** de **scikit-learn** pour transformer les tweets lemmatisés en une matrice binaire. Le paramètre stop_words='english' permet de supprimer les mots vides (comme "the", "and", etc.), tandis que l'option binary=True indique que la matrice résultante doit être binaire, c'est-à-dire que chaque cellule de la matrice contient 1 si le terme est présent dans le tweet, et 0 si le terme est absent. Cette transformation permet de traiter les données textuelles comme un ensemble de caractéristiques binaires pour des modèles d'apprentissage automatique, ce qui est utile dans des tâches comme la classification de texte.

```
vectorizer = CountVectorizer(stop_words='english', binary=True)  # Binary to treat as set-like presence
binary_features = vectorizer.fit_transform(data.tweet_lemmatized)
```

- ## Calcul des distances de Jaccard entre les tweets

Description

La commande convertit d'abord la matrice binaire, obtenue par CountVectorizer, en un tableau dense avec .toarray(). Ensuite, elle utilise la fonction pairwise_distances de scikit-learn pour calculer les distances de **Jaccard** entre chaque paire de tweets. La **distance de Jaccard** est une mesure de similarité qui compare la présence et l'absence de termes entre deux ensembles, en calculant le rapport de l'intersection à l'union des termes présents dans les tweets. Cette matrice de distances est utile pour des tâches comme le clustering ou la classification, où la similarité entre les éléments doit être prise en compte.

```
2]: dense_matrix = binary_features.toarray()

    # Compute pairwise Jaccard distances
    from sklearn.metrics import pairwise_distances
    jaccard_distance_matrix = pairwise_distances(dense_matrix, metric='jaccard')

    C:\Users\ichig\anaconda3\Lib\site-packages\sklearn\metrics\pairwise.py:2361: DataConversionWarning: Data was converted to boolean for metric jaccard
      warnings.warn(msg, DataConversionWarning)

8]: kmeans=KMeans(n_clusters=7, random_state=42)
    kmeans.fit(jaccard_distance_matrix)

8]: KMeans
    KMeans(n_clusters=7, random_state=42)

0]: tweets = data['Tweet'].tolist()
    tweets
```

- ## Obtention des centroïdes et des étiquettes de clusters

Description

Après avoir appliqué K-Means sur la matrice de distances de Jaccard, la commande récupère les **centroïdes des clusters** avec kmeans.cluster_centers_. Les centroïdes représentent les centres des 7 clusters dans l'espace des caractéristiques. Ensuite, les **étiquettes de clusters** sont attribuées à chaque tweet

à l'aide de kmeans.labels_, qui indique à quel cluster chaque tweet appartient. Ces étiquettes sont utiles pour analyser la composition des clusters et identifier les groupes de tweets similaires.

```
# Obtener los centroides de los clusters
centroids = kmeans.cluster_centers_
# Asignar etiquetas de cluster a cada tweet
cluster_labels = kmeans.labels_

cluster_labels
```

```
array([2, 6, 6, ..., 5, 6, 5])
```

- ## Affichage des tweets par cluster

Description

La commande crée un DataFrame results qui associe chaque tweet à son label de cluster respectif en utilisant les colonnes tweet (contenant les tweets) et cluster (contenant les étiquettes des clusters). Ensuite, elle affiche les tweets de chaque cluster en utilisant une boucle pour parcourir chaque cluster (de 0 à 6). Pour chaque cluster, les tweets correspondants sont extraits et affichés. Cette étape permet de visualiser la composition des différents clusters et d'analyser les groupes de tweets similaires.

[124]:
```
results = pd.DataFrame({'tweet': tweets,'cluster': cluster_labels})
for cluster in range(7):
    print(f"Cluster {cluster}:")
    cluster_users = results[results['cluster'] == cluster]
    print(cluster_users)
```

```
Cluster 0:
                                                  tweet  cluster
4     Possibly imagining it, but feel like there are...        0
15    Tyler Adams was the best player on the pitch t...        0
18    One of the worst referee jobs I've had the dis...        0
20    I know it's the oldest trope in the book, but ...        0
175   They must be using a timekeeper at this World ...        0
...                                                 ...      ...
3823  argentina doesn’t deserve it. they played so d...        0
3826  Pov: Thouran clearly gets fouled in the penalt...        0
3835  Today proved that Argentina can only win world...        0
3837  Iya yin! Infact Baba Iya yin! If I hear, it wa...        0
3862  Been notable for lots of things this game. Ref...        0

[882 rows x 2 columns]
Cluster 1:
                                                  tweet  cluster
204   #FIFAWorldCup\n#Canada played brilliantly till...        1
1243  with all this uproar about the world cup, one ...        1
1632  if i flop on getting tickets to the world cup ...        1
2178  r we gonna talk about this ref not calling any...        1
2194  The referee in this Cameroon/Brazil game looks...        1
```

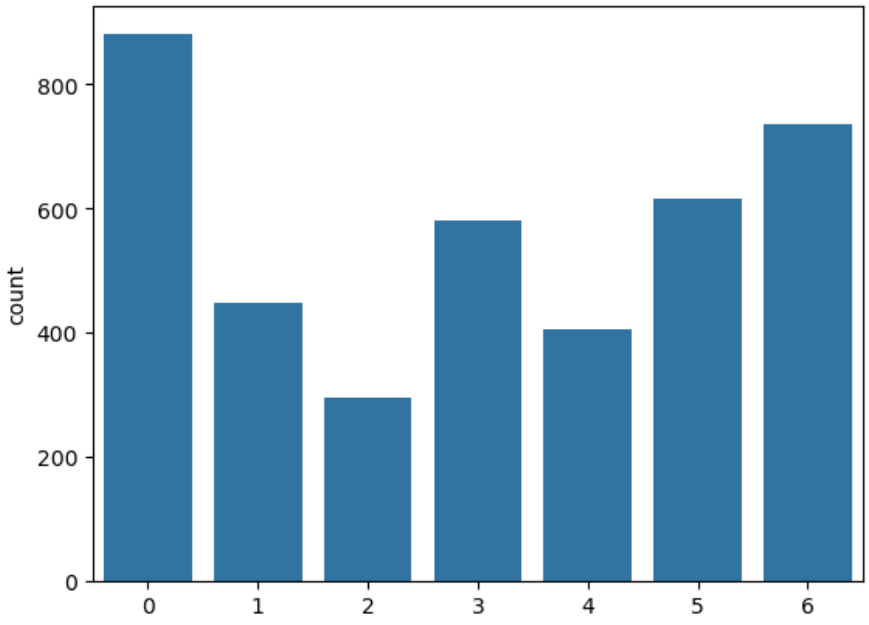
- ## Sélection des tweets représentatifs par proximité au centroïde

Description

Cette commande applique l'algorithme K-Means pour déterminer un tweet représentatif de chaque cluster en fonction de la proximité avec le centroïde du cluster. Pour chaque cluster, la commande récupère les indices des tweets, les vecteurs associés à ces tweets, et le centroïde du cluster. Ensuite, elle calcule les **distances euclidiennes** entre chaque tweet et le centroïde pour déterminer lequel est le plus proche. Le tweet avec la distance minimale est sélectionné comme le **tweet représentatif** du cluster. Cela permet de trouver un tweet qui capture le mieux les caractéristiques du cluster.

```
]: import seaborn as sns
   sns.countplot(data =results, x = 'cluster')
```

```
]: <Axes: xlabel='cluster', ylabel='count'>
```



```
vectorized_tweets =jaccard_distance_matrix
# Choose representative tweets based on proximity to centroid
representative_tweets = []

for cluster in range(7):  # Assuming 7 clusters
    print(f"Cluster {cluster}:")

    # Get the indices of the tweets in the current cluster
    cluster_indices = np.where(cluster_labels == cluster)[0]

    # Get the vectors of the tweets in this cluster
    cluster_vectors = vectorized_tweets[cluster_indices]

    # Get the centroid of the current cluster
    centroid = centroids[cluster]

    # Calculate distances from each tweet to the centroid
    distances = euclidean_distances(cluster_vectors, centroid.reshape(1, -1))

    # Find the tweet with the minimum distance to the centroid
    closest_tweet_index = cluster_indices[np.argmin(distances)]

    # Get the tweet text from the closest tweet
    representative_tweet = tweets[closest_tweet_index]

    print(f"Representative Tweet: {representative_tweet}")
    representative_tweets.append(representative_tweet)
```

- ## Importation de la bibliothèque FCM pour le clustering flou

Description

La commande importe la classe FCM de la bibliothèque **fcmeans**, qui implémente l'algorithme de clustering flou Fuzzy C-Means. Contrairement à K-Means, où chaque point appartient exclusivement à un seul cluster, le Fuzzy C-Means attribue à chaque point une probabilité d'appartenance à plusieurs clusters. Cette méthode est particulièrement utile lorsque les frontières entre les clusters ne sont pas bien définies ou lorsque les données peuvent appartenir à plusieurs groupes simultanément. L'importation de cette bibliothèque est une étape nécessaire pour appliquer cet algorithme sur les données textuelles

```
from fcmeans import FCM
```

- ## Transformation des tweets lemmatisés en une matrice TF-IDF

Description

La commande utilise le **TfidfVectorizer** de scikit-learn pour convertir les tweets lemmatisés en une matrice TF-IDF (Term Frequency-Inverse Document Frequency). Cette méthode pondère les termes selon leur fréquence dans un document individuel et leur rareté dans l'ensemble des documents. Cela permet de réduire l'influence des mots courants tout en mettant en valeur les mots pertinents. La matrice résultante est

convertie en un tableau dense avec .toarray(), rendant les données prêtes pour une analyse supplémentaire, comme le clustering ou la classification.

```
vectorizer = TfidfVectorizer()  # Binary to treat as set-like presence
binary_features = vectorizer.fit_transform(data.tweet_lemmatized)
dense_matrix = binary_features.toarray()
```

- ## Analyse des métriques FPC et PEC pour le clustering flou

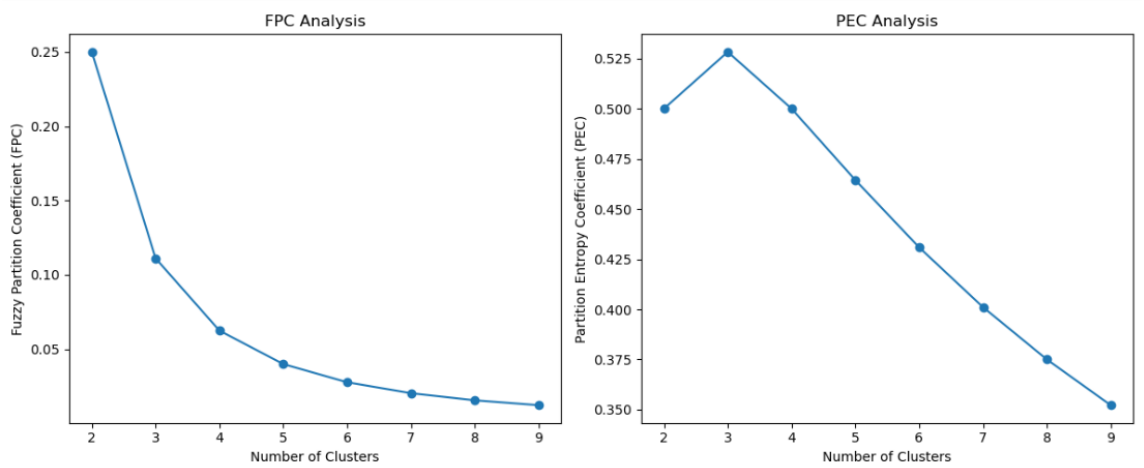
  Description

La commande utilise l'algorithme **Fuzzy C-Means (FCM)** pour effectuer un clustering flou sur la matrice TF-IDF dense des tweets. Deux métriques sont calculées pour chaque nombre de clusters (de 2 à 9) :

1. **Fuzzy Partition Coefficient (FPC)** : Indique la qualité des partitions floues en évaluant la concentration des données dans un cluster. Une valeur élevée indique une meilleure partition.
2. **Partition Entropy Coefficient (PEC)** : Mesure le degré d'incertitude ou de flou dans les partitions. Une valeur plus basse indique des partitions plus nettes.

Les valeurs de FPC et PEC sont tracées sur des graphiques pour identifier le nombre optimal de clusters, permettant de choisir un équilibre entre la cohérence intra-cluster et la séparation inter-cluster.

```
[152]: FPC = []
       PEC = []

       # Iterate over a range of clusters
       for n_clusters in range(2, 10):
           fcm = FCM(n_clusters=n_clusters)
           fcm.fit(dense_matrix)  # Replace `data` with your dataset

           FPC.append(fcm.partition_coefficient)  # Use FPC
           PEC.append(fcm.partition_entropy_coefficient)  # Use PEC

       # Plot the FPC and PEC
       plt.figure(figsize=(12, 5))

       plt.subplot(1, 2, 1)
       plt.plot(range(2, 10), FPC, marker='o')
       plt.xlabel('Number of Clusters')
       plt.ylabel('Fuzzy Partition Coefficient (FPC)')
       plt.title('FPC Analysis')

       plt.subplot(1, 2, 2)
       plt.plot(range(2, 10), PEC, marker='o')
       plt.xlabel('Number of Clusters')
       plt.ylabel('Partition Entropy Coefficient (PEC)')
       plt.title('PEC Analysis')

       plt.tight_layout()
       plt.show()
```



- ## Application de Fuzzy C-Means et attribution des clusters

Description

L'algorithme **Fuzzy C-Means (FCM)** est utilisé pour regrouper les tweets en 4 clusters avec un facteur de flou m=2m=2m=2, une valeur standard pour contrôler la flou des appartenances. Une fois le modèle ajusté avec fcm.fit, une matrice de **membership** (uuu) est obtenue. Cette matrice contient les degrés d'appartenance de chaque tweet à tous les clusters. Pour chaque tweet, le cluster avec le degré d'appartenance le plus élevé est sélectionné en utilisant np.argmax, et les étiquettes des clusters sont stockées dans cluster_labels.

```
# Step 2: Apply Fuzzy C-Means clustering
fcm = FCM(n_clusters=4, m=2)  # m=2 is a common choice for the fuzziness factor
fcm.fit(dense_matrix)

# Get the degree of membership (membership matrix)
membership = fcm.u

# Step 3: Assign each tweet to a cluster
cluster_labels = np.argmax(membership, axis=1)
```

- ## Sélection des tweets représentatifs pour chaque cluster

Description

Pour chaque cluster, cette commande identifie les tweets qui lui appartiennent en fonction des étiquettes de cluster cluster_labels. Elle utilise la matrice de **membership** pour trouver le tweet ayant le **plus haut degré d'appartenance** à ce cluster. Ce tweet est considéré comme le plus représentatif du cluster car il est le plus proche du centre du cluster dans l'espace flou défini par Fuzzy C-Means. Les tweets représentatifs sont collectés dans la liste representative_tweets et affichés pour chaque cluster.

```python
# Step 4: Choose representative tweets based on maximum membership
representative_tweets = []

for cluster in range(4):  # Assuming 4 clusters, adjust if necessary
    # Get the indices of tweets in the current cluster
    cluster_indices = np.where(cluster_labels == cluster)[0]

    # Get the membership values for the cluster
    cluster_membership = membership[cluster_indices, cluster]

    # Find the tweet with the highest membership to the cluster
    closest_tweet_index = cluster_indices[np.argmax(cluster_membership)]

    # Get the tweet text from the closest tweet
    representative_tweet = tweets[closest_tweet_index]

    print(f"Cluster {cluster}:")
    print(f"Representative Tweet: {representative_tweet}")
    representative_tweets.append(representative_tweet)
```

```
Cluster 0:
Representative Tweet: If a ref can give this as penalty then Nigeria ref is qualified to ref at this world Cup
#WorldCup #ARGvsPOL
Cluster 1:
Representative Tweet: This #worldCup is the first one to have  a female referee take charge of games.
But there is a story of a heroic Ghanaian woman MERCY TAGOE
Cluster 2:
Representative Tweet: How is this ref at a World Cup 😱😱😱
Cluster 3:
Representative Tweet: Referee🤩 #WorldCup
```