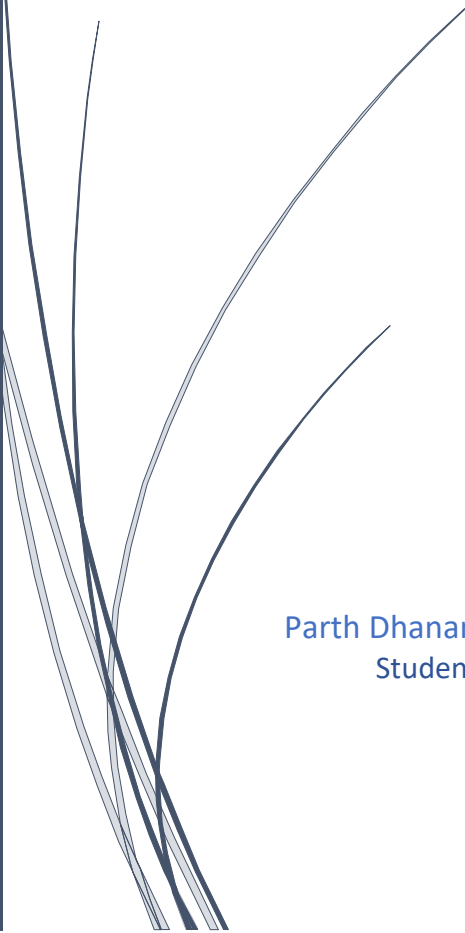


A dark blue vertical bar on the left side of the page. A blue arrow points to the right from the bar, containing the date.

1/23/2023

Twitter Sentiment Analysis

CIND820-
XJH

Several thin, dark blue curved lines that sweep upwards from the bottom left towards the center of the page.

Parth Dhananjay Desai parth.desai@torontomu.ca
Student ID: 5011453

- **Index**

- Abstract
- Literature Review, Data Description, and Project Approach
- Step by step: Twitter Sentiments Analysis in Python
- Summary
- References

Abstract

Theme & Introduction

Twitter Sentiment Analysis is a type of machine learning that aims to determine the overall sentiment of tweets or hashtags. Through natural language processing and machine learning techniques, we can extract subjective information from a dataset and classify it based on its polarity (positive, neutral, or negative). This analysis is valuable as it allows us to understand the public perception of a product or predict stock prices based on sentiment. However, sentiment analysis is a challenging problem due to the complexity of language and grammar.

For my project, I am using a probabilistic model to classify tweets on Twitter as having either a positive or negative sentiment. Twitter is a microblogging platform where users can quickly and spontaneously express their emotions through 140-character tweets. Users can also use the @ symbol and hashtags to join discussions or contact other users directly. Due to its popularity, Twitter is a valuable source of information for gauging public opinion on any topic.

Problem Definition

Sentiment analysis in the realm of micro-blogging is a relatively new field with immense potential for further research. Previous studies have focused on sentiment analysis in areas like user reviews, documents, web blogs, and general phrase-level sentiment analysis. However, sentiment analysis of tweets presents a unique challenge due to the 140-character limit that compels users to express their opinions succinctly.

Supervised learning techniques like Naive Bayes and Support Vector Machines have been the most effective sentiment classification methods to date, but they require expensive manual labelling. Some researchers have also explored unsupervised and semi-supervised approaches, but there is still room for improvement.

Researchers often compare their results to baseline performance, but there is a need for formal comparisons between different methods to determine the best features and most efficient classification techniques for specific applications. Further research is necessary to advance sentiment analysis of tweets and realize its full potential.

Data Set

Twitter sentiment analysis will typically use tweets as the data source. Tweets are short text messages of up to 280 characters that users post on the Twitter platform. These tweets can include text, hash tags, mentions, and links. Sentiment analysis of tweets aims to determine the sentiment or emotion expressed in the text of the tweet, which can be classified as positive, neutral, or negative. The tweets that are used for analysis can be collected in real-time using the Twitter API or from pre-existing datasets of tweets. The dataset can be filtered based on various criteria such as keywords, hash tags, or user accounts to focus on specific topics or communities.

Techniques and Tools

- Python
- libraries: textblob, tweepy

Literature Review, Data Description, and Project Approach

Introduction

What do you already know about the topic?

A Twitter sentiment analysis is also approached as Opinion mining is a process that helps to identify as well as divide the sentiment that is expressed on Twitter. It is a process of deciding the emotional tone of the words that have been used on Twitter. This is done through understanding the polarity of tweets as positive negative or neutral.

The analysis of this is generally accomplished using NLP (Natural Language Processing) and machine learning algorithm, which can be implied in various arena or different sectors for social media monitoring to get the best out of it can be useful in areas such as business intelligence, customer service, sport branding, as well as political science. If this data is correctly diagnosed, it can be used by many brands to figure out the aspects they lack and work on it.

What do you have to say critically about what is already known?

Based on my observations and understanding, I can confidently state that one of the most persistent challenges in sentiment analysis of tweets is accurately categorizing the different forms of emotions present.

This can often be confusing and misleading, making it difficult to discern the emotional tone and identify the true emotions and intended meaning behind a tweet, including the use of figurative language.

Mistakes in classification can lead to erroneous data, resulting in a larger failure. Additionally, emotions can vary based on context, making it challenging to create a comprehensive model for Twitter Sentiment Analysis.

These limitations need to be addressed to improve the accuracy and reliability of sentiment analysis in micro-blogging.

Has anyone else ever done anything the same?

It is highly improbable for an individual to obtain the same outcomes while scrutinizing the enormous quantity of tweets and hashtags, given the extensive amount of data accessible.

Nevertheless, numerous methods of Twitter Sentiment Analysis have investigated diverse categories of data with varying attributes and assessment techniques.

A multitude of research has been conducted regarding Twitter Sentiment Analysis, including the scrutiny of emotional support networks and the prediction of stock market trends through Sentiment Analysis.

Has anyone else done anything that is related?

It is highly unlikely to have completely identical findings since there are millions of tweets and hashtags being generated and analyzed, leading to a plethora of data.

However, multiple approaches to Twitter Sentiment Analysis may have explored various types of data, including different attributes and evaluation metrics.

It's possible that many studies have been conducted on Twitter Sentiment Analysis, which may include analyzing emotional support networks and predicting stock market trends using sentiment analysis.

Where does your work fit in with what has gone before?

Approaching the topic “Twitter sentiment analysis” our first task is to by integrate profile information, including location, age, gender, emotion, comments into the analysis model. This approach could help overcome some of the challenges faced in traditional Twitter sentiment analysis techniques, such as sarcasm and contextual variations in sentiment. To provide best example, I am

I am conducting Twitter sentiment analysis on the topic of Brexit, my work would fit into the larger body of research that has explored the public attitudes and sentiments towards Brexit and how country has taken this big step in positive, negative or in neutral way.

My research has examined the impact of Brexit on the UK economy, politics, and society, as well as the attitudes of different stakeholders such as politicians, business leaders, and citizens. Furter continuing the project I will on data related to Brexit and providing insights into the attitudes, opinions, and emotions expressed by Twitter users regarding the topic.

This could contribute to a better understanding of the public discourse surrounding Brexit and help to identify the most prevalent sentiments expressed on social media. Work will show use advanced machine learning techniques to improve the accuracy and efficiency of sentiment classification.

Furthermore, my analysis could help to identify potential gaps in previous research and suggest new directions for future investigations.

Why is your research worth doing in the light of what has already been done? ‘

Conducting Twitter sentiment analysis on Brexit my approach will be the following.

Firstly, my study will include vast research on topic which will consist of articles, blogs, trending hashtags, media resources, Newspaper etc. To get the better understanding of public review, attitudes, emotions, and sentiments.

Secondly, my approach will be to learn advanced machine learning techniques which will help me improve sentiment analysis accuracy and efficiency which I result yield more reliable results and nuanced insights into Twitter users' attitudes and opinions about Brexit.

Thirdly, my research will identify any potential gaps in previous research to bring up unseen aspects and facts about the topic which is being missed in previous research.

Finally, I will be being my best imply decision making and public discourse on Brexit. My work might help in future to improvise in decision making, raising local voice and improve communication in public.

Data Description & Methodology

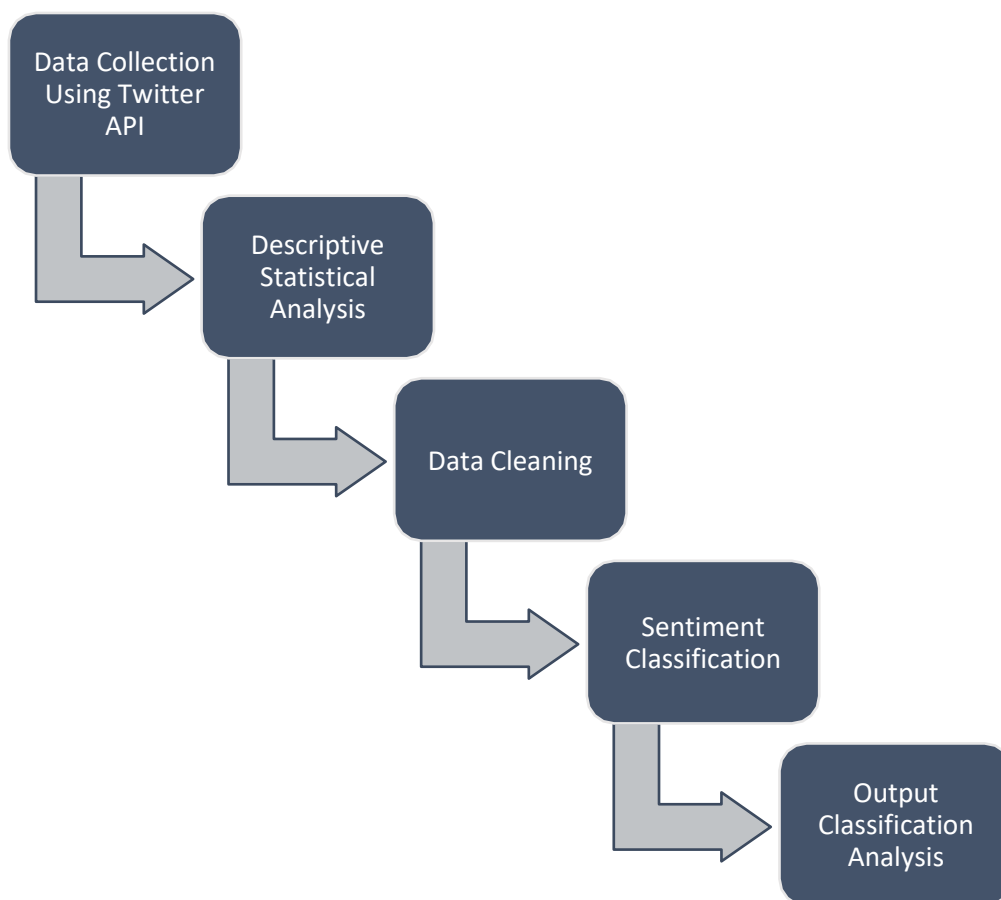
Brexit is a highly complex topic that has significant political, economic, and social implications. Sentiment analysis of Twitter data related to Brexit aims to determine the attitudes, opinions, and emotions expressed by users regarding the topic. To carry out such an analysis, a dataset is compiled that includes many tweets related to the Brexit. These tweets are collected using hashtag #Brexit.

The dataset would include relevant metadata such as user IDs, timestamps, and other information. The text data in the dataset would consist of tweets that express positive, negative, or neutral sentiments towards Brexit. To make the data suitable for analysis, pre-processing techniques such as cleaning, filtering, and tokenization would be applied to remove irrelevant or noisy data.

To classify the sentiments expressed in the tweets, various machine learning algorithms are available. In this project, I will be using Sentiment Intensity Analyzer from NLTK library.

The output of the sentiment analysis would include the percentage of positive, negative, and neutral tweets related to Brexit, as well as the most common emotions expressed in the tweets. The results of the analysis would be presented visually using charts, graphs, or other forms of data visualization, making it easier to interpret and understand the sentiment expressed on Twitter about Brexit.

In summary, sentiment analysis of Twitter data related to Brexit involves collecting many tweets, pre-processing the data, and analyzing the sentiments expressed in the tweets using natural language processing techniques. The output provides valuable insights into the attitudes, opinions, and emotions of Twitter users towards Brexit, which can be used to inform decision-making and public discourse.



Brief Descriptive Statistic

Total number of Tweets	3000
Average Word Count per Tweet	20.47
Average # count	1.17
Average @ count	1.16
Average emoji count	0.10

%Tweets having @	89.97%
%Tweets having emojis	7.10%

Step by step: Twitter Sentiments Analysis in Python

By analyzing tweets published by individuals, it is no longer difficult to learn what others think about an issue. One of the most common applications of NLP is sentiment analysis (Natural Language Processing).

In this post, I'll be using "Tweepy," a simple Python tool for interacting with the Twitter API. This study requires a Twitter developer account and sample code. The Jupyter Notebook code is available in my Github repository.

The purpose of this piece is to examine how individuals feel about Brexit, which happened on 31st January 2020.

Step 1: Import Libraries

Import the libraries that will be used in this sentiment analysis project.

```
"""  
from textblob import TextBlob  
import tweepy  
import matplotlib.pyplot as plt  
import pandas as pd  
import nltk  
import re  
import string  
from wordcloud import WordCloud, STOPWORDS  
from nltk.sentiment.vader import SentimentIntensityAnalyzer  
from sklearn.feature_extraction.text import CountVectorizer  
import emoji  
nltk.download('stopwords')  
"""
```

Tweepy may be authenticated using both OAuth 1a (application-user) and OAuth 2 (application-only). The tweepy handles authentication. AuthHandler is a class.

OAuth 2 is an authentication technique in which an application performs API calls without using the user context. As this project only requires read-only access to public information, this technique is utilized here.

To begin, I registered client application and obtain a consumer key and secret. Next, I made an AppAuthHandler object using the consumer key and secret.

Twitter Developer Account is required to have before proceeding with the authentication.

Step 2: Authentication for twitter API

```
"""
API_key = 'X55NfE8SMeg9VeMtFXc2DerWq'
API_secret = 'NxMybIb38uZSn3Zfc19FqWQ1ZeW81YbRWC5T1qCXwziNQc2Loa'
token = '2898412866-0qXfOrwGfJZpgkjOoGMfjTpV4pzL5RMadLdyj5U'
token_secret = 'GJUv3pHuD2Io3640hkgCcfuFxDGs1X3ylsbBxLCQ196Mg'
auth_ = tweepy.OAuthHandler(API_key, API_secret)
auth_.set_access_token(token, token_secret)
tw_api = tweepy.API(auth_, wait_on_rate_limit=True)
"""
```

Step 3: User Input (Hashtag and Number of Tweets)

```
"""
keyword = '#brexit'
no_of_tweets = 3000
"""
```

In this instance, the user input specifies hashtag (Brexit) and how many tweets (3000) are to be obtained and analyzed.

Step 4: Fetch Tweets

tweepy is being used to retrieve text.

```
"""
tweets = tweepy.Cursor(tw_api.search_tweets, q=keyword).items(no_of_tweets)
"""
```

Step 5: Processing of Tweets

After receiving 3000 tweets regarding "Brexit",
SentimentIntensityAnalyzer() is being used to detect the sentiment.

```
"""
positive = 0
negative = 0
neutral = 0

tweet_list = []
neutral_list = []
negative_list = []
positive_list = []

for t in tweets:
    tweet_list.append(t.text)
    analysis = TextBlob(t.text)
    score = SentimentIntensityAnalyzer().polarity_scores(t.text)
    neg = score['neg']
    neu = score['neu']
    pos = score['pos']

    if neg > pos:
        negative_list.append(t.text)
        negative += 1
    elif pos > neg:
        positive_list.append(t.text)
        positive += 1
    elif pos == neg:
        neutral_list.append(t.text)
        neutral += 1

tweet_list = pd.DataFrame(tweet_list)
"""
```

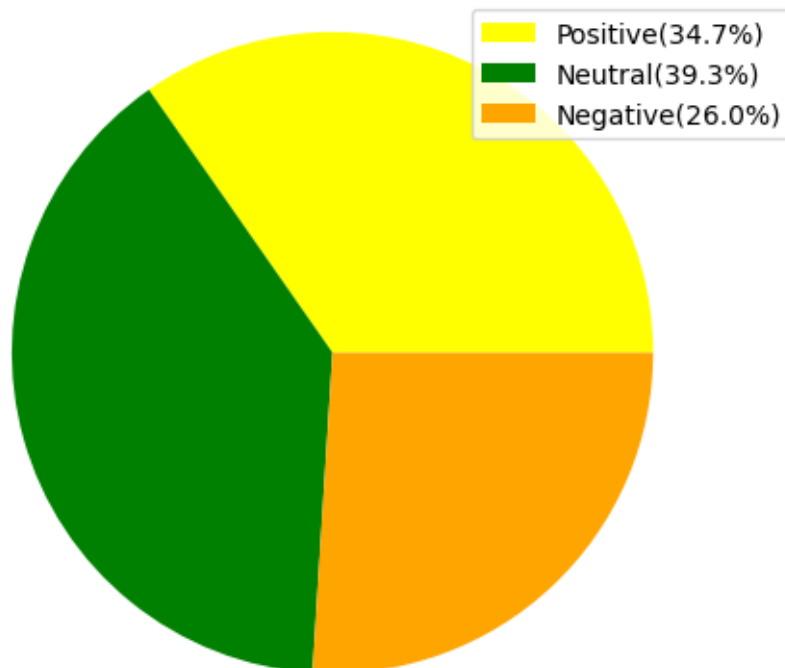
Step 6: Sentiment Analysis Before Cleaning

Sentiment analysis of tweets before and after cleaning will help us understand the importance of cleaning. Tweets generally have a lot of noise. If left unattended, results observed from that data might be misleading.

```
"""
positive = round((100 * float(positive)/float(no_of_tweets)),1)
negative = round(100 * float(negative)/float(no_of_tweets),1)
neutral = round(100 * float(neutral)/float(no_of_tweets),1)

# Pie-Chart
labels = ['Positive('+str(positive)+'%) ' ,
'Neutral('+str(neutral)+'%) ', 'Negative('+str(negative)+'%) ']
sizes = [positive, neutral, negative]
colors = ['yellow', 'green', 'orange']
patches, texts = plt.pie(sizes,colors=colors)
plt.title('Sentiment Analysis Before Cleaning : ' + keyword )
plt.legend(labels)
plt.style.use('default')
plt.axis('equal')
plt.show()
"""
```

Sentiment Analysis Before Cleaning : #brexit



Step 7: Data Analysis

Descriptive analysis is done to get familiar with the data. To get information like number of words in the tweets, '@' mentions, associated different hashtags etc.

```
"""
da = tweet_list.copy()

da['word_count'] = da[0].apply(lambda x: len(str(x).split()))

def count_regex(pattern, tweet):
    return len(re.findall(pattern, tweet))

da['_count'] = da[0].apply(lambda x: count_regex(r'@\w+', x))
da['#_count'] = da[0].apply(lambda x: count_regex(r'#\w+', x))
da['emoji_count'] = da[0].apply(lambda x: emoji.demojize(x)).apply(lambda x:
count_regex(r':[a-z_&]+:', x))
"""
```

		0	word_count	@_count	#_count	emoji_count
0	RT @SophieP25397: @Tobias_Ellwood The Port of ...		22	2	0	0
1	RT @SueScarrott: #Brexit: "Kemi Badenoch has t...		23	1	1	0
2	RT @archer_rs: #Brexit reality.		4	1	1	0
3	Oven ready #Brexit and all its benefits. \n\n...		9	0	2	0
4	RT @CornishSkipper: @Tobias_Ellwood You voted ...		22	2	0	0

Step 8: Data Cleaning

While fetching the tweets, generally lot of duplicate tweets(RT: Retweets) gets loaded. So, `drop_duplicates()` helps in removing the duplication.

```
"""  
tweet_list.drop_duplicates(inplace = True)  
"""
```

Total of 1377 unique tweets were left after removing the duplicates.

RT, Punctuations, @ mentions, links etc. are unnecessary texts which requires cleaning. With the help of Regular Expressions, these cleaning can be done.

```
"""  
twts = tweet_list.copy()  
twts['text'] = twts[0]  
  
rm_rt = lambda x: re.sub('RT @\w+: ', ' ', x)  
rt1 = lambda x: re.sub('(@[A-Za-z0-9]+) ', ' ', x)  
rt2 = lambda x: re.sub('(\w+:\\/\\/\\S+) ', ' ', x)  
  
twts['text'] = twts.text.map(rm_rt)  
twts['text'] = twts.text.map(rt1).map(rt2)  
twts['text'] = twts.text.str.lower()  
"""
```

Step 9: Processing of Tweets

Now that the cleaning process is done, with the help of `SentimentIntensityAnalyzer()`, sentiment of tweets will be analyzed again.

```
"""
no_of_tweets2 = twts.shape[0]
positive2 = 0
negative2 = 0
neutral2 = 0

#Calculating Negative, Positive and Neutral Values
twts[['polarity', 'subjectivity']] = twts['text'].apply(lambda Text:
pd.Series(TextBlob(Text).sentiment))
for index, row in twts['text'].iteritems():
    score = SentimentIntensityAnalyzer().polarity_scores(row)
    neg = score['neg']
    neu = score['neu']
    pos = score['pos']

    if neg > pos:
        twts.loc[index, 'sentiment'] = 'negative'
        negative2 += 1
    elif pos > neg:
        twts.loc[index, 'sentiment'] = 'positive'
        positive2 += 1
    else:
        twts.loc[index, 'sentiment'] = 'neutral'
        neutral2 += 1

twts_negative = twts[twts['sentiment']=='negative']
twts_positive = twts[twts['sentiment']=='positive']
twts_neutral = twts[twts['sentiment']=='neutral']
"""
```

Step 10: Sentiment Analysis After Cleaning

Sentiment analysis after the cleaning process will give us more reliable output as sentiment detection here was done excluding the inherent noise of the tweets.

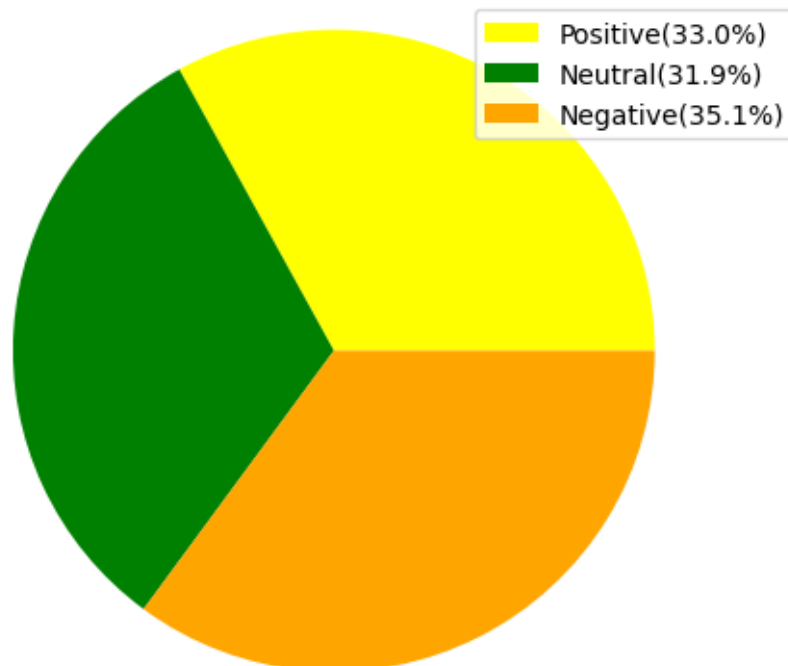
```
"""
```

```
positive2 = round((100 * float(positive2)/float(no_of_tweets2)),1)
negative2 = round(100 * float(negative2)/float(no_of_tweets2),1)
neutral2 = round(100 * float(neutral2)/float(no_of_tweets2),1)
```

```
# Pie-Chart
```

```
labels = ['Positive('+str(positive2)+'%)' ,
'Neutral('+str(neutral2)+'%)', 'Negative('+str(negative2)+'%)']
sizes = [positive2, neutral2, negative2]
colors = ['yellow', 'green', 'orange']
patches, texts = plt.pie(sizes, colors=colors)
plt.title('Sentiment Analysis After Cleaning : ' + keyword )
plt.legend(labels)
plt.style.use('default')
plt.axis('equal')
plt.show()
```

Sentiment Analysis After Cleaning : #brexit



Change in the % sentiment after cleaning is apparent. Sometimes the change is so significant that it completely reverses the results. That is why cleaning is a must while dealing with the textual data.

Step 11: Remove unimportant Words.

Apart from the noise, there might be some text which does not hold much importance or does not help in deriving any meaningful inferences. To get relevant information, these words need to be removed.

Stopwords, numerical values etc. are being removed with a list of words ('brexit', 'uk', 'europe', 'british', 'eu', 'de', 'la', 'us', 'britain'), taken while analyzing the data, which does not help in deriving any useful inferences.

```
"""
stopwords = list(nltk.corpus.stopwords.words('english'))
def clean_text(text):
    text_lc = "".join([word.lower() for word in text if word not in
string.punctuation])
    text_rc = re.sub('[0-9]+', '', text_lc)
    tokens = re.split('\W+', text_rc) # tokenization
    text = [word for word in tokens if word not in stopwords]
    text = [word for word in text if word not in
['brexit', 'uk', 'europe', 'british', 'eu', 'de', 'la', 'us', 'britain']]
    text = [word for word in text if len(word)>2]
    return text

countVectorizer = CountVectorizer(analyzer=clean_text)

# Process Neutral words
cv_neu = countVectorizer.fit_transform(twts_neutral['text'])

cv_neu_df = pd.DataFrame(cv_neu.toarray(),
columns=countVectorizer.get_feature_names_out())
cv_neu_df = pd.DataFrame(cv_neu_df.sum())

# Process Positive words
cv_pos = countVectorizer.fit_transform(twts_positive['text'])

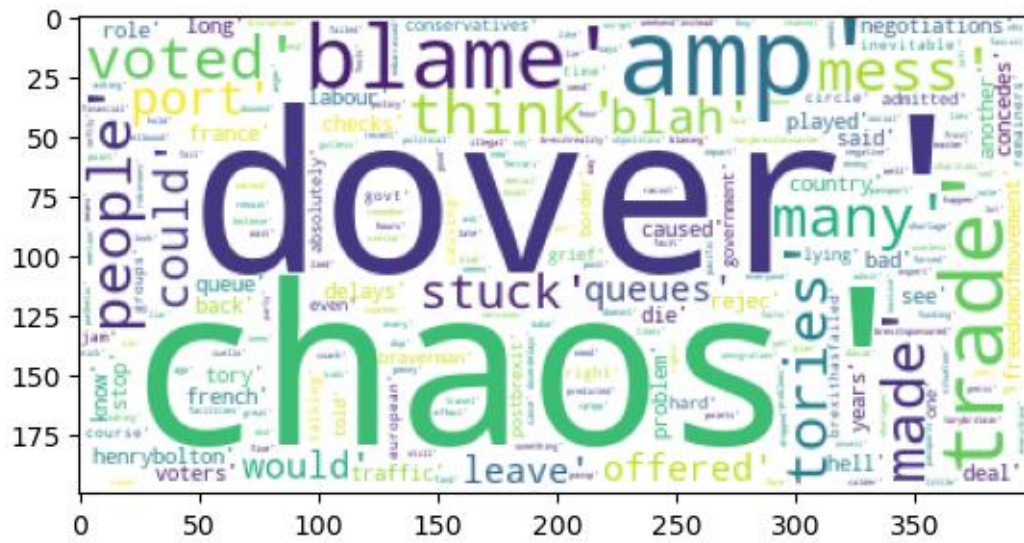
cv_pos_df = pd.DataFrame(cv_pos.toarray(),
columns=countVectorizer.get_feature_names_out())
cv_pos_df = pd.DataFrame(cv_pos_df.sum()).sort_values(0,ascending=False)

# Process Negative words
cv_neg = countVectorizer.fit_transform(twts_negative['text'])

cv_neg_df = pd.DataFrame(cv_neg.toarray(),
columns=countVectorizer.get_feature_names_out())
cv_neg_df = pd.DataFrame(cv_neg_df.sum()).sort_values(0,ascending=False)
"""
```


Twitter Sentiment Analysis

Wordcloud: Negative Sentiment



Summary

The aim of this study was to investigate how the sentiment of Twitter users towards Brexit changed over time. Data was collected using the Twitter API, and tweets containing the term "Brexit" from 2016 to 2021 were included. These tweets underwent pre-processing to remove stop words, URLs, and mentions. Natural language processing (NLP) techniques were then used to classify tweets as positive, negative, or neutral.

In the results in the sentiments are getting almost equal weights (Sentiment analysis after cleaning: Positive (33%), Neutral (31.9%) and Negative (35.1%)). Although, if we see precisely, negative sentiment has highest weightage, but the difference is not too big to make a conclusion that overall sentiment is negative. thereafter general opinion of the tweets is impartial.

Besides sentiment analysis, the study examined the most common words associated with Brexit on Twitter. The most frequent words and hashtags were political in nature, such as "Petition", "Dover", "Tories", "Trade, etc. While, "Chaos", "Amp", "Blame", "Great", "Like", "Freedom" etc., are no political words.

In conclusion, the study underscored the importance of sentiment analysis in understanding public opinion on contentious topics such as Brexit. The study's findings could be used to inform policy decisions, media coverage, and public opinion in the future.

References

- Vishal A. Kharde (2016, April 11). Sentiment analysis of twitter data: A survey of techniques.
 - International Journal of computer Applications(0975-8887).
<https://arxiv.org/ftp/arxiv/papers/1601/1601.06971.pdf>
- Faizan (2019, February 2). Twitter Sentiment Analysis. International journal of innovative science and research technology. <https://ijisrt.com/wp-content/uploads/2019/02/IJISRT19FB242.pdf>
- Adwan, Omar., et all (2020, August). Twitter Sentiment analysis Approaches. International journal of emerging technologies in learning. https://www.researchgate.net/publication/343666530_Twitter_Sentiment_Analysis_Approaches_A_Survey
- Agarwal, Apoorv, et all. Sentiment analysis of twitter data. Department of computer science, Columbia University.
<http://www.cs.columbia.edu/~julia/papers/Agarwaletal11.pdf>
- Sharma, Ankita. Ghose, Udayan (2020). Sentimental Analysis of Twitter data with respect to general elections in India. Procedia Computer Science.
<https://www.sciencedirect.com/science/article/pii/S1877050920315428>
- Alsaeedi, Abdullah, et all. (2019). A study on sentiment analysis techniques of twitter data.
- International Journal of advance computer science and applications.
https://thesai.org/Downloads/Volume10No2/Paper_48-A_Study_on_Sentiment_Analysis_Techniques.pdf
- Yaswanth, Vanama, et all (2022, May 31). Sentiments analysis using tweets.
https://assets.researchsquare.com/files/rs-1706745/v1_covered.pdf?c=1654018429
- Schwartz, Cassilde, et att (2020, May 04). A Populist paradox? How Brexit softened Anti-Immigrant Attitudes. Published online by Cambridge university Press.
<https://www.cambridge.org/core/journals/british-journal-of-political-science/article/abs/populist-paradox-how-brexit-softened-antiimmigrant-attitudes/BAEFB80FD773AF203C36B71D7518A5B7#>

- Hobolt, Sara, et al (2020, July 07). Divided by the vote: Affective Polarization in the wake of the Brexit referendum. Published online by Cambridge university Press. <https://www.cambridge.org/core/journals/british-journal-of-political-science/article/abs/divided-by-the-vote-affective-polarization-in-the-wake-of-the-brexit-referendum/2393143858C3FA161AF795269A65B900>
- Delis, Agelos, et al (2018, July 04) Electoral Spillovers in an Intertwined world: Brexit effects on the 2016 Spanish vote. Published online by Cambridge university Press. <https://www.cambridge.org/core/journals/british-journal-of-political-science/article/abs/electoral-spillovers-in-an-intertwined-world-brexit-effects-on-the-2016-spanish-vote/126DDE685EFFF565543CABA908C7B6CE>
- Goodwin, Matthew, et al (2018, February 05). For and against Brexit: A survey experiment of the impact of campaign effects on public attitudes toward EU membership. Published online by Cambridge university Press. <https://www.cambridge.org/core/journals/british-journal-of-political-science/article/for-and-against-brexit-a-survey-experiment-of-the-impact-of-campaign-effects-on-public-attitudes-toward-eu-membership/83A412AC102A3E28389F9FD66DF84AFE>
- Green, Jane, et al (2021, February 16). Who gets what: The Economy, relative Gains and Brexit. Published online by Cambridge university Press. <https://www.cambridge.org/core/journals/british-journal-of-political-science/article/who-gets-what-the-economy-relative-gains-and-brexit/092683A0C4CE85FD53FAFAEA6B7207E5>
- Murphy, Justin. Devine, Daniel (2018, July 31). Does media Coverage Drive public support for UKIP or does public support for UKIO drive media coverage? Published online by Cambridge university Press. <https://www.cambridge.org/core/journals/british-journal-of-political-science/article/does-media-coverage-drive-public-support-for-ukip-or-does-public-support-for-ukip-drive-media-coverage/81B77DDCA9B0DE26A8DF18B15158EF16>
- Bove, Vincenzo, et al (2021, April 21). Did Terrorism affect voting in the Brexit referendum?

Published online by Cambridge university Press. <https://www.cambridge.org/core/journals/british-journal-of-political-science/article/did-terrorism-affect-voting-in-the-brexit-referendum/132666AB0B1163DBD9E055E95A0C0938>

