TEXT ANALYSIS REPORT

Mini Project – Experiential Learning

To Extract Real-Time Twitter Data using Twitter API of Python and its Analysis

Topic: UFC279

Submitted on:
11th Sep 2022
Submitted to:
Dr Ajey Kumar

Submitted by:
Group 13

Student Name	PRN No
Ayush Gupta	22030242011
Sriraj Varanasi	22030242059
Saloni Jain	22030242074
Lakshmi Naveena	22030242054
Abhishek Kumar	22030242004

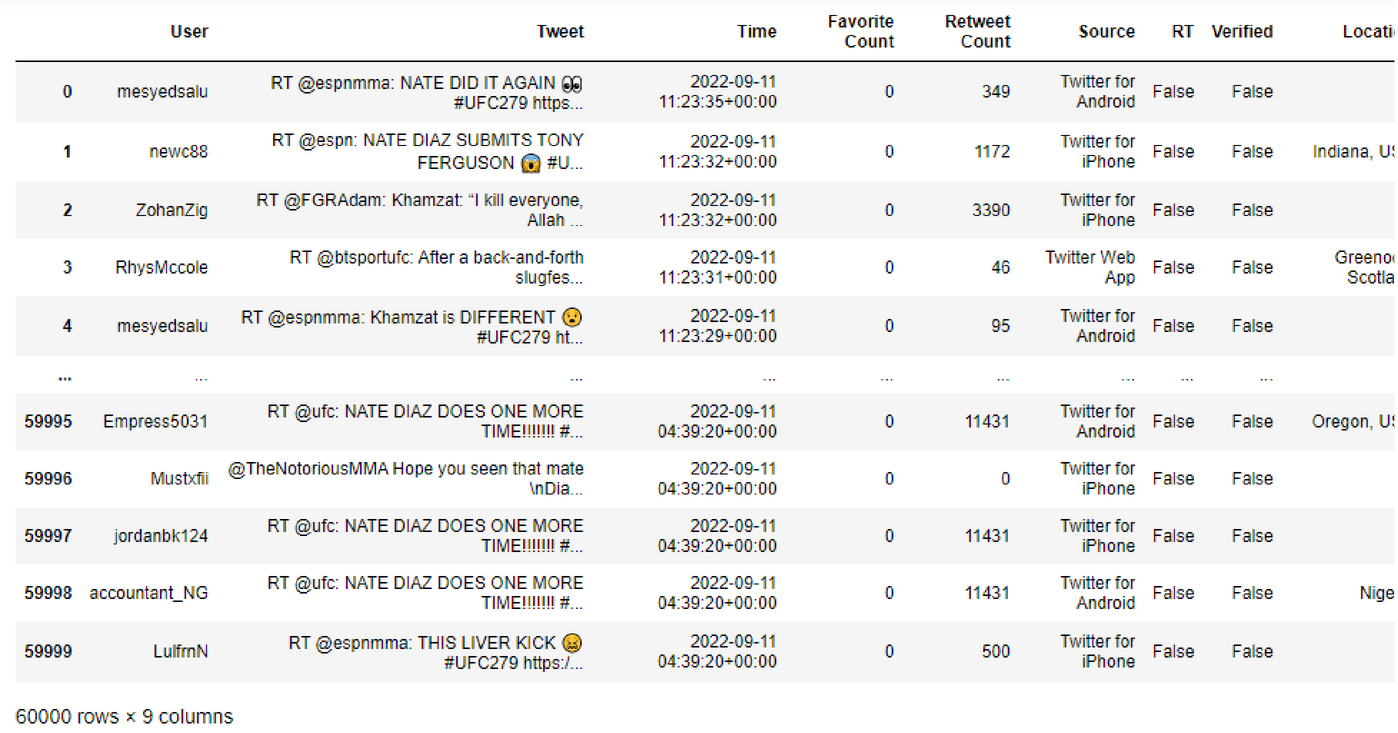
1.Introduction

UFC279 was a mixed martial arts event produced by the Ultimate Fighting championship that took place on September 10 2022.

2.Methodology and Implementation

Reading the dataset

Size of Data Downloaded: We have downloaded over 60000 tweets on the UFC279 Tweets using the API of Python.



	User	Tweet	Time	Favorite Count	Retweet Count	Source	RT	Verified	Locati
0	mesyedsalu	RT @espnmma: NATE DID IT AGAIN 👀 #UFC279 https...	2022-09-11 11:23:35+00:00	0	349	Twitter for Android	False	False	
1	newc88	RT @espn: NATE DIAZ SUBMITS TONY FERGUSON 😱 #U...	2022-09-11 11:23:32+00:00	0	1172	Twitter for iPhone	False	False	Indiana, U
2	ZohanZig	RT @FGRAdam: Khamzat: “I kill everyone, Allah ...	2022-09-11 11:23:32+00:00	0	3390	Twitter for iPhone	False	False	
3	RhysMccole	RT @btsportufc: After a back-and-forth slugfes...	2022-09-11 11:23:31+00:00	0	46	Twitter Web App	False	False	Greeno Scotla
4	mesyedsalu	RT @espnmma: Khamzat is DIFFERENT 😮 #UFC279 ht...	2022-09-11 11:23:29+00:00	0	95	Twitter for Android	False	False	
...	...	...	...	...	...	...	...	...	
59995	Empress5031	RT @ufc: NATE DIAZ DOES ONE MORE TIME!!!!!!! #...	2022-09-11 04:39:20+00:00	0	11431	Twitter for Android	False	False	Oregon, U
59996	Mustxfii	@TheNotoriousMMA Hope you seen that mate \nDia...	2022-09-11 04:39:20+00:00	0	0	Twitter for iPhone	False	False	
59997	jordanbk124	RT @ufc: NATE DIAZ DOES ONE MORE TIME!!!!!!! #...	2022-09-11 04:39:20+00:00	0	11431	Twitter for iPhone	False	False	
59998	accountant_NG	RT @ufc: NATE DIAZ DOES ONE MORE TIME!!!!!!! #...	2022-09-11 04:39:20+00:00	0	11431	Twitter for Android	False	False	Nige
59999	LulfrnN	RT @espnmma: THIS LIVER KICK 😖 #UFC279 https:/...	2022-09-11 04:39:20+00:00	0	500	Twitter for iPhone	False	False	

60000 rows × 9 columns

Importing Packages

Here we have used Python packages “Tweepy” to capture public tweets as they are generated worldwide.
Starting by importing the required packages:
1.**Tweepy**
Tweepy is an open-source Python package that gives an easy way to access the Twitter API with Python. Tweepy includes a set of classes and methods that represent Twitter's models and API endpoints and it transparently handles different implementation details such as Data encoding and decoding.

2.**Pandas**
Pandas is a product library composed for the Python programming
language for information control and investigation. Specifically, it offers information designs and tasks for controlling mathematical tables and time series.

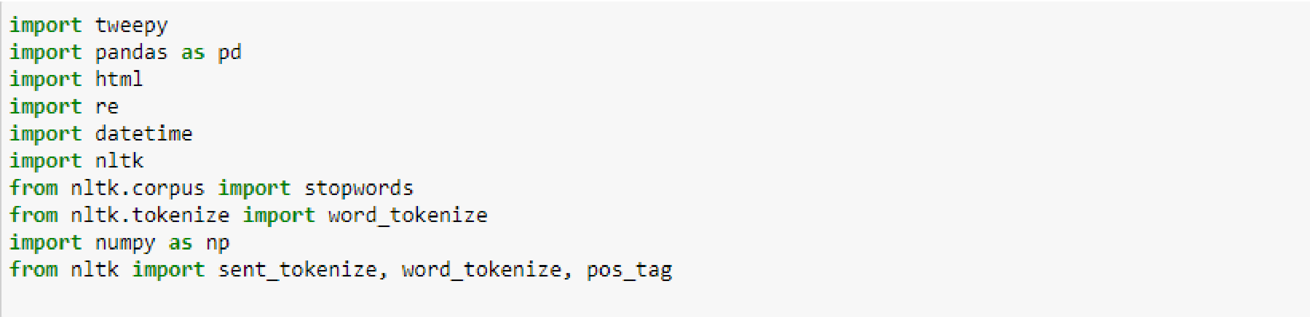
It tends to be utilized to perform information control and investigation. Pandas give amazing and simple-to-utilize information structures, just as the resources to rapidly perform procedures on these constructions.

3.**re**

re or a regular expression is an uncommon arrangement of characters that assists you with coordinating or discovering different strings or sets of strings, utilizing a particular sentence structure held in an example. Regular expressions are broadly utilized in the UNIX world.

4.**NLTK**

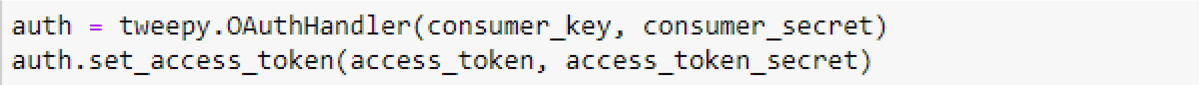
NLTK is a powerful Python package that provides a set of diverse natural languages algorithms. It is free, opensource, easy to use, large community, and well documented. NLTK consists of the most common algorithms such as tokenizing, part-of-speech tagging, stemming, sentiment analysis, topic segmentation, and named entity recognition. NLTK helps the computer to analysis, preprocess, and understand the written text.



```
import tweepy
import pandas as pd
import html
import re
import datetime
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
import numpy as np
from nltk import sent_tokenize, word_tokenize, pos_tag
```

API Keys

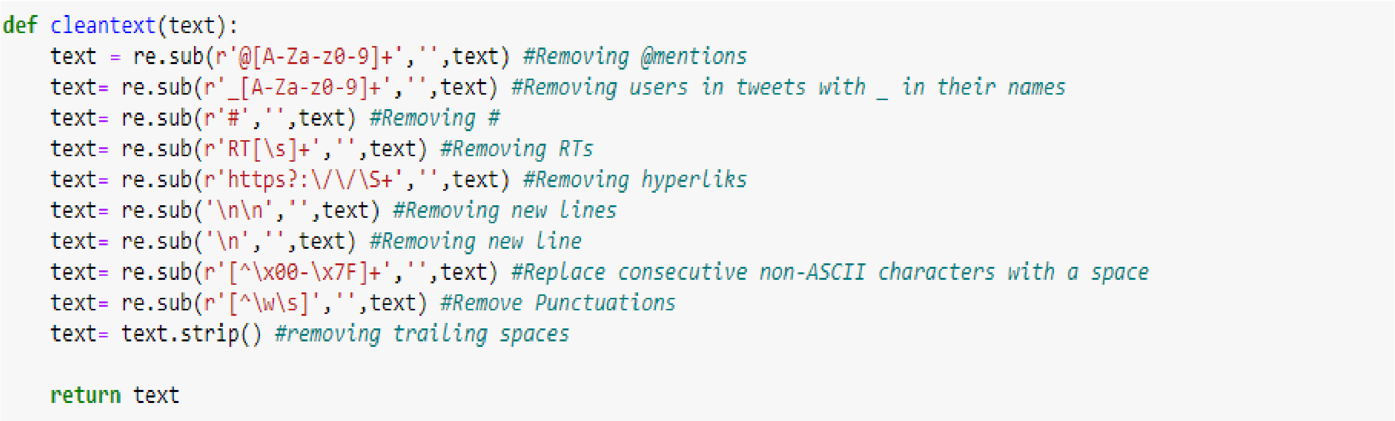
Setting the credentials (Consumer API keys and Access token & access token secret) to authenticate with the API. The keys are generated and Authenticate details are passed to Tweepy’s Authenticate handler.



```
auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_token_secret)
```

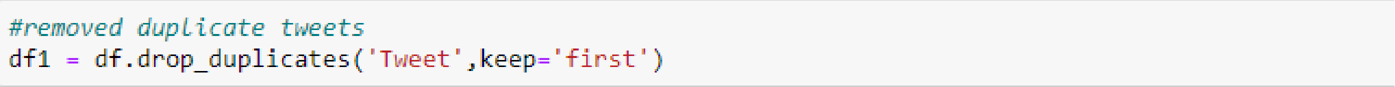
Pre-processing

Raw tweets scraped from twitter generally result in a noisy dataset. This is due to the casual nature of people’s usage of social media. Tweets have certain special characteristics such as retweets, emoticons, user mentions, etc. which have to be suitably extracted. Therefore, raw twitter data has to be normalized to create a dataset which can be easily learned by various classifiers. We have applied an extensive number of pre-processing steps to standardize the dataset and reduce its size. We first do some general pre-processing on tweets which is as follows.

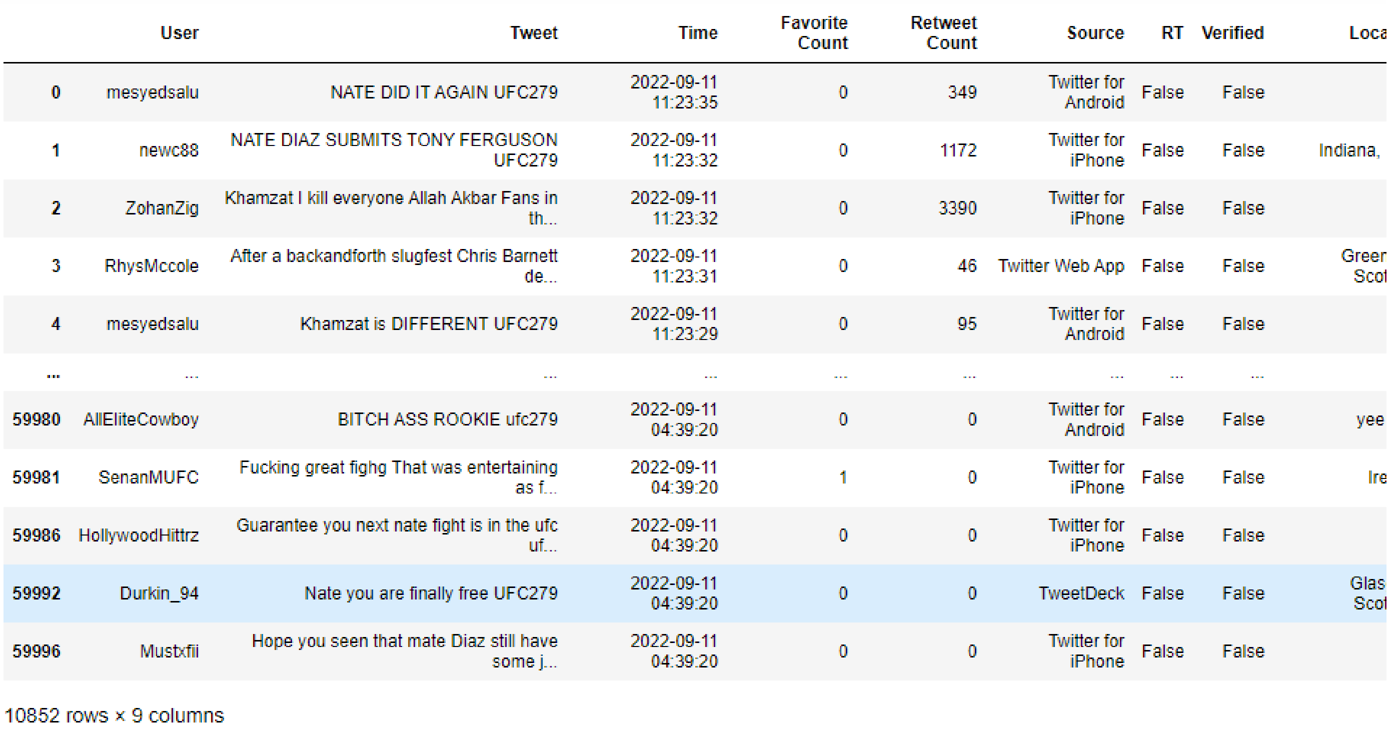


```python
def cleantext(text):
    text = re.sub(r'@[A-Za-z0-9]+','',text) #Removing @mentions
    text= re.sub(r'_[A-Za-z0-9]+','',text) #Removing users in tweets with _ in their names
    text= re.sub(r'#','',text) #Removing #
    text= re.sub(r'RT[\s]+','',text) #Removing RTs
    text= re.sub(r'https?:\/\/\S+','',text) #Removing hyperliks
    text= re.sub('\n\n','',text) #Removing new lines
    text= re.sub('\n','',text) #Removing new line
    text= re.sub(r'[^\x00-\x7F]+','',text) #Replace consecutive non-ASCII characters with a space
    text= re.sub(r'[^\w\s]','',text) #Remove Punctuations
    text= text.strip() #removing trailing spaces

    return text
```

- Remove the duplicate tweets :



```python
#removed duplicate tweets
df1 = df.drop_duplicates('Tweet',keep='first')
```



	User	Tweet	Time	Favorite Count	Retweet Count	Source	RT	Verified	Loca
0	mesyedsalu	NATE DID IT AGAIN UFC279	2022-09-11 11:23:35	0	349	Twitter for Android	False	False	
1	newc88	NATE DIAZ SUBMITS TONY FERGUSON UFC279	2022-09-11 11:23:32	0	1172	Twitter for iPhone	False	False	Indiana,
2	ZohanZig	Khamzat I kill everyone Allah Akbar Fans in th...	2022-09-11 11:23:32	0	3390	Twitter for iPhone	False	False	
3	RhysMccole	After a backandforth slugfest Chris Barnett de...	2022-09-11 11:23:31	0	46	Twitter Web App	False	False	Greer Scot
4	mesyedsalu	Khamzat is DIFFERENT UFC279	2022-09-11 11:23:29	0	95	Twitter for Android	False	False	
...	...	...	...	...	...	...	...	...	
59980	AllEliteCowboy	BITCH ASS ROOKIE ufc279	2022-09-11 04:39:20	0	0	Twitter for Android	False	False	yee
59981	SenanMUFC	Fucking great fighg That was entertaining as f...	2022-09-11 04:39:20	1	0	Twitter for iPhone	False	False	Ire
59986	HollywoodHittrz	Guarantee you next nate fight is in the ufc uf...	2022-09-11 04:39:20	0	0	Twitter for iPhone	False	False	
59992	Durkin_94	Nate you are finally free UFC279	2022-09-11 04:39:20	0	0	TweetDeck	False	False	Glas Scot
59996	Mustxfii	Hope you seen that mate Diaz still have some j...	2022-09-11 04:39:20	0	0	Twitter for iPhone	False	False	

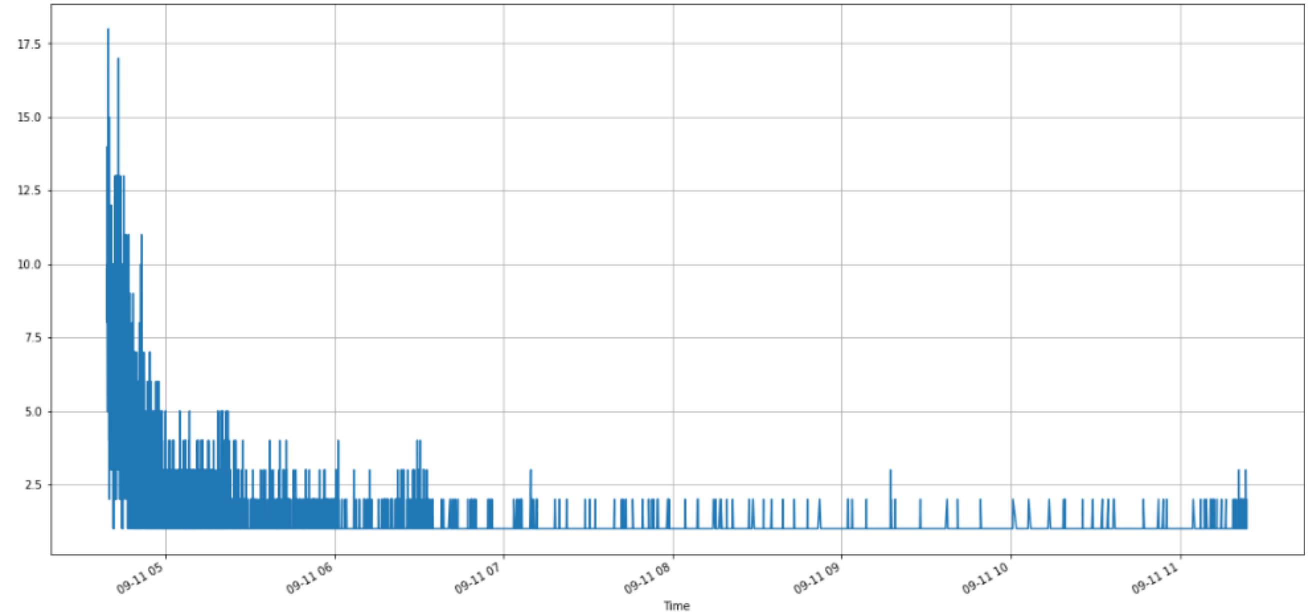
10852 rows × 9 columns

3.Descriptive Statistics

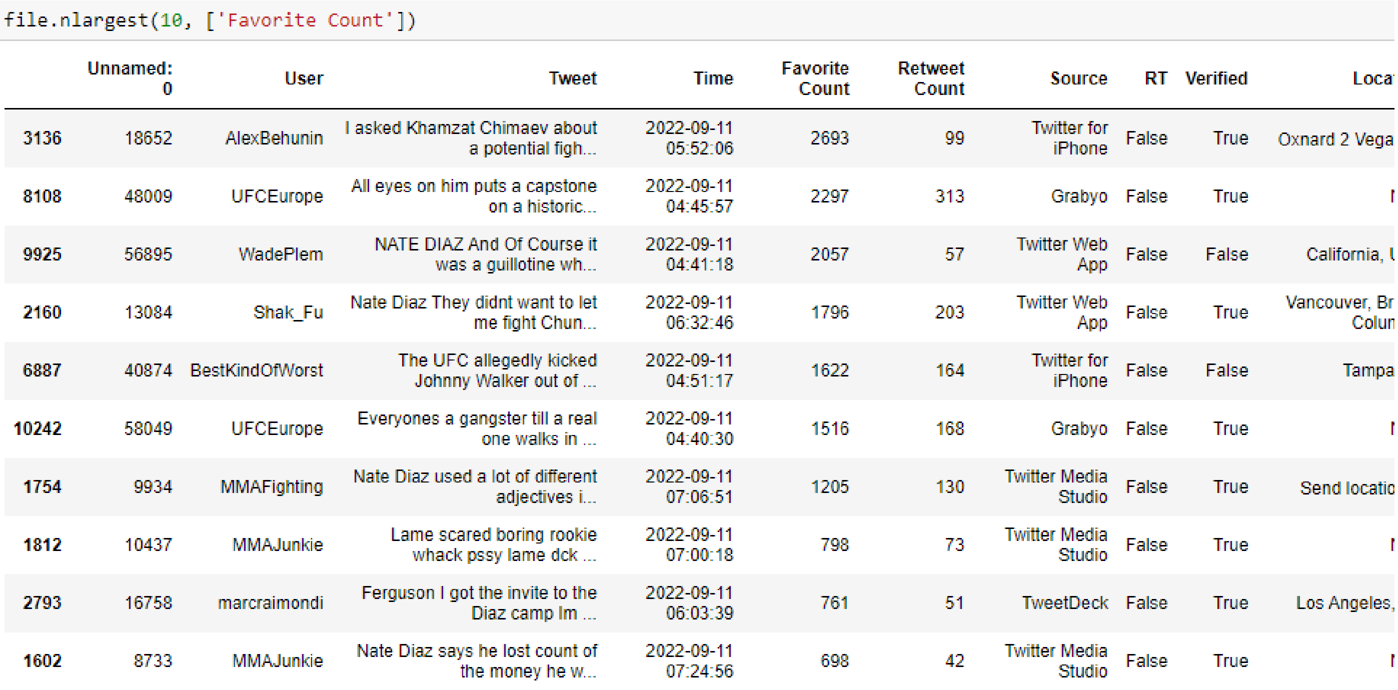
Descriptive statistics help describe and understand the features of a specific data set by giving short summaries about the sample and measures of the data.

Plotting graph on the basis of number of tweets and time

Below graph is stating that on 11th Sep, we are having the highest tweets.



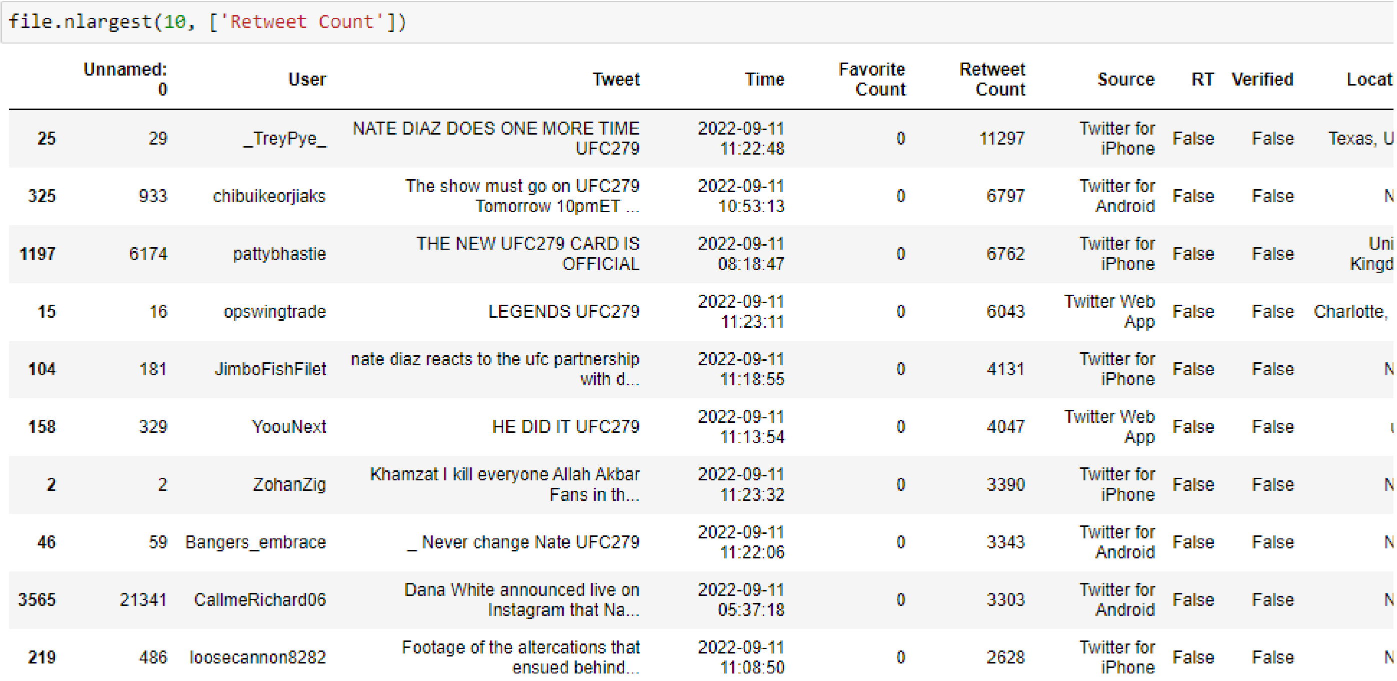
Finding out the top 10 liked tweets



```
file.nlargest(10, ['Favorite Count'])
```

	Unnamed: 0	User	Tweet	Time	Favorite Count	Retweet Count	Source	RT	Verified	Loca
3136	18652	AlexBehunin	I asked Khamzat Chimaev about a potential figh...	2022-09-11 05:52:06	2693	99	Twitter for iPhone	False	True	Oxnard 2 Vega
8108	48009	UFCEurope	All eyes on him puts a capstone on a historic...	2022-09-11 04:45:57	2297	313	Grabyo	False	True	I
9925	56895	WadePlem	NATE DIAZ And Of Course it was a guillotine wh...	2022-09-11 04:41:18	2057	57	Twitter Web App	False	False	California, U
2160	13084	Shak_Fu	Nate Diaz They didnt want to let me fight Chun...	2022-09-11 06:32:46	1796	203	Twitter Web App	False	True	Vancouver, Br Colun
6887	40874	BestKindOfWorst	The UFC allegedly kicked Johnny Walker out of ...	2022-09-11 04:51:17	1622	164	Twitter for iPhone	False	False	Tampa
10242	58049	UFCEurope	Everyones a gangster till a real one walks in ...	2022-09-11 04:40:30	1516	168	Grabyo	False	True	I
1754	9934	MMAFighting	Nate Diaz used a lot of different adjectives i...	2022-09-11 07:06:51	1205	130	Twitter Media Studio	False	True	Send locatio
1812	10437	MMAJunkie	Lame scared boring rookie whack pssy lame dck ...	2022-09-11 07:00:18	798	73	Twitter Media Studio	False	True	I
2793	16758	marcraimondi	Ferguson I got the invite to the Diaz camp Im ...	2022-09-11 06:03:39	761	51	TweetDeck	False	True	Los Angeles,
1602	8733	MMAJunkie	Nate Diaz says he lost count of the money he w...	2022-09-11 07:24:56	698	42	Twitter Media Studio	False	True	I

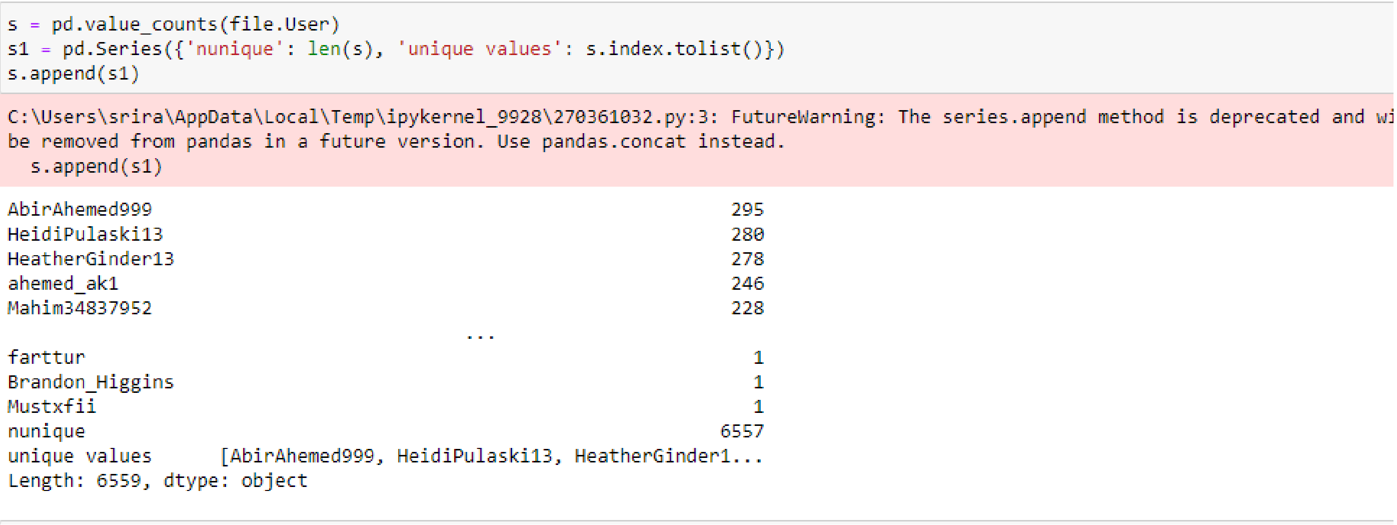
Finding out the top 10 retweeted tweets



```
file.nlargest(10, ['Retweet Count'])
```

	Unnamed: 0	User	Tweet	Time	Favorite Count	Retweet Count	Source	RT	Verified	Locat
25	29	_TreyPye_	NATE DIAZ DOES ONE MORE TIME UFC279	2022-09-11 11:22:48	0	11297	Twitter for iPhone	False	False	Texas, U
325	933	chibuikeorjiaks	The show must go on UFC279 Tomorrow 10pmET ...	2022-09-11 10:53:13	0	6797	Twitter for Android	False	False	N
1197	6174	pattybhastie	THE NEW UFC279 CARD IS OFFICIAL	2022-09-11 08:18:47	0	6762	Twitter for iPhone	False	False	Uni Kingd
15	16	opswingtrade	LEGENDS UFC279	2022-09-11 11:23:11	0	6043	Twitter Web App	False	False	Charlotte,
104	181	JimboFishFilet	nate diaz reacts to the ufc partnership with d...	2022-09-11 11:18:55	0	4131	Twitter for iPhone	False	False	N
158	329	YouuNext	HE DID IT UFC279	2022-09-11 11:13:54	0	4047	Twitter Web App	False	False	l
2	2	ZohanZig	Khamzat I kill everyone Allah Akbar Fans in th...	2022-09-11 11:23:32	0	3390	Twitter for iPhone	False	False	N
46	59	Bangers_embrace	_ Never change Nate UFC279	2022-09-11 11:22:06	0	3343	Twitter for Android	False	False	N
3565	21341	CallmeRichard06	Dana White announced live on Instagram that Na...	2022-09-11 05:37:18	0	3303	Twitter for Android	False	False	N
219	486	loosecannon8282	Footage of the altercations that ensued behind...	2022-09-11 11:08:50	0	2628	Twitter for iPhone	False	False	N

Finding out the Unique Users



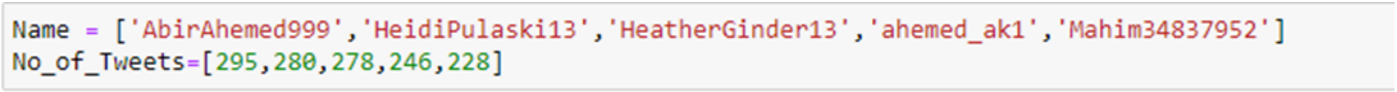
```
s = pd.value_counts(file.User)
s1 = pd.Series({'nunique': len(s), 'unique values': s.index.tolist()})
s.append(s1)
```

```
C:\Users\srira\AppData\Local\Temp\ipykernel_9928\270361032.py:3: FutureWarning: The series.append method is deprecated and wi
be removed from pandas in a future version. Use pandas.concat instead.
  s.append(s1)
```

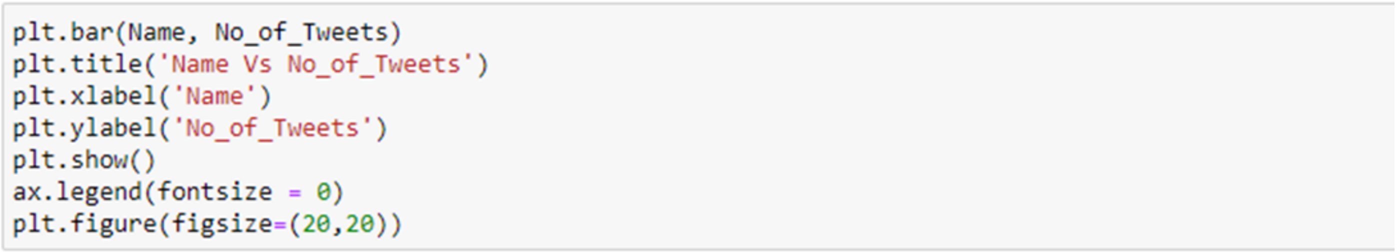
```
AbirAhemed999                                                  295
HeidiPulaski13                                                 280
HeatherGinder13                                                278
ahemed_ak1                                                     246
Mahim34837952                                                  228
                                        ...
farttur                                                          1
Brandon_Higgins                                                  1
Mustxfii                                                         1
nunique                                                       6557
unique values       [AbirAhemed999, HeidiPulaski13, HeatherGinder1...
Length: 6559, dtype: object
```

Bar Graph

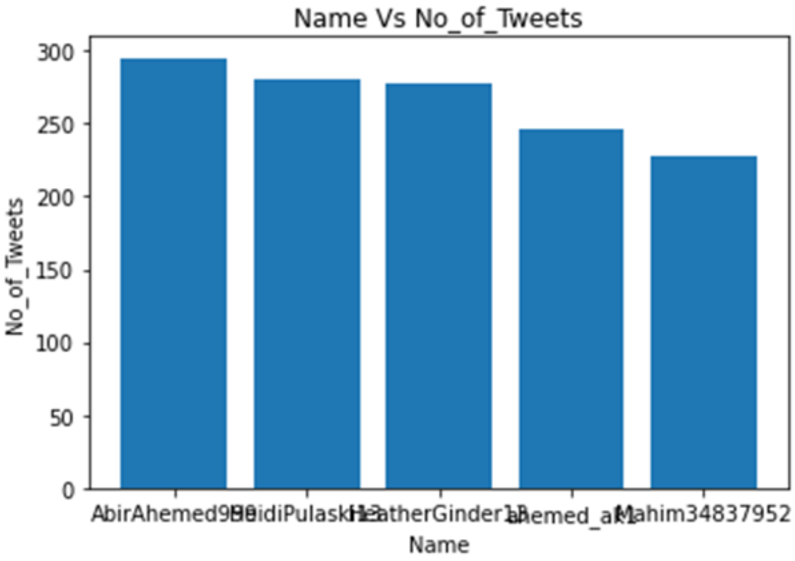
First, we find the top Users having maximum number of tweets .Then we are doing analysis on the top 5 users and their total number of tweets.



```
Name = ['AbirAhemed999','HeidiPulaski13','HeatherGinder13','ahemed_ak1','Mahim34837952']
No_of_Tweets=[295,280,278,246,228]
```



```
plt.bar(Name, No_of_Tweets)
plt.title('Name Vs No_of_Tweets')
plt.xlabel('Name')
plt.ylabel('No_of_Tweets')
plt.show()
ax.legend(fontsize = 0)
plt.figure(figsize=(20,20))
```



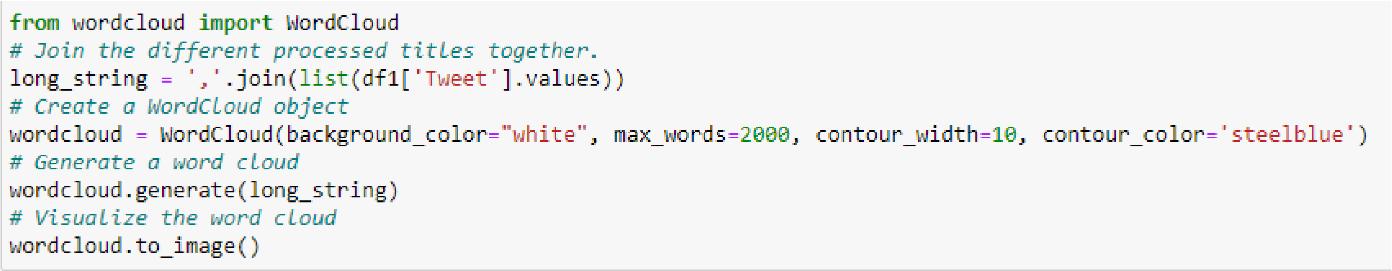
Wordcloud

Word Cloud is an information perception procedure used for addressing text information in which the size of each word shows its recurrence or significance. Critical text-based information focuses can be featured utilizing a word cloud. Word mists are generally utilized for breaking down information from

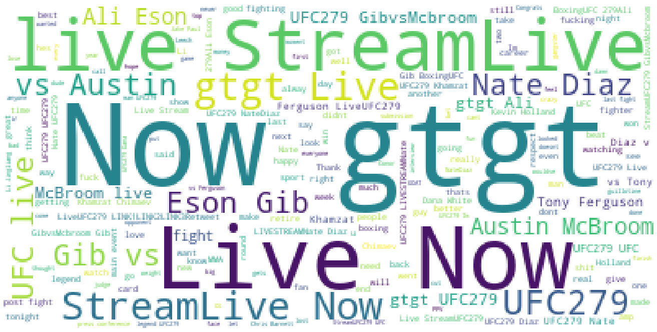
interpersonal organization sites.

For creating word cloud in Python, modules required are – matplotlib, pandas and word cloud.

A word cloud of all the tweets posted during the event was created to better understand the most trending keywords. It can be noticed that 'Now', 'Live', 'gtgt' and 'Stream' were some of the most tweeted words in the various tweets.



```
from wordcloud import WordCloud
# Join the different processed titles together.
long_string = ','.join(list(df1['Tweet'].values))
# Create a WordCloud object
wordcloud = WordCloud(background_color="white", max_words=2000, contour_width=10, contour_color='steelblue')
# Generate a word cloud
wordcloud.generate(long_string)
# Visualize the word cloud
wordcloud.to_image()
```



4.Sentiment Analysis

Sentiment analysis (also known as opinion mining or emotion AI) is the use of natural language processing, text analysis, computational linguistics, and biometrics to systematically identify, extract, quantify, and study affective states and subjective information

A subjectivity and polarity of the tweets were created in the Data Frame to conduct the sentiment analysis. To do so, 2 functions: one to get tweets classified as Subjectivity (how subjective or opinionated a tweet is on a score of 0-1 where 0 is fact, and a score of 1 is very much an opinion), and another to get tweets classified as Polarity (how positive or negative a tweet is on a score of (-1,1) where -1 is the highest negative score, and a score of +1 is the highest positive score)

were defined. The data was separated into two columns:

❖ Subjectivity

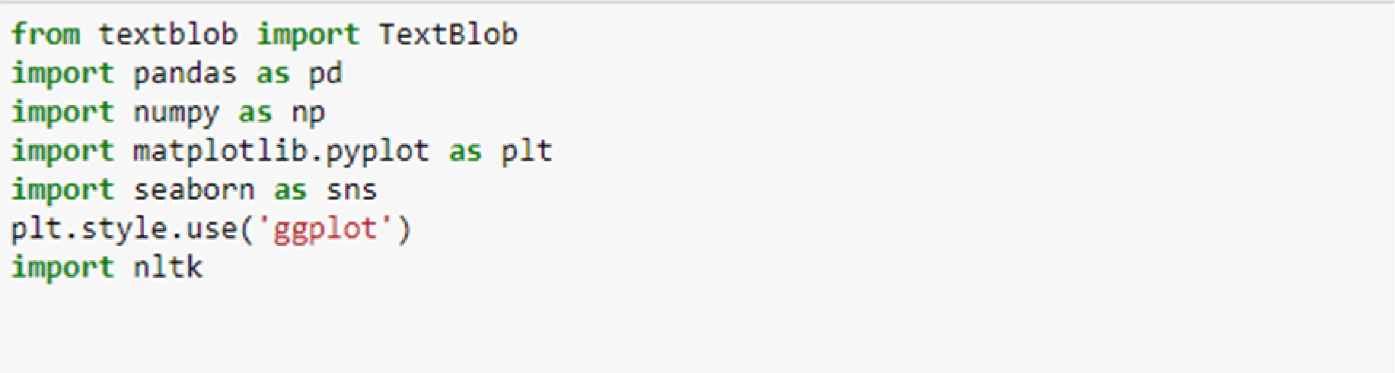
❖ Polarity

Importing Packages

Textblob: It provides a simple API for diving into common natural language processing (NLP) tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and more.

Matplotlib: Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible.

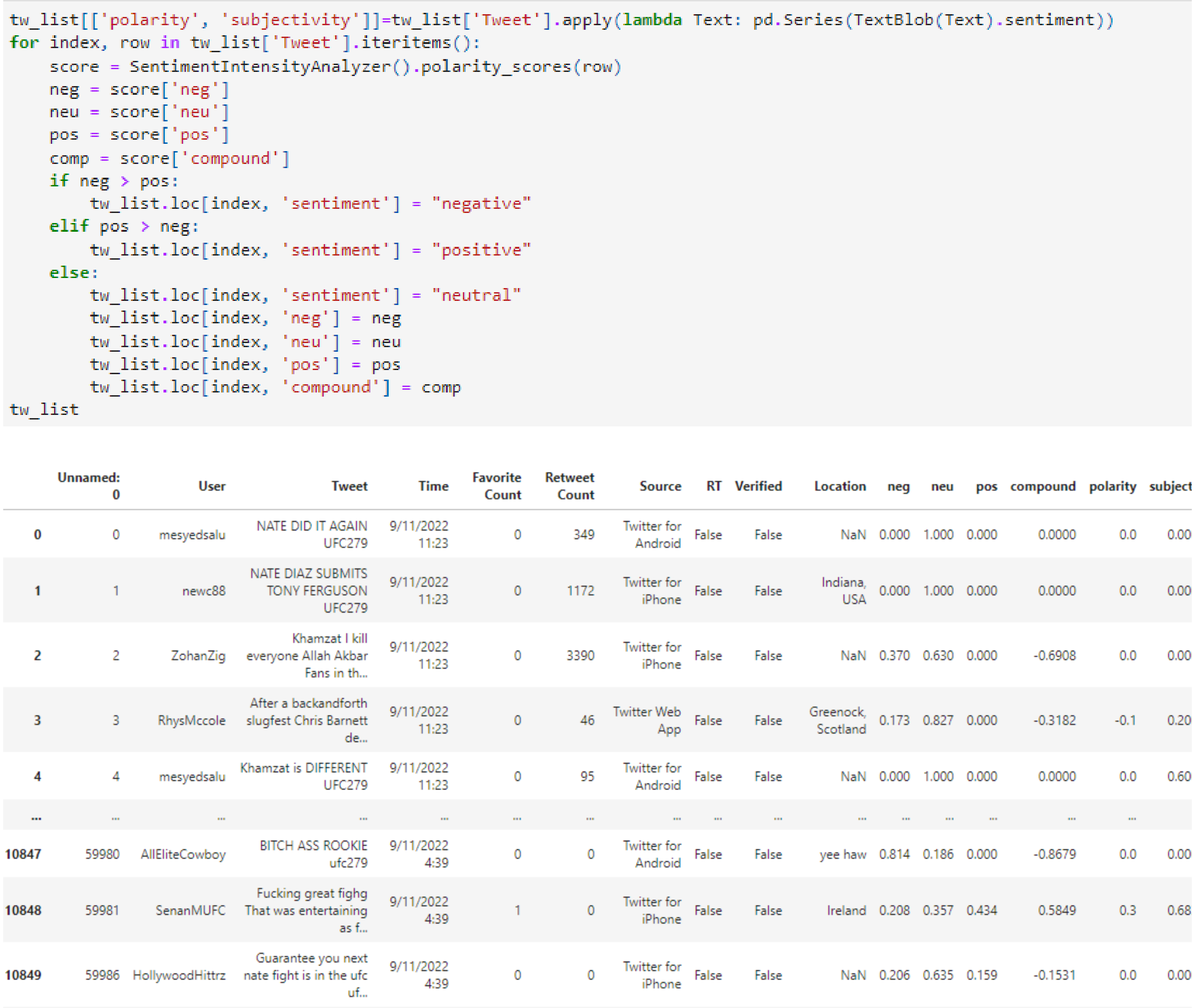
Seaborn: Seaborn is an amazing visualization library for statistical graphics plotting in Python. It provides beautiful default styles and color palettes to make statistical plots more attractive.



```python
from textblob import TextBlob
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
plt.style.use('ggplot')
import nltk
```

Finding out the negative, positive and neutral score

In this, we are measuring the sentiment score of the tweets.

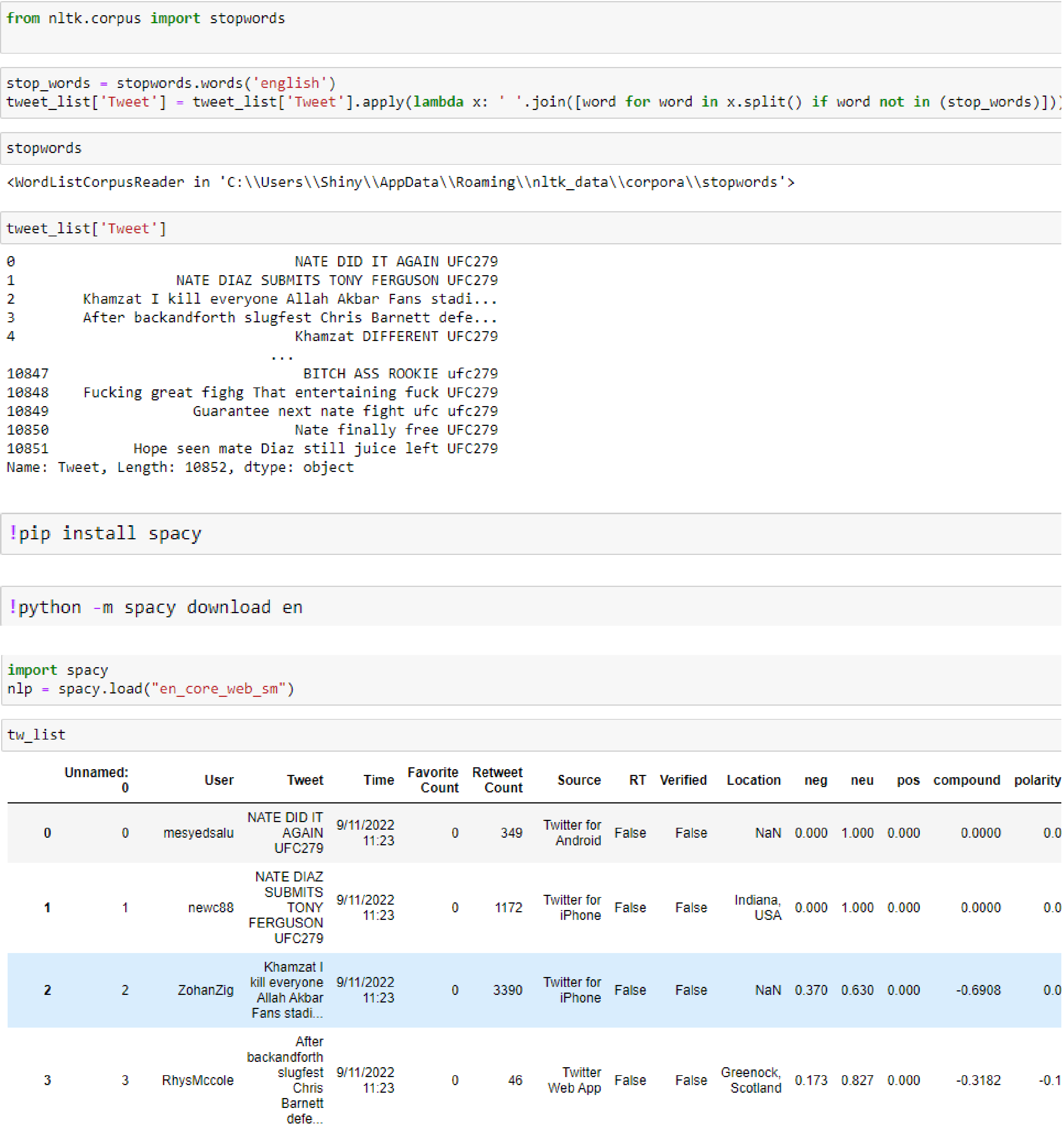


```python
tw_list[['polarity', 'subjectivity']]=tw_list['Tweet'].apply(lambda Text: pd.Series(TextBlob(Text).sentiment))
for index, row in tw_list['Tweet'].iteritems():
    score = SentimentIntensityAnalyzer().polarity_scores(row)
    neg = score['neg']
    neu = score['neu']
    pos = score['pos']
    comp = score['compound']
    if neg > pos:
        tw_list.loc[index, 'sentiment'] = "negative"
    elif pos > neg:
        tw_list.loc[index, 'sentiment'] = "positive"
    else:
        tw_list.loc[index, 'sentiment'] = "neutral"
        tw_list.loc[index, 'neg'] = neg
        tw_list.loc[index, 'neu'] = neu
        tw_list.loc[index, 'pos'] = pos
        tw_list.loc[index, 'compound'] = comp
tw_list
```

	Unnamed: 0	User	Tweet	Time	Favorite Count	Retweet Count	Source	RT	Verified	Location	neg	neu	pos	compound	polarity	subject
0	0	mesyedsalu	NATE DID IT AGAIN UFC279	9/11/2022 11:23	0	349	Twitter for Android	False	False	NaN	0.000	1.000	0.000	0.0000	0.0	0.00
1	1	newc88	NATE DIAZ SUBMITS TONY FERGUSON UFC279	9/11/2022 11:23	0	1172	Twitter for iPhone	False	False	Indiana, USA	0.000	1.000	0.000	0.0000	0.0	0.00
2	2	ZohanZig	Khamzat I kill everyone Allah Akbar Fans in th...	9/11/2022 11:23	0	3390	Twitter for iPhone	False	False	NaN	0.370	0.630	0.000	-0.6908	0.0	0.00
3	3	RhysMccole	After a backandforth slugfest Chris Barnett de...	9/11/2022 11:23	0	46	Twitter Web App	False	False	Greenock, Scotland	0.173	0.827	0.000	-0.3182	-0.1	0.20
4	4	mesyedsalu	Khamzat is DIFFERENT UFC279	9/11/2022 11:23	0	95	Twitter for Android	False	False	NaN	0.000	1.000	0.000	0.0000	0.0	0.60
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
10847	59980	AllEliteCowboy	BITCH ASS ROOKIE ufc279	9/11/2022 4:39	0	0	Twitter for Android	False	False	yee haw	0.814	0.186	0.000	-0.8679	0.0	0.00
10848	59981	SenanMUFC	Fucking great fighg That was entertaining as f...	9/11/2022 4:39	1	0	Twitter for iPhone	False	False	Ireland	0.208	0.357	0.434	0.5849	0.3	0.68
10849	59986	HollywoodHittrz	Guarantee you next nate fight is in the ufc uf...	9/11/2022 4:39	0	0	Twitter for iPhone	False	False	NaN	0.206	0.635	0.159	-0.1531	0.0	0.00

5.Aspect Based Sentiment Analysis

Aspect-based sentiment analysis is the task of identifying fine-grained opinion polarity towards a specific aspect associated with a given target.



```
from nltk.corpus import stopwords
```

```
stop_words = stopwords.words('english')
tweet_list['Tweet'] = tweet_list['Tweet'].apply(lambda x: ' '.join([word for word in x.split() if word not in (stop_words)]))
```

```
stopwords
```

```
<WordListCorpusReader in 'C:\\Users\\Shiny\\AppData\\Roaming\\nltk_data\\corpora\\stopwords'>
```

```
tweet_list['Tweet']
```

```
0                            NATE DID IT AGAIN UFC279
1                   NATE DIAZ SUBMITS TONY FERGUSON UFC279
2         Khamzat I kill everyone Allah Akbar Fans stadi...
3         After backandforth slugfest Chris Barnett defe...
4                            Khamzat DIFFERENT UFC279
                                   ...
10847                          BITCH ASS ROOKIE ufc279
10848    Fucking great fighg That entertaining fuck UFC279
10849               Guarantee next nate fight ufc ufc279
10850                         Nate finally free UFC279
10851          Hope seen mate Diaz still juice left UFC279
Name: Tweet, Length: 10852, dtype: object
```

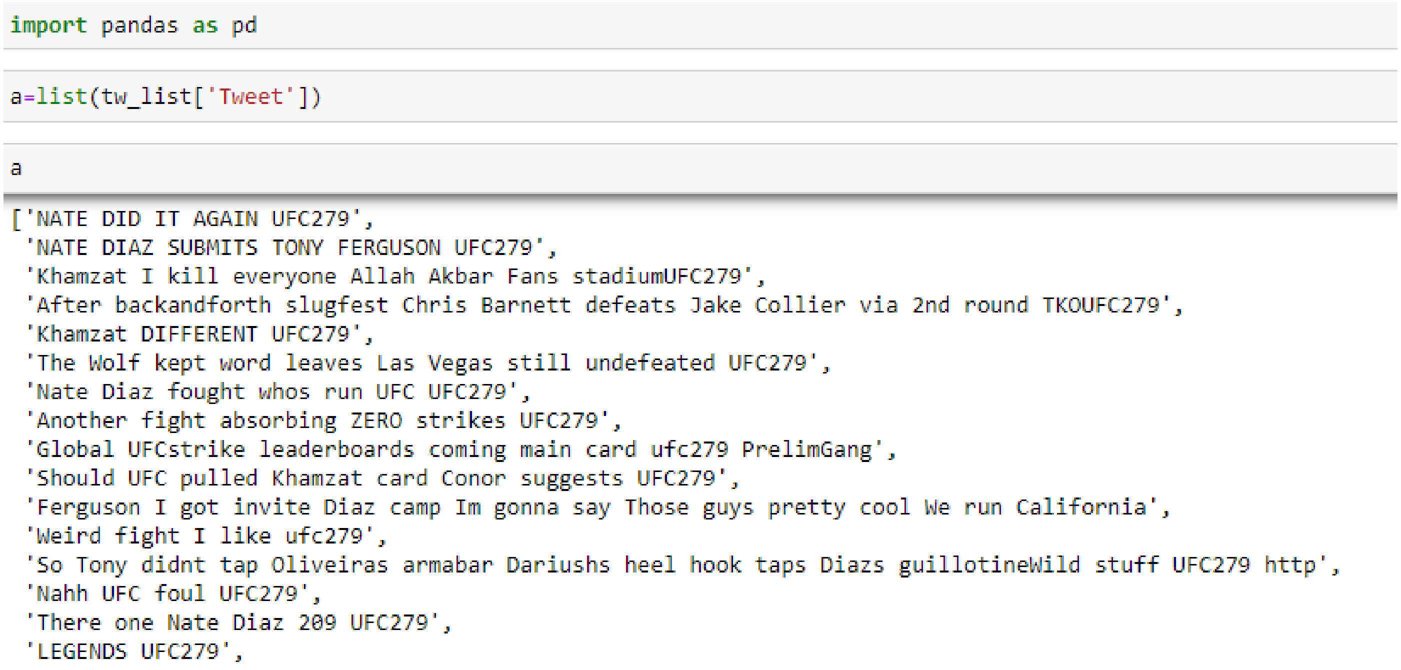
```
!pip install spacy
```

```
!python -m spacy download en
```

```
import spacy
nlp = spacy.load("en_core_web_sm")
```

```
tw_list
```

	Unnamed: 0	User	Tweet	Time	Favorite Count	Retweet Count	Source	RT	Verified	Location	neg	neu	pos	compound	polarity
0	0	mesyedsalu	NATE DID IT AGAIN UFC279	9/11/2022 11:23	0	349	Twitter for Android	False	False	NaN	0.000	1.000	0.000	0.0000	0.0
1	1	newc88	NATE DIAZ SUBMITS TONY FERGUSON UFC279	9/11/2022 11:23	0	1172	Twitter for iPhone	False	False	Indiana, USA	0.000	1.000	0.000	0.0000	0.0
2	2	ZohanZig	Khamzat I kill everyone Allah Akbar Fans stadi...	9/11/2022 11:23	0	3390	Twitter for iPhone	False	False	NaN	0.370	0.630	0.000	-0.6908	0.0
3	3	RhysMccole	After backandforth slugfest Chris Barnett defe...	9/11/2022 11:23	0	46	Twitter Web App	False	False	Greenock, Scotland	0.173	0.827	0.000	-0.3182	-0.1

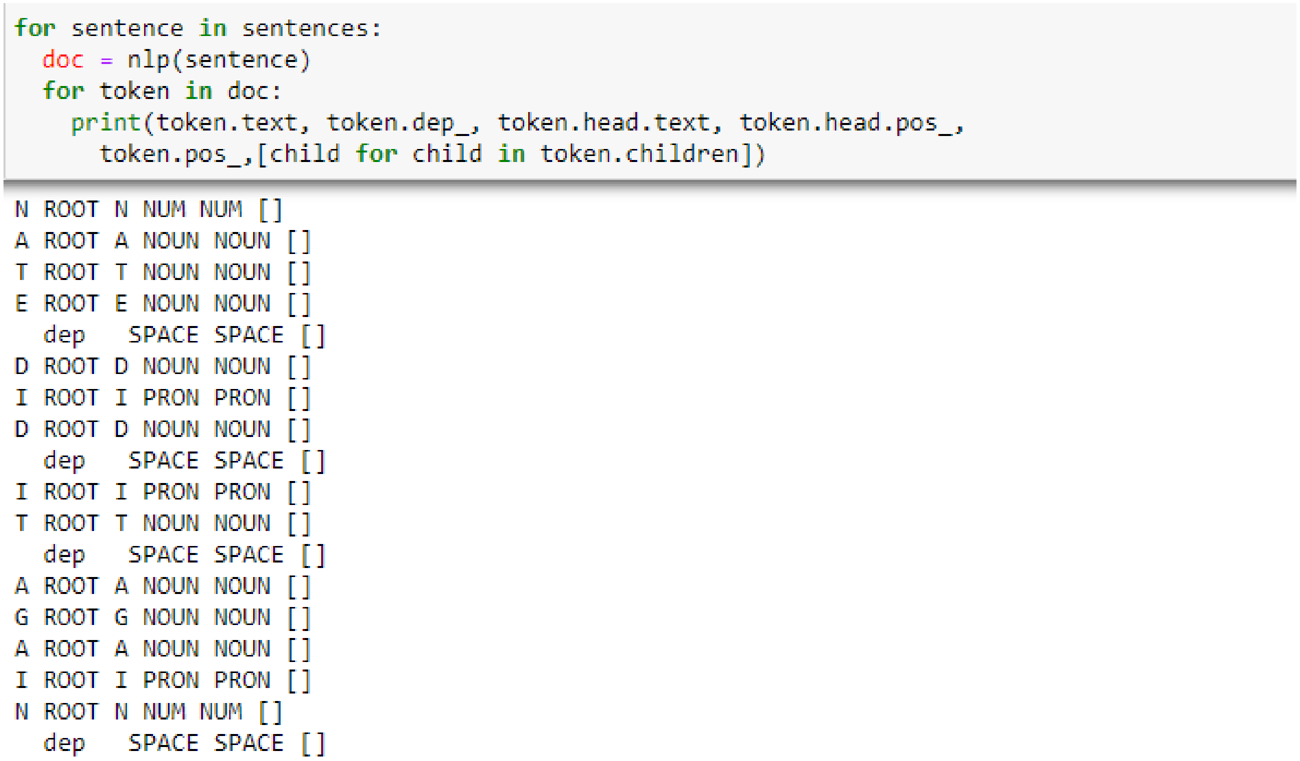


```
import pandas as pd
```

```
a=list(tw_list['Tweet'])
```

```
a
```

```
['NATE DID IT AGAIN UFC279',
 'NATE DIAZ SUBMITS TONY FERGUSON UFC279',
 'Khamzat I kill everyone Allah Akbar Fans stadiumUFC279',
 'After backandforth slugfest Chris Barnett defeats Jake Collier via 2nd round TKOUFC279',
 'Khamzat DIFFERENT UFC279',
 'The Wolf kept word leaves Las Vegas still undefeated UFC279',
 'Nate Diaz fought whos run UFC UFC279',
 'Another fight absorbing ZERO strikes UFC279',
 'Global UFCstrike leaderboards coming main card ufc279 PrelimGang',
 'Should UFC pulled Khamzat card Conor suggests UFC279',
 'Ferguson I got invite Diaz camp Im gonna say Those guys pretty cool We run California',
 'Weird fight I like ufc279',
 'So Tony didnt tap Oliveiras armabar Dariushs heel hook taps Diazs guillotineWild stuff UFC279 http',
 'Nahh UFC foul UFC279',
 'There one Nate Diaz 209 UFC279',
 'LEGENDS UFC279',
```



```
for sentence in sentences:
  doc = nlp(sentence)
  for token in doc:
    print(token.text, token.dep_, token.head.text, token.head.pos_,
      token.pos_,[child for child in token.children])
```

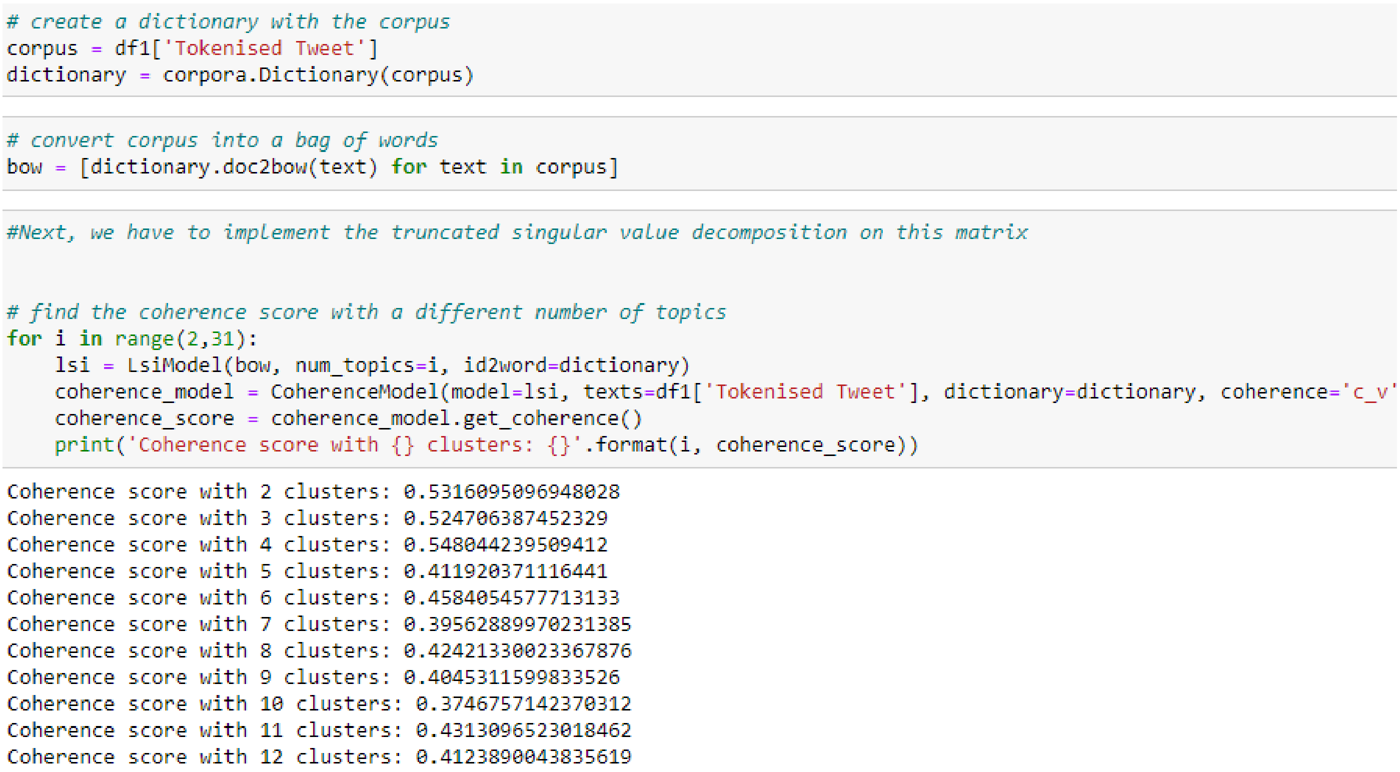
```
N ROOT N NUM NUM []
A ROOT A NOUN NOUN []
T ROOT T NOUN NOUN []
E ROOT E NOUN NOUN []
  dep   SPACE SPACE []
D ROOT D NOUN NOUN []
I ROOT I PRON PRON []
D ROOT D NOUN NOUN []
  dep   SPACE SPACE []
I ROOT I PRON PRON []
T ROOT T NOUN NOUN []
  dep   SPACE SPACE []
A ROOT A NOUN NOUN []
G ROOT G NOUN NOUN []
A ROOT A NOUN NOUN []
I ROOT I PRON PRON []
N ROOT N NUM NUM []
  dep   SPACE SPACE []
```

6.Topic Modelling

Topic modelling is a part of machine learning that essentially helps us discover topics from a collection.

Latent Semantic Analysis

In Latent semantic analysis,we have created a dictionary of tokenized tweet and then we implement the truncated singular value decomposition to find out the coherence score with different number of topics.

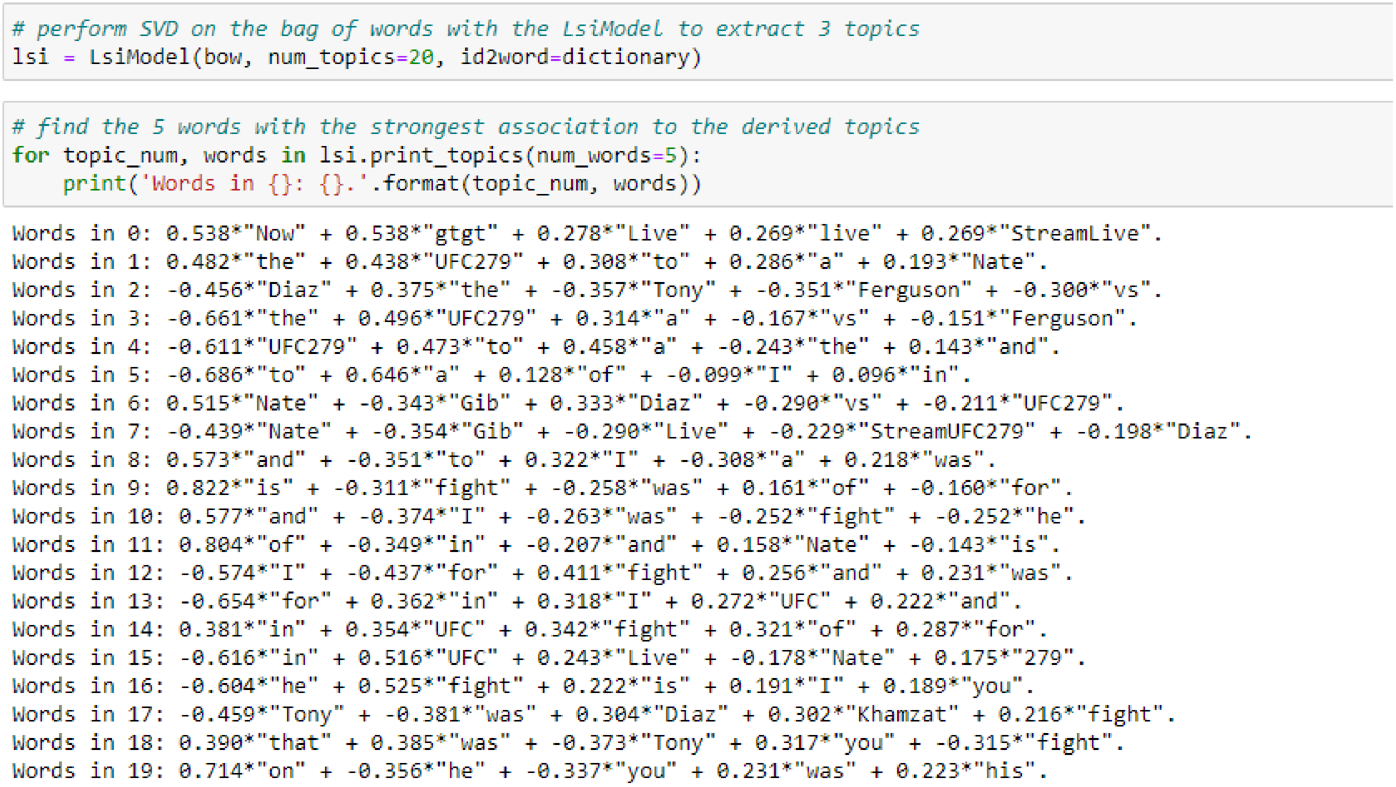


```
# create a dictionary with the corpus
corpus = df1['Tokenised Tweet']
dictionary = corpora.Dictionary(corpus)
```

```
# convert corpus into a bag of words
bow = [dictionary.doc2bow(text) for text in corpus]
```

```
#Next, we have to implement the truncated singular value decomposition on this matrix


# find the coherence score with a different number of topics
for i in range(2,31):
    lsi = LsiModel(bow, num_topics=i, id2word=dictionary)
    coherence_model = CoherenceModel(model=lsi, texts=df1['Tokenised Tweet'], dictionary=dictionary, coherence='c_v'
    coherence_score = coherence_model.get_coherence()
    print('Coherence score with {} clusters: {}'.format(i, coherence_score))
```

```
Coherence score with 2 clusters: 0.5316095096948028
Coherence score with 3 clusters: 0.524706387452329
Coherence score with 4 clusters: 0.548044239509412
Coherence score with 5 clusters: 0.411920371116441
Coherence score with 6 clusters: 0.4584054577713133
Coherence score with 7 clusters: 0.39562889970231385
Coherence score with 8 clusters: 0.42421330023367876
Coherence score with 9 clusters: 0.4045311599833526
Coherence score with 10 clusters: 0.3746757142370312
Coherence score with 11 clusters: 0.4313096523018462
Coherence score with 12 clusters: 0.4123890043835619
```



```
# perform SVD on the bag of words with the LsiModel to extract 3 topics
lsi = LsiModel(bow, num_topics=20, id2word=dictionary)
```

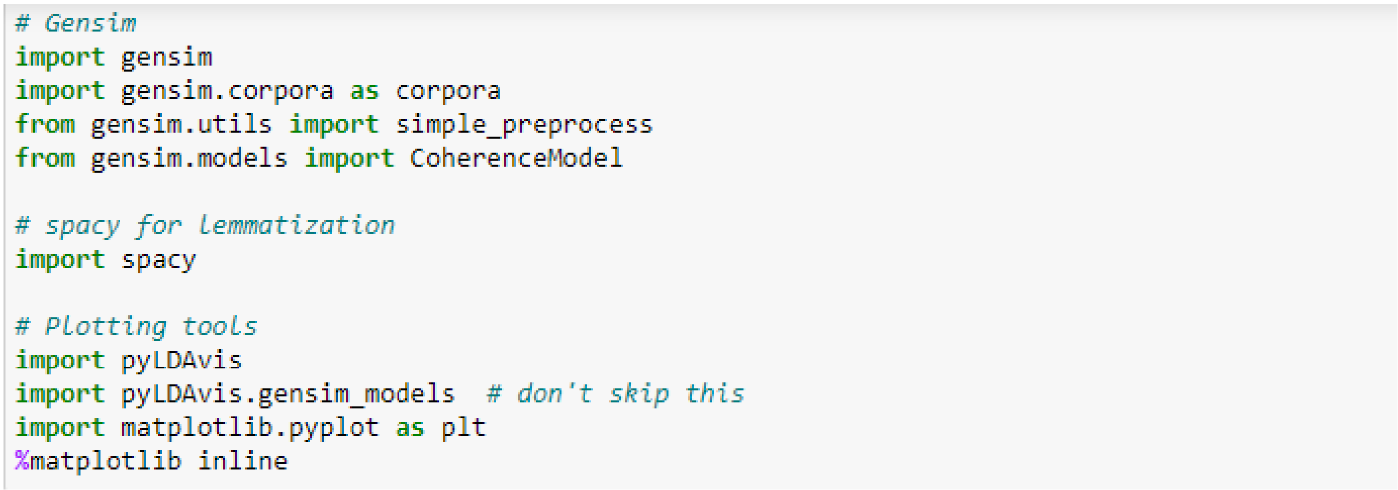
```
# find the 5 words with the strongest association to the derived topics
for topic_num, words in lsi.print_topics(num_words=5):
    print('Words in {}: {}.'.format(topic_num, words))
```

```
Words in 0: 0.538*"Now" + 0.538*"gtgt" + 0.278*"Live" + 0.269*"live" + 0.269*"StreamLive".
Words in 1: 0.482*"the" + 0.438*"UFC279" + 0.308*"to" + 0.286*"a" + 0.193*"Nate".
Words in 2: -0.456*"Diaz" + 0.375*"the" + -0.357*"Tony" + -0.351*"Ferguson" + -0.300*"vs".
Words in 3: -0.661*"the" + 0.496*"UFC279" + 0.314*"a" + -0.167*"vs" + -0.151*"Ferguson".
Words in 4: -0.611*"UFC279" + 0.473*"to" + 0.458*"a" + -0.243*"the" + 0.143*"and".
Words in 5: -0.686*"to" + 0.646*"a" + 0.128*"of" + -0.099*"I" + 0.096*"in".
Words in 6: 0.515*"Nate" + -0.343*"Gib" + 0.333*"Diaz" + -0.290*"vs" + -0.211*"UFC279".
Words in 7: -0.439*"Nate" + -0.354*"Gib" + -0.290*"Live" + -0.229*"StreamUFC279" + -0.198*"Diaz".
Words in 8: 0.573*"and" + -0.351*"to" + 0.322*"I" + -0.308*"a" + 0.218*"was".
Words in 9: 0.822*"is" + -0.311*"fight" + -0.258*"was" + 0.161*"of" + -0.160*"for".
Words in 10: 0.577*"and" + -0.374*"I" + -0.263*"was" + -0.252*"fight" + -0.252*"he".
Words in 11: 0.804*"of" + -0.349*"in" + -0.207*"and" + 0.158*"Nate" + -0.143*"is".
Words in 12: -0.574*"I" + -0.437*"for" + 0.411*"fight" + 0.256*"and" + 0.231*"was".
Words in 13: -0.654*"for" + 0.362*"in" + 0.318*"I" + 0.272*"UFC" + 0.222*"and".
Words in 14: 0.381*"in" + 0.354*"UFC" + 0.342*"fight" + 0.321*"of" + 0.287*"for".
Words in 15: -0.616*"in" + 0.516*"UFC" + 0.243*"Live" + -0.178*"Nate" + 0.175*"279".
Words in 16: -0.604*"he" + 0.525*"fight" + 0.222*"is" + 0.191*"I" + 0.189*"you".
Words in 17: -0.459*"Tony" + -0.381*"was" + 0.304*"Diaz" + 0.302*"Khamzat" + 0.216*"fight".
Words in 18: 0.390*"that" + 0.385*"was" + -0.373*"Tony" + 0.317*"you" + -0.315*"fight".
Words in 19: 0.714*"on" + -0.356*"he" + -0.337*"you" + 0.231*"was" + 0.223*"his".
```

Latent Dirichlet Analysis

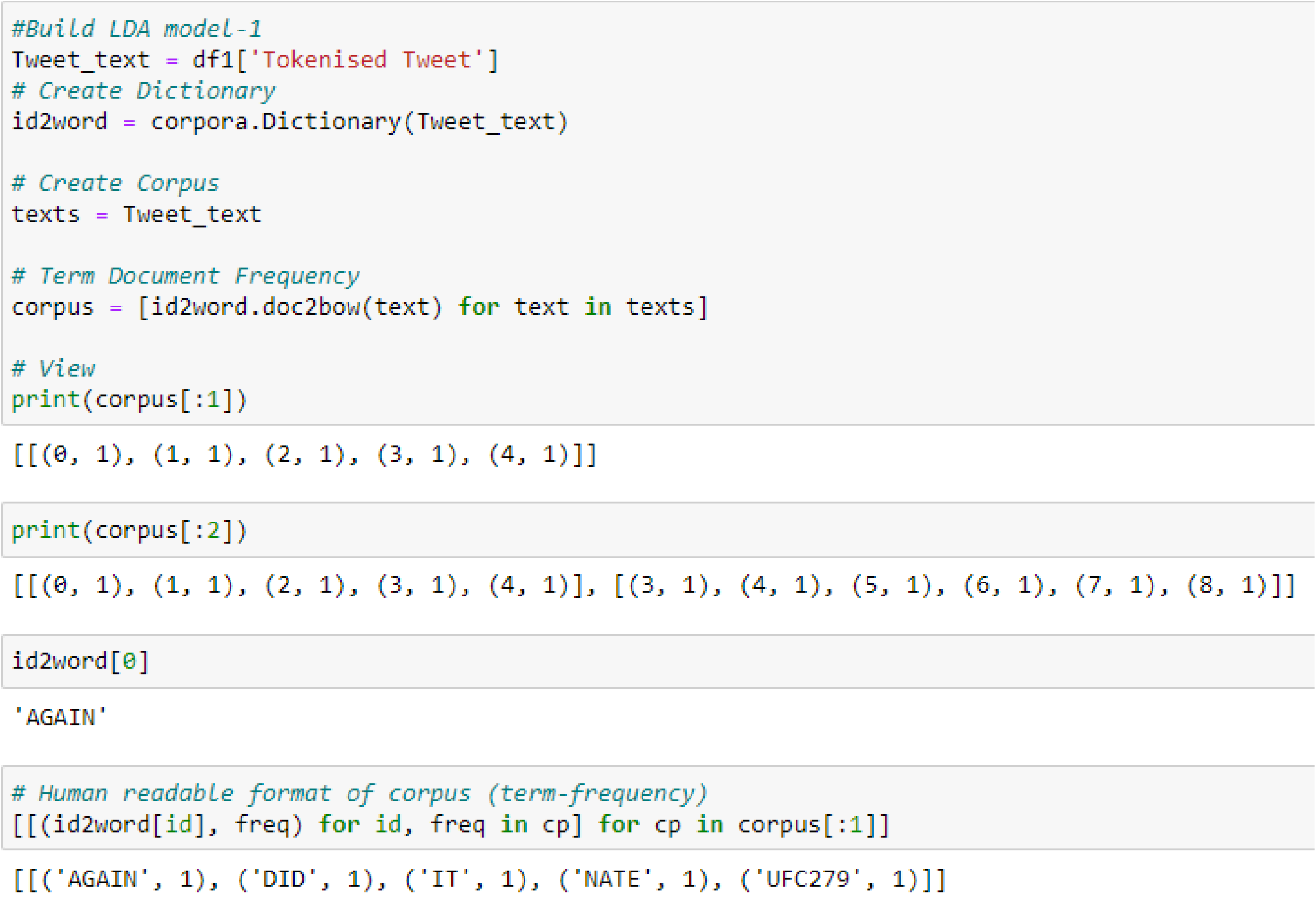
The latent Dirichlet allocation (LDA) is a generative statistical model used in natural language processing that allows groups of observations to be explained by unobserved groups that explain why some sections of the data are similar.

Import Libraries



```
# Gensim
import gensim
import gensim.corpora as corpora
from gensim.utils import simple_preprocess
from gensim.models import CoherenceModel

# spacy for lemmatization
import spacy

# Plotting tools
import pyLDAvis
import pyLDAvis.gensim_models  # don't skip this
import matplotlib.pyplot as plt
%matplotlib inline
```

Building the LDA model: In this, we are creating term document frequency.



```
#Build LDA model-1
Tweet_text = df1['Tokenised Tweet']
# Create Dictionary
id2word = corpora.Dictionary(Tweet_text)

# Create Corpus
texts = Tweet_text

# Term Document Frequency
corpus = [id2word.doc2bow(text) for text in texts]

# View
print(corpus[:1])
```

```
[[(0, 1), (1, 1), (2, 1), (3, 1), (4, 1)]]
```

```
print(corpus[:2])
```

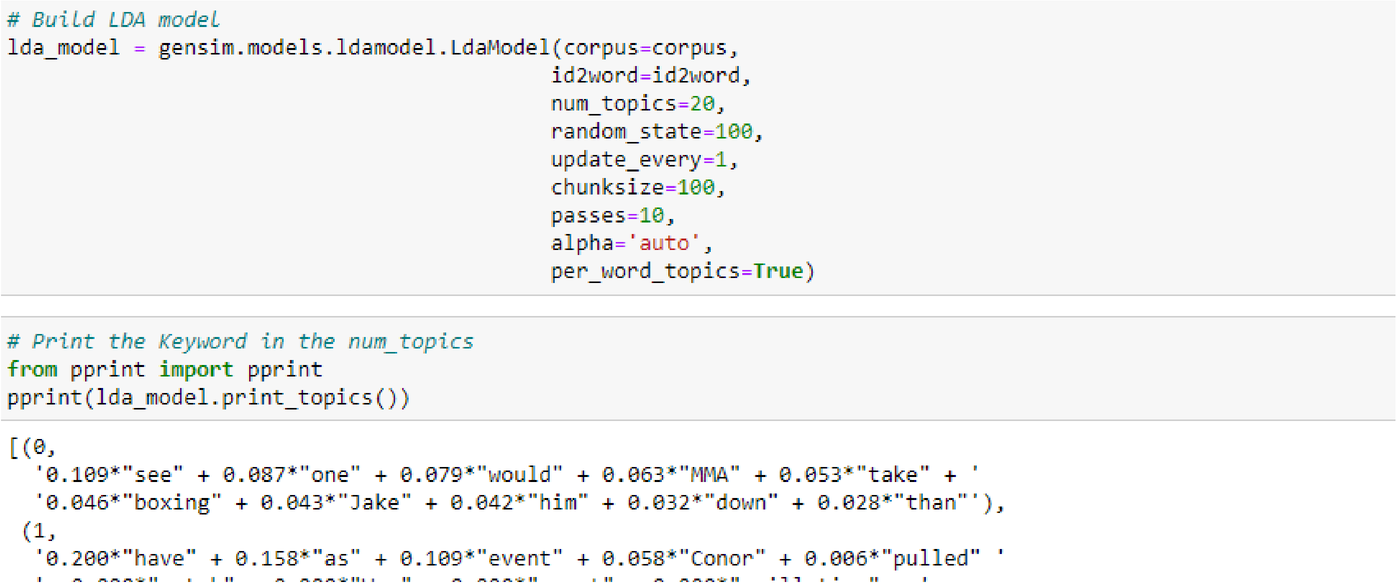
```
[[(0, 1), (1, 1), (2, 1), (3, 1), (4, 1)], [(3, 1), (4, 1), (5, 1), (6, 1), (7, 1), (8, 1)]]
```

```
id2word[0]
```

```
'AGAIN'
```

```
# Human readable format of corpus (term-frequency)
[[(id2word[id], freq) for id, freq in cp] for cp in corpus[:1]]
```

```
[[('AGAIN', 1), ('DID', 1), ('IT', 1), ('NATE', 1), ('UFC279', 1)]]
```



```
# Build LDA model
lda_model = gensim.models.ldamodel.LdaModel(corpus=corpus,
                                           id2word=id2word,
                                           num_topics=20,
                                           random_state=100,
                                           update_every=1,
                                           chunksize=100,
                                           passes=10,
                                           alpha='auto',
                                           per_word_topics=True)
```

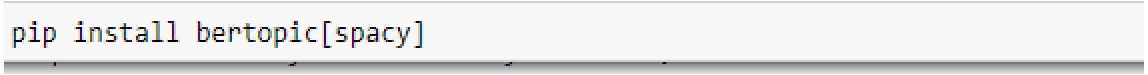
```
# Print the Keyword in the num_topics
from pprint import pprint
pprint(lda_model.print_topics())
```

```
[(0,
  '0.109*"see" + 0.087*"one" + 0.079*"would" + 0.063*"MMA" + 0.053*"take" + '
  '0.046*"boxing" + 0.043*"Jake" + 0.042*"him" + 0.032*"down" + 0.028*"than"'),
 (1,
  '0.200*"have" + 0.158*"as" + 0.109*"event" + 0.058*"Conor" + 0.006*"pulled" '
```

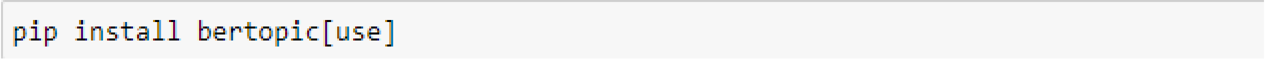
BERTopic

BERTopic is a topic modeling technique that leverages transformers and c-TF-IDF to create dense clusters allowing for easily interpretable topics along with keeping important words in the topic description.

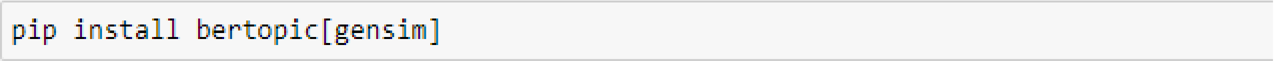
- ***Importing libraries***



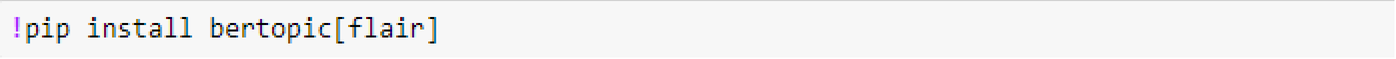
```
pip install bertopic[spacy]
```



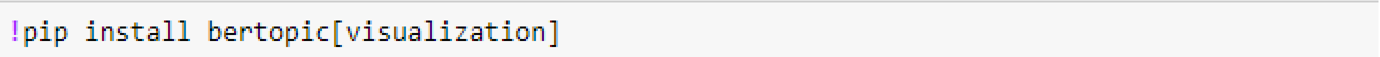
```
pip install bertopic[use]
```



```
pip install bertopic[gensim]
```

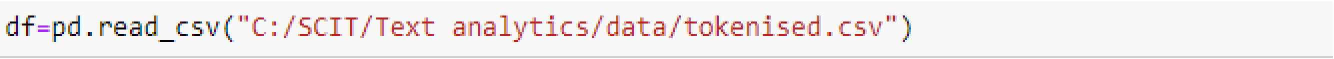


```
!pip install bertopic[flair]
```



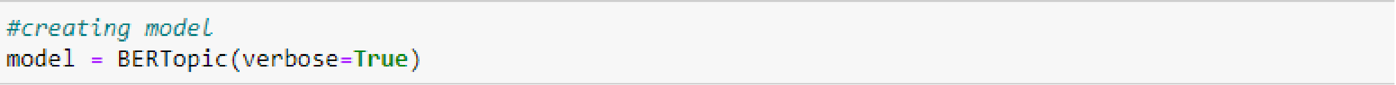
```
!pip install bertopic[visualization]
```

- ***Reading the dataset***

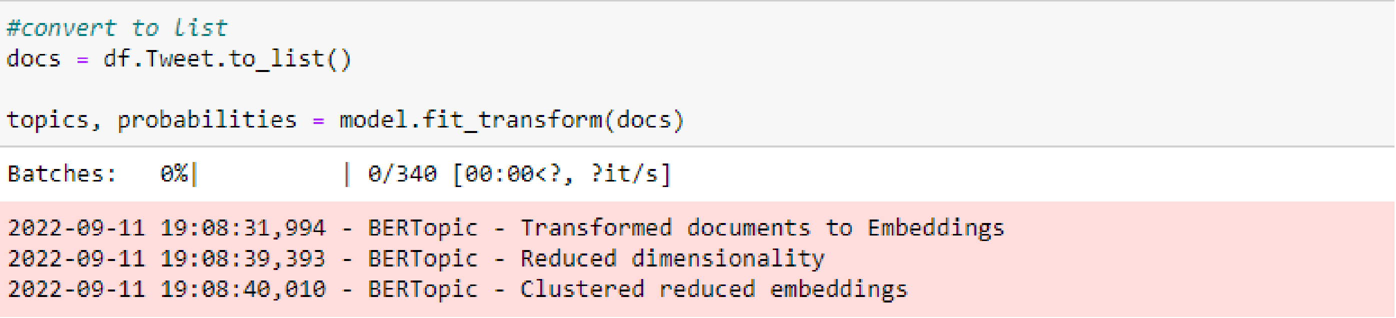


```
df=pd.read_csv("C:/SCIT/Text analytics/data/tokenised.csv")
```

- ***Creating the model***



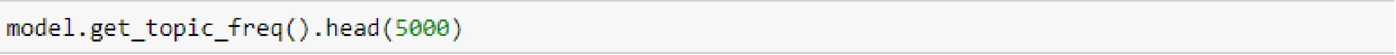
```
#creating model
model = BERTopic(verbose=True)
```



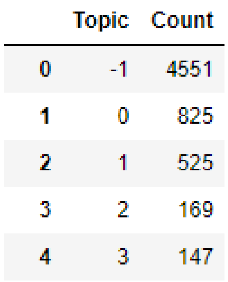
```
#convert to list
docs = df.Tweet.to_list()

topics, probabilities = model.fit_transform(docs)
```

```
Batches:   0%|          | 0/340 [00:00<?, ?it/s]
```

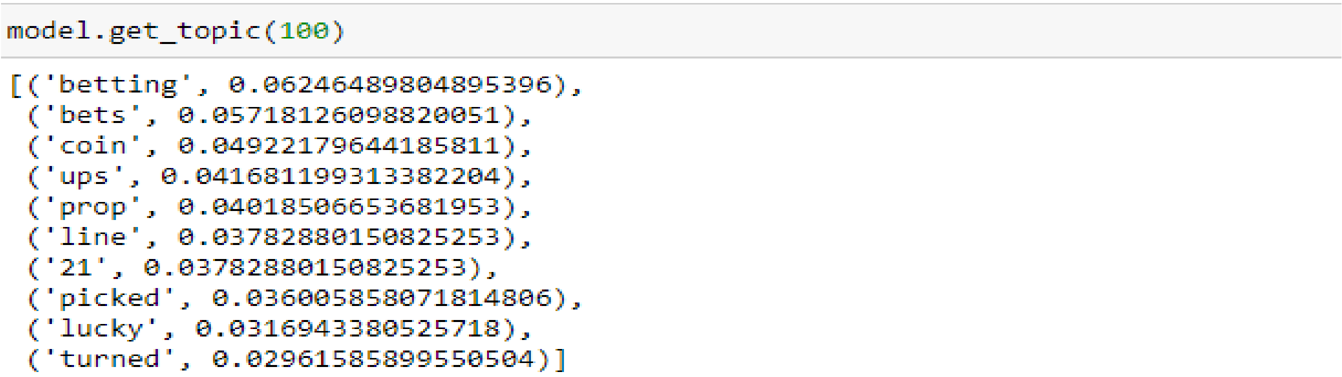
```
2022-09-11 19:08:31,994 - BERTopic - Transformed documents to Embeddings
2022-09-11 19:08:39,393 - BERTopic - Reduced dimensionality
2022-09-11 19:08:40,010 - BERTopic - Clustered reduced embeddings
```



```
model.get_topic_freq().head(5000)
```



	Topic	Count
0	-1	4551
1	0	825
2	1	525
3	2	169
4	3	147



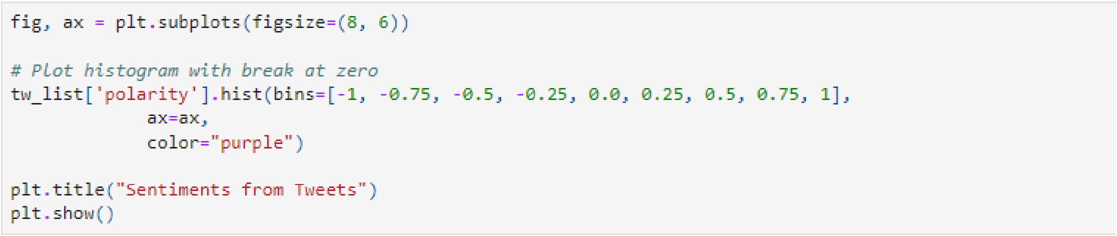
```
model.get_topic(100)
```

```
[('betting', 0.06246489804895396),
 ('bets', 0.05718126098820051),
 ('coin', 0.04922179644185811),
 ('ups', 0.041681199313382204),
 ('prop', 0.04018506653681953),
 ('line', 0.03782880150825253),
 ('21', 0.03782880150825253),
 ('picked', 0.036005858071814806),
 ('lucky', 0.0316943380525718),
 ('turned', 0.02961585899550504)]
```

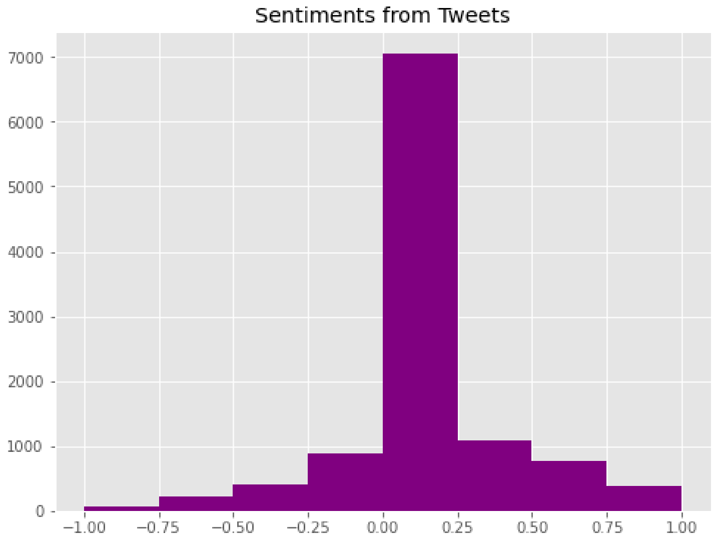
7. Data Visualization

Data visualization is the graphical representation of information and data. By using visual elements like charts, graphs, and maps, data visualization tools provide an accessible way to see and understand trends, outliers, and patterns in data.

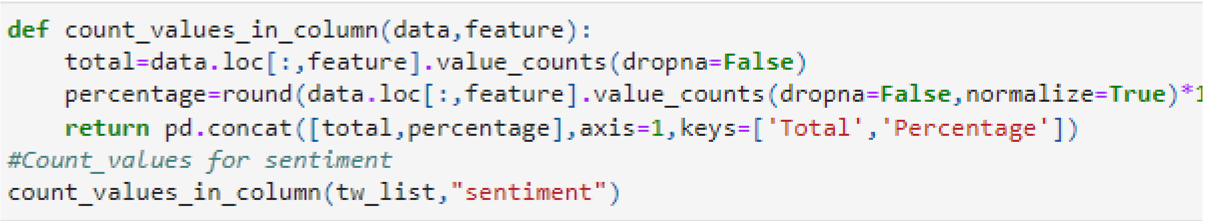
- Graph based on Polarity score of the tweets.



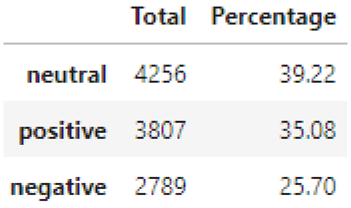
```
fig, ax = plt.subplots(figsize=(8, 6))

# Plot histogram with break at zero
tw_list['polarity'].hist(bins=[-1, -0.75, -0.5, -0.25, 0.0, 0.25, 0.5, 0.75, 1],
             ax=ax,
             color="purple")

plt.title("Sentiments from Tweets")
plt.show()
```



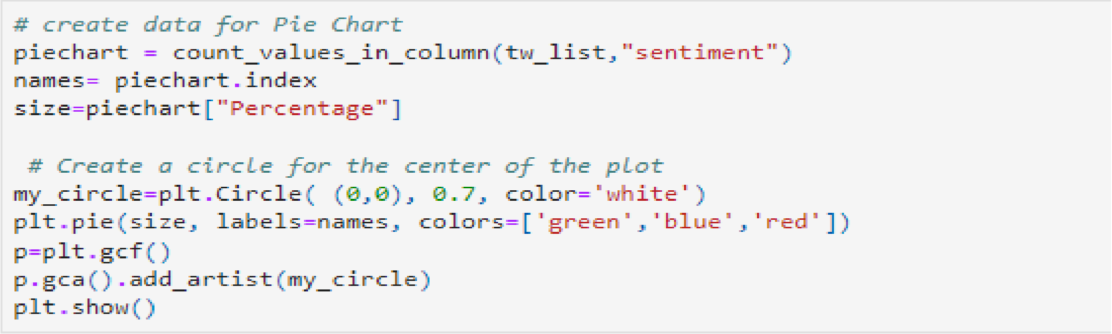
- Pie Chart of the total percentage of positive, negative and neutral score.



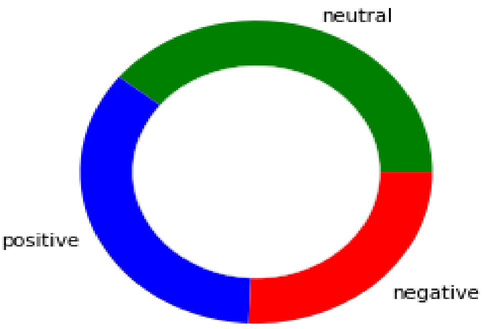
```
def count_values_in_column(data,feature):
    total=data.loc[:,feature].value_counts(dropna=False)
    percentage=round(data.loc[:,feature].value_counts(dropna=False,normalize=True)*1
    return pd.concat([total,percentage],axis=1,keys=['Total','Percentage'])
#Count_values for sentiment
count_values_in_column(tw_list,"sentiment")
```



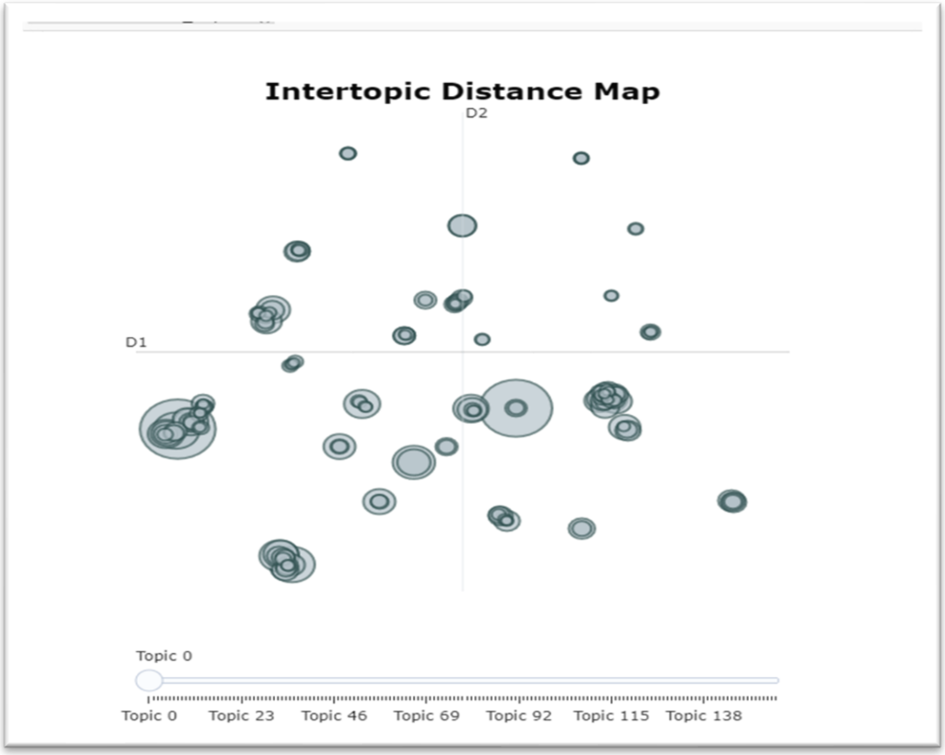
	Total	Percentage
neutral	4256	39.22
positive	3807	35.08
negative	2789	25.70



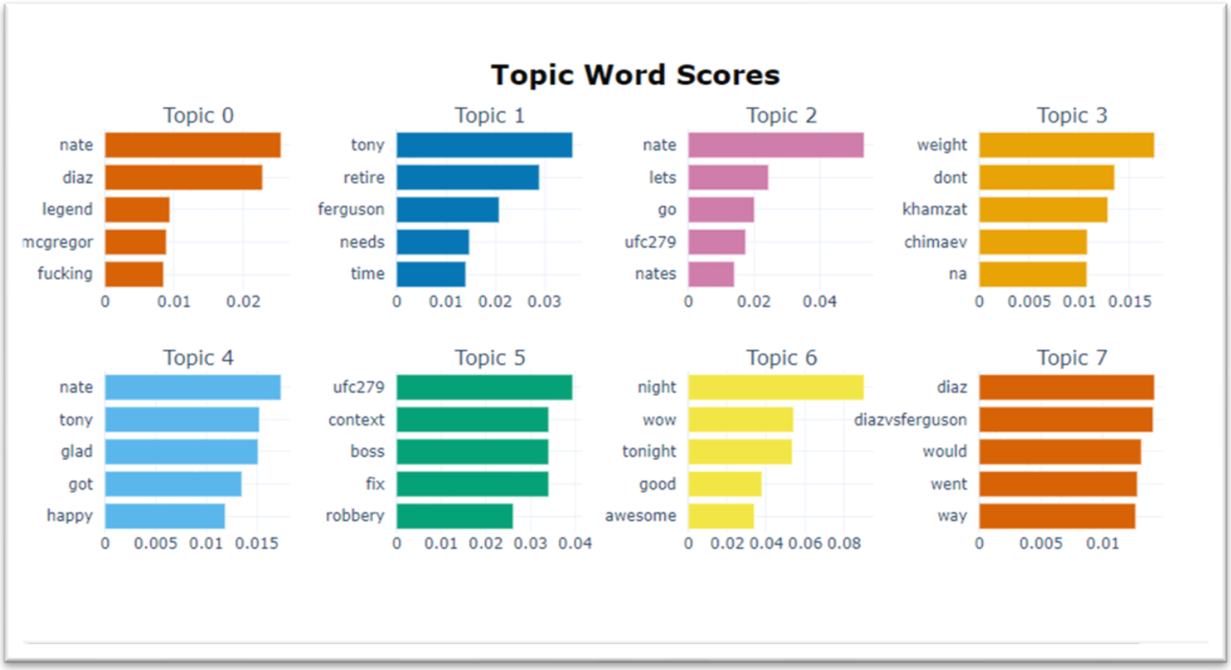
```
# create data for Pie Chart
piechart = count_values_in_column(tw_list,"sentiment")
names= piechart.index
size=piechart["Percentage"]

 # Create a circle for the center of the plot
my_circle=plt.Circle( (0,0), 0.7, color='white')
plt.pie(size, labels=names, colors=['green','blue','red'])
p=plt.gcf()
p.gca().add_artist(my_circle)
plt.show()
```



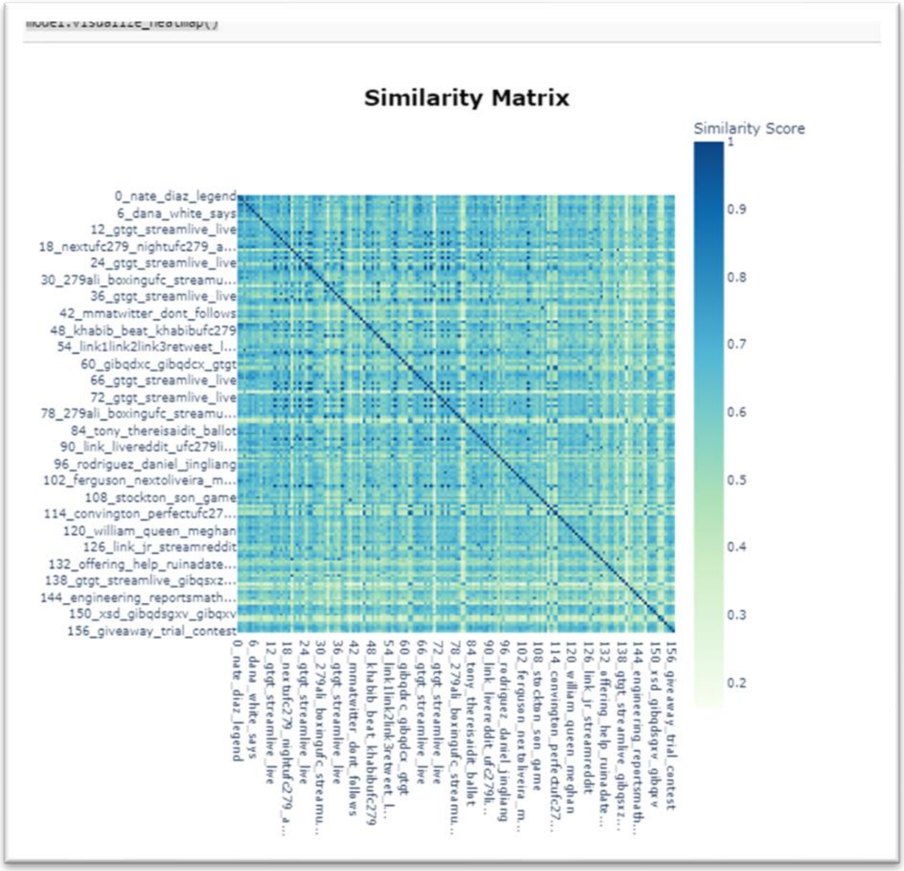
- Distance map of topics



- Bar charts based on topic word score.



- Heatmap based on Similarity score

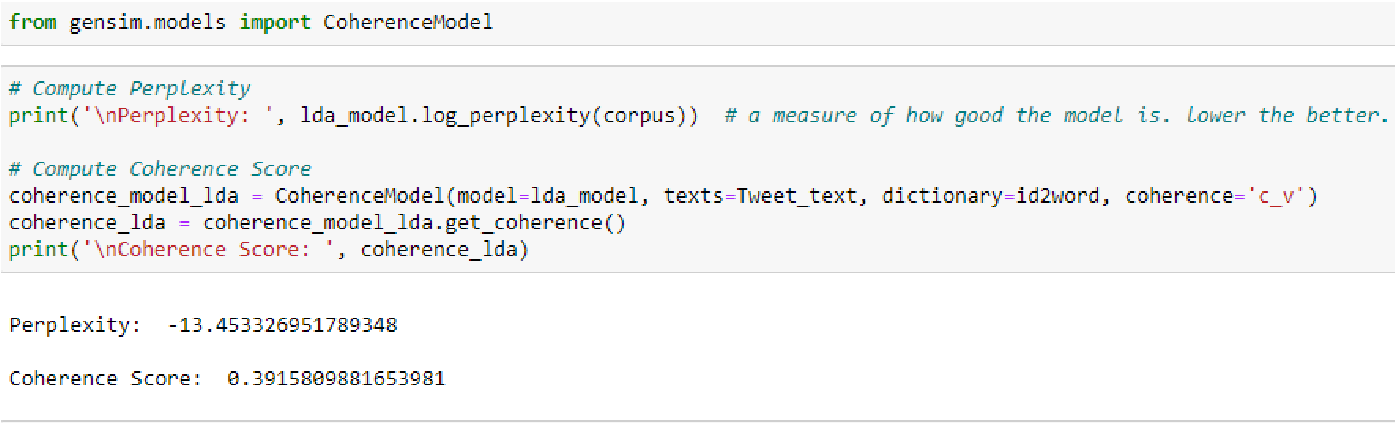


```
model.visualize_heatmap()
```

8. Compute the perplexity and coherence score

Perplexity is a statistical measure of how well a probability model predicts a sample or It captures how surprised a model is of new data it has not seen before, and is measured as the normalized log-likelihood of a held-out test set.
Topic Coherence measures score a single topic by measuring the degree of semantic similarity between high scoring words in the topic. These measurements help distinguish between topics that are semantically interpretable topics and topics that are artifacts of statistical inference.

In Latent Dirichlet analysis, we are getting perplexity score as -13.4533 and coherence score as 0.3915.



```
from gensim.models import CoherenceModel
```

```
# Compute Perplexity
print('\nPerplexity: ', lda_model.log_perplexity(corpus))  # a measure of how good the model is. lower the better.

# Compute Coherence Score
coherence_model_lda = CoherenceModel(model=lda_model, texts=Tweet_text, dictionary=id2word, coherence='c_v')
coherence_lda = coherence_model_lda.get_coherence()
print('\nCoherence Score: ', coherence_lda)
```

```

Perplexity:  -13.453326951789348

Coherence Score:  0.3915809881653981
```