

```
1 # Install Libraries
2 !pip install textblob
3 !pip install tweepy
```

```
1 # Import Libraries
2
3 from textblob import TextBlob
4 import sys
5 import tweepy
6 import matplotlib.pyplot as plt
7 import pandas as pd
8 import numpy as np
9 import os
10 import nltk
11 import pycountry
12 import re
13 import string
14
15 from wordcloud import WordCloud, STOPWORDS
16 from PIL import Image
17 from nltk.sentiment.vader import SentimentIntensityAnalyzer
18 from langdetect import detect
19 from nltk.stem import SnowballStemmer
20 from nltk.sentiment.vader import SentimentIntensityAnalyzer
21 from sklearn.feature_extraction.text import CountVectorizer
```

```
1 # Authentication
2 consumerKey = "Type your key"
3 consumerSecret = "Type your secret"
4 accessToken = "Type your authentication token"
5 accessTokenSecret = "Type your authentication token secret"
6
7 auth = tweepy.OAuthHandler(consumerKey, consumerSecret)
8 auth.set_access_token(accessToken, accessTokenSecret)
9 api = tweepy.API(auth)
```

```
1 #Sentiment Analysis
2
3 def percentage(part,whole):
4     return 100 * float(part)/float(whole)
5
6 keyword = input("enter keyword or hashtag: ")
7 noOfTweet = int(input ("enter how many tweets you need to analyze:"))
8
```

```
9
10 tweets = tweepy.Cursor(api.search, q=keyword).items(noOfTweet)
11 positive = 0
12 negative = 0
13 neutral = 0
14 polarity = 0
15 tweet_list = []
16 neutral_list = []
17 negative_list = []
18 positive_list = []
19
20 for tweet in tweets:
21
22     #print(tweet.text)
23     tweet_list.append(tweet.text)
24     analysis = TextBlob(tweet.text)
25     score = SentimentIntensityAnalyzer().polarity_scores(tweet.text)
26     neg = score['neg']
27     neu = score['neu']
28     pos = score['pos']
29     comp = score['compound']
30     polarity += analysis.sentiment.polarity
31
32     if neg > pos:
33         negative_list.append(tweet.text)
34         negative += 1
35
36     elif pos > neg:
37         positive_list.append(tweet.text)
38         positive += 1
39
40     elif pos == neg:
41         neutral_list.append(tweet.text)
42         neutral += 1
43
44 positive = percentage(positive, noOfTweet)
45 negative = percentage(negative, noOfTweet)
46 neutral = percentage(neutral, noOfTweet)
47 polarity = percentage(polarity, noOfTweet)
48 positive = format(positive, '.1f')
49 negative = format(negative, '.1f')
50 neutral = format(neutral, '.1f')
```

Please enter keyword or hashtag to search: lockdown2 london  
Please enter how many tweets to analyze: 2500

```

1 #Number of Tweets (Total, Positive, Negative, Neutral)
2 tweet_list = pd.DataFrame(tweet_list)
3 neutral_list = pd.DataFrame(neutral_list)
4 negative_list = pd.DataFrame(negative_list)
5 positive_list = pd.DataFrame(positive_list)
6 print("total : ",len(tweet_list))
7 print("positive: ",len(positive_list))
8 print("negative: ", len(negative_list))
9 print("neutral: ",len(neutral_list))

```

```

total number: 2500
positive number: 1025
negative number: 580
neutral number: 895

```

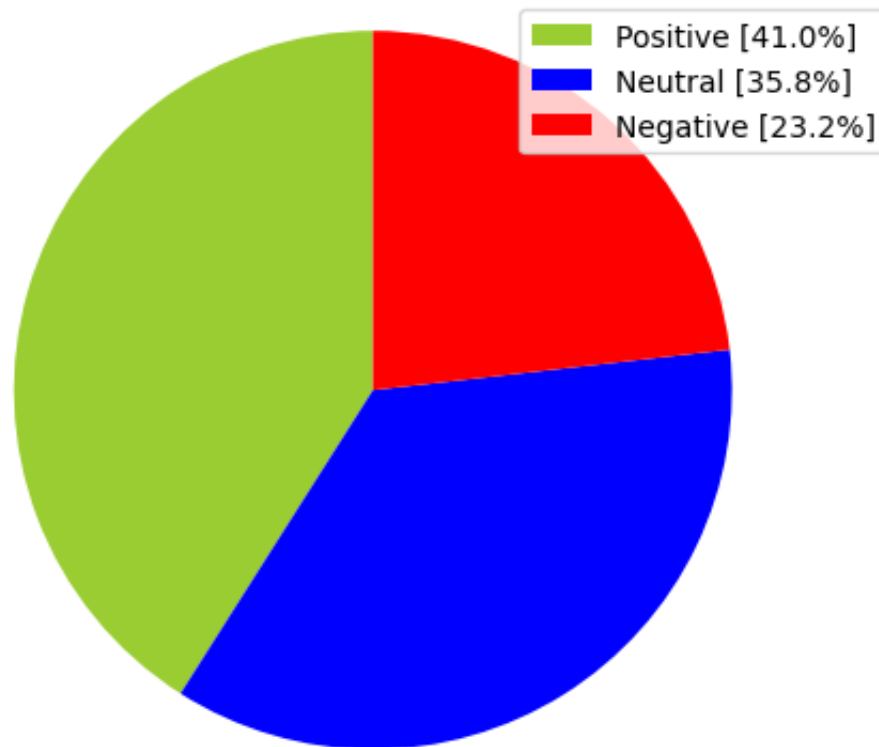
```
1 tweet_list
```

	0
0	RT @Petethestreet1: #loweringsun on #christmas...
1	RT @LondonEconomic: Protesters, very few of wh...
2	RT @LondonEconomic: Protesters, very few of wh...
3	Photo Journal - Day 01\n\n#lockdown2 #lockdown...
4	God love 'em - @SlowRichies opened the doors o...
...	...
2495	RT @lucywonder14: @Ldn_Ambulance 68% increase ...
2496	RT @judehaste_write: 🌟 #comedy 🌟 #contemporary...
2497	RT @judehaste_write: 🌟 #comedy 🌟 #contemporary...
2498	RT @petsarefound: Please #RT to help #FindLola...
2499	RT @judehaste_write: 🌟 #comedy 🌟 #contemporary...

2500 rows x 1 columns

```
1 #Creating PieCart
2
3 labels = ['Positive ['+str(positive)+'%]' , 'Neutral ['+str(neutral)+'%]', 'Ne
4 sizes = [positive, neutral, negative]
5 colors = ['yellowgreen', 'blue','red']
6 patches, texts = plt.pie(sizes,colors=colors, startangle=90)
7 plt.style.use('default')
8 plt.legend(labels)
9 plt.axis('equal')
10 plt.show()
```

Sentiment Analysis Result for keyword= lockdown2 london



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

Double-click (or enter) to edit

## ▼ Extracting text values

```
text_all = tweet_list[0].values text_neutral = neutral_list[0].values text_positive =
positive_list[0].values text_negative = negative_list[0].values
```

```

1 tw_list = pd.DataFrame(tweet_list)
2 tw_list["text"] = tw_list[0]
3 tw_list

```

	0	text	polarity	subjectivity	sentin
0	RT @Petethstreet1: #loweringsun on #christmas...	RT @Petethstreet1: #loweringsun on #christmas...	0.700000	0.600000	pos
1	RT @LondonEconomic: Protesters, very few of wh...	RT @LondonEconomic: Protesters, very few of wh...	-0.260000	0.130000	pos
3	Photo Journal - Day 01\n\n#lockdown2 #lockdown...	Photo Journal - Day 01\n\n#lockdown2 #lockdown...	0.000000	0.000000	ne
4	God love 'em - @SlowRichies opened the doors o...	God love 'em - @SlowRichies opened the doors o...	0.375000	0.466667	pos
5	So might wear my #addidas #prideshots for #lo...	So might wear my #addidas #prideshots for #lo...	0.000000	0.000000	ne
...	...	...	...	...	...
2461	You are NOT alone ❤️ \n#WeLoveYouChanyeol \n#Mad...	You are NOT alone ❤️ \n#WeLoveYouChanyeol \n#Mad...	0.000000	0.000000	pos
2465	So I live in West Yorkshire, heading into tier...	So I live in West Yorkshire, heading into tier...	0.136364	0.500000	ne
2467	RT @julian2tweet: Expect to see similar scenes...	RT @julian2tweet: Expect to see similar scenes...	0.000000	0.200000	ne

```
1 tweet_list
```

```
0
```

---

```

0      RT @Petethestreet1: #loweringsun on #christmas...
1      RT @LondonEconomic: Protesters, very few of wh...
3      Photo Journal - Day 01\n\n#lockdown2 #lockdown...
4      God love 'em - @SlowRichies opened the doors o...
5      So might wear my #addidas #pridesshorts for #lo...
...
2461  You are NOT alone ❤️\n#WeLoveYouChanyeol \n#Mad...
2465      So I live in West Yorkshire, heading into tier...
2467      RT @julian2tweet: Expect to see similar scenes...
2472      RT @petsarefound: Please #RT to help #FindLola...
2473      Expect to see similar scenes for London #Lockd...
```

```
1281 rows x 1 columns
```

```

1 #Cleaning Text (RT, Punctuation etc)
2
3 #Creating new dataframe and new features
4 tw_list = pd.DataFrame(tweet_list)
5 tw_list["text"] = tw_list[0]
6
7 #Removing RT, Punctuation etc
8 remove_rt = lambda x: re.sub('RT @\w+: ', "", x)
9 rt = lambda x: re.sub("(@[A-Za-z0-9]+)|([^0-9A-Za-z \t])|(\w+:\/\/\S+)", "", x)
10 tw_list["text"] = tw_list.text.map(remove_rt).map(rt)
11 tw_list["text"] = tw_list.text.str.lower()
12 tw_list.head(10)

```

	0	text
0	RT @Petethstreet1: #loweringsun on #christmas...	loweringsun on christmaslights thestrand ...
1	RT @LondonEconomic: Protesters, very few of wh...	protesters very few of whom were wearing fac...
3	Photo Journal - Day 01\n\n#lockdown2 #lockdown...	photo journal day 01 lockdown2 lockdown20...
4	God love 'em - @SlowRichies opened the doors O...	god love em opened the doors of their res...
5	So might wear my #addidas #pridesshorts for #lo...	so might wear my addidas pridesshorts for lo...
6	RT @basicincome_uk: BREAKING: @sianberry &amp;...	breaking amp will be putting forward a...
7	Praticamente è così \n#6Novembre #COVID19 #l...	praticamente cos 6novembre covid19 look

```

1 #Calculating Negative, Positive, Neutral and Compound values
2
3 tw_list[['polarity', 'subjectivity']] = tw_list['text'].apply(lambda Text: pd
4 for index, row in tw_list['text'].iteritems():
5     score = SentimentIntensityAnalyzer().polarity_scores(row)
6     neg = score['neg']
7     neu = score['neu']
8     pos = score['pos']
9     comp = score['compound']
10    if neg > pos:
11        tw_list.loc[index, 'sentiment'] = "negative"
12    elif pos > neg:
13        tw_list.loc[index, 'sentiment'] = "positive"
14    else:
15        tw_list.loc[index, 'sentiment'] = "neutral"
16    tw_list.loc[index, 'neg'] = neg
17    tw_list.loc[index, 'neu'] = neu
18    tw_list.loc[index, 'pos'] = pos
19    tw_list.loc[index, 'compound'] = comp
20
21 tw_list.head(10)

```

	0	text	polarity	subjectivity	sentiment	neg	1
0	RT @Petethestreet1: #loweringsun on #christmas...	loweringsun on christmaslights thestrans...	0.700	0.600000	positive	0.000	0.0
1	RT @LondonEconomic: Protesters, very few of wh...	protesters very few of whom were wearing fac...	-0.260	0.130000	positive	0.079	0.0
3	Photo Journal - Day 01\n\n#lockdown2 #lockdown...	photo journal day 01 lockdown2 lockdown20...	0.000	0.000000	neutral	0.000	1.0
4	God love 'em - @SlowRichies opened the doors o...	god love em opened the doors of their res...	0.375	0.466667	positive	0.000	0.0



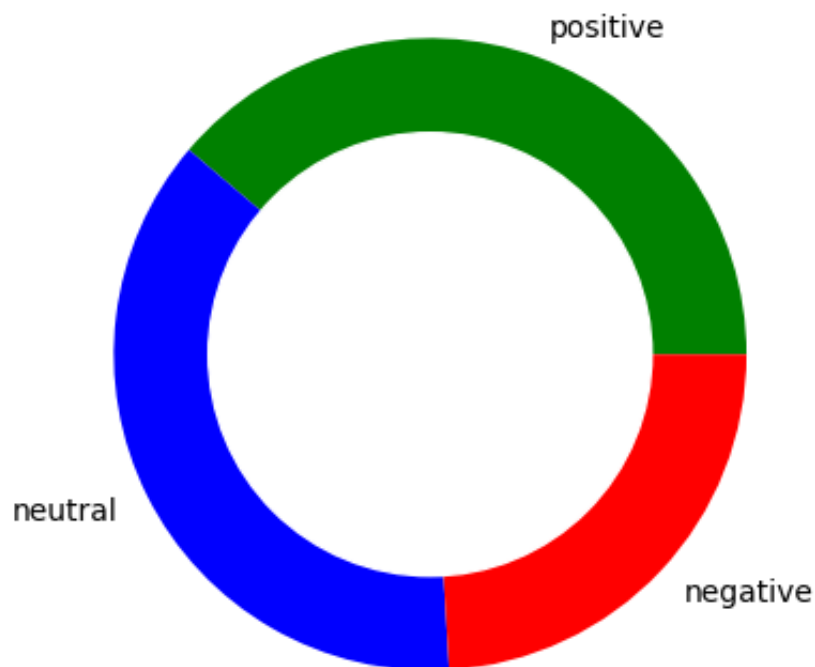
```
1 #Creating new data frames for all sentiments (positive, negative and neutral)
2
3 tw_list_negative = tw_list[tw_list["sentiment"]=="negative"]
4 tw_list_positive = tw_list[tw_list["sentiment"]=="positive"]
5 tw_list_neutral = tw_list[tw_list["sentiment"]=="neutral"]

1 #Function for count_values_in single columns
2
3 def count_values_in_column(data,feature):
4     total=data.loc[:,feature].value_counts(dropna=False)
5     percentage=round(data.loc[:,feature].value_counts(dropna=False,normalize=
6     return pd.concat([total,percentage],axis=1,keys=['Total','Percentage'])

1 #Count_values for sentiment
2 count_values_in_column(tw_list,"sentiment")
```

	Total	Percentage
positive	497	38.80
neutral	476	37.16
negative	308	24.04

```
1 # create data for Pie Chart
2 pichart = count_values_in_column(tw_list,"sentiment")
3 names= pc.index
4 size=pc["Percentage"]
5
6 # Create a circle for the center of the plot
7 my_circle=plt.Circle( (0,0), 0.7, color='white')
8 plt.pie(size, labels=names, colors=['green','blue','red'])
9 p=plt.gcf()
10 p.gca().add_artist(my_circle)
11 plt.show()
```



```
1 #Creating wordcloud for all tweets
2 create_wordcloud(tw_list["text"].values)
```

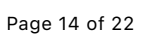
Word Cloud Saved Successfully



```
1 #Creating wordcloud for negative sentiment  
2 create_wordcloud(tw_list_negative["text"].values)
```

Word Cloud Saved Successfully







```
1 round(pd.DataFrame(tw_list.groupby("sentiment").text_len.mean()),2)
```

text_len	
sentiment	
negative	109.17
neutral	97.20
positive	108.87

```
1 round(pd.DataFrame(tw_list.groupby("sentiment").text_word_count.mean()),2)
```

text_word_count	
sentiment	
negative	17.48
neutral	14.70
positive	17.99

```
1 #Removing Punctuation
2 def remove_punct(text):
3     text = "".join([char for char in text if char not in string.punctuation])
4     text = re.sub('[0-9]+', '', text)
5     return text
6
7 tw_list['punct'] = tw_list['text'].apply(lambda x: remove_punct(x))
```

```
1 #Applying tokenization
2 def tokenization(text):
3     text = re.split('\W+', text)
4     return text
5
6 tw_list['tokenized'] = tw_list['punct'].apply(lambda x: tokenization(x.lower()))
```



```
1 #Removing stopwords
2 stopword = nltk.corpus.stopwords.words('english')
3 def remove_stopwords(text):
4     text = [word for word in text if word not in stopword]
5     return text
6
7 tw_list['nonstop'] = tw_list['tokenized'].apply(lambda x: remove_stopwords(x))

1 #Appliyng Stemmer
2 ps = nltk.PorterStemmer()
3
4 def stemming(text):
5     text = [ps.stem(word) for word in text]
6     return text
7
8 tw_list['stemmed'] = tw_list['nonstop'].apply(lambda x: stemming(x))

1 #Cleaning Text
2 def clean_text(text):
3     text_lc = "".join([word.lower() for word in text if word not in string.pu
4     text_rc = re.sub('[0-9]+', '', text_lc)
5     tokens = re.split('\W+', text_rc)    # tokenization
6     text = [ps.stem(word) for word in tokens if word not in stopword]  # remo
7     return text
```

```
1 tw_list.head()
```

	0	text	polarity	subjectivity	sentiment	neg	pos
0	RT @Petethestreet1: #loweringsun on #christmas...	loweringsun on christmaslights thestrand ...	0.700	0.600000	positive	0.000	0.8
1	RT @LondonEconomic: Protesters, very few of wh...	protesters very few of whom were wearing fac...	-0.260	0.130000	positive	0.079	0.7
3	Photo Journal - Day 01\n\n#lockdown2 #lockdown...	photo journal day 01 lockdown2 lockdown20...	0.000	0.000000	neutral	0.000	1.0
4	God love 'em - @SlowRichies opened the doors o...	god love em opened the doors of their res...	0.375	0.466667	positive	0.000	0.7
5	So might wear my #addidas #prideshots for #lo...	so might wear my addidas prideshots for lo...	0.000	0.000000	neutral	0.000	1.0

```
1 #Applying Countvectorizer
```

```
2 countVectorizer = CountVectorizer(analyzer=clean_text)
```

```
3 countVector = countVectorizer.fit_transform(tw_list['text'])
```

```
4 print('{} Number of reviews has {} words'.format(countVector.shape[0], countV
```

```
5 #print(countVectorizer.get_feature_names())
```

```
1281 Number of reviews has 2966 words
```

```
1 count_vect_df = pd.DataFrame(countVector.toarray(), columns=countVectorizer.g
2 count_vect_df.head()
```

		aba	abbey	abc	abi	abo	abseil	absolut	ac	acab	...	zatwardzia	zć
0	2	0	0	0	0	0	0	0	0	0	...	0	
1	2	0	0	0	0	0	0	0	0	0	...	0	
2	1	0	0	0	0	0	0	0	0	0	...	0	
3	1	0	0	0	0	0	0	0	0	0	...	0	
4	1	0	0	0	0	0	0	0	0	0	...	0	

5 rows × 2966 columns

```
1 # Most Used Words
2 count = pd.DataFrame(count_vect_df.sum())
3 countdf = count.sort_values(0,ascending=False).head(20)
4 countdf[1:11]
```

	0
<b>lockdown</b>	976
<b>london</b>	793
<b>day</b>	110
<b>covid</b>	106
<b>amp</b>	82
<b>uk</b>	70
<b>go</b>	67
<b>new</b>	67
<b>last</b>	61
<b>morn</b>	60

```
1 #Function to ngram
2 def get_top_n_gram(corpus,ngram_range,n=None):
3     vec = CountVectorizer(ngram_range=ngram_range,stop_words = 'english').fit
4     bag_of_words = vec.transform(corpus)
5     sum_words = bag_of_words.sum(axis=0)
6     words_freq = [(word, sum_words[0, idx]) for word, idx in vec.vocabulary_.
7     words_freq =sorted(words_freq, key = lambda x: x[1], reverse=True)
8     return words_freq[:n]
```

```
1 #n2_bigram
2 n2_bigrams = get_top_n_gram(tw_list['text'],(2,2),20)
3
4 n2_bigrams
```

```
[('london lockdown2', 81),
 ('lockdown2 london', 58),
 ('day lockdown2', 30),
 ('central london', 29),
 ('wednesdaymegaword elections2020', 27),
 ('lockdown lockdown2', 26),
 ('new lockdown', 23),
 ('lockdown2 lockdownuk', 23),
 ('elections2020 covid19', 23),
 ('gallery london', 22),
 ('covid19 poland', 21),
 ('london lockdown', 20),
 ('lockdown2 lockdown', 20),
 ('political cartoon', 18),
 ('cartoon gallery', 18),
 ('london new', 16),
 ('national lockdown', 16),
 ('uknews london', 15),
 ('lockdown london', 14),
 ('breaking uknews', 14)]
```

```
1 #n3_trigram
2 n3_trigrams = get_top_n_gram(tw_list['text'],(3,3),20)
3
4 n3_trigrams

[('wednesdaymegaword elections2020 covid19', 23),
 ('elections2020 covid19 poland', 20),
 ('political cartoon gallery', 18),
 ('cartoon gallery london', 17),
 ('breaking uknews london', 14),
 ('covid19 poland electionnight', 12),
 ('million mask march', 11),
 ('missing ginger tabby', 11),
 ('ginger tabby female', 11),
 ('uknews london brexit', 10),
 ('lockdown lockdown2 political', 10),
 ('lockdown2 political cartoon', 10),
 ('borisjohnson imposing new', 10),
 ('imposing new lockdown', 10),
 ('new lockdown lockdown2', 10),
 ('findlola missing ginger', 9),
 ('tabby female lostcat', 9),
 ('female lostcat stokenewington', 9),
 ('lostcat stokenewington n16', 9),
 ('stokenewington n16 london', 9)]
```

[Colab paid products](#) - [Cancel contracts here](#)

