Text Analytics Assignment-2

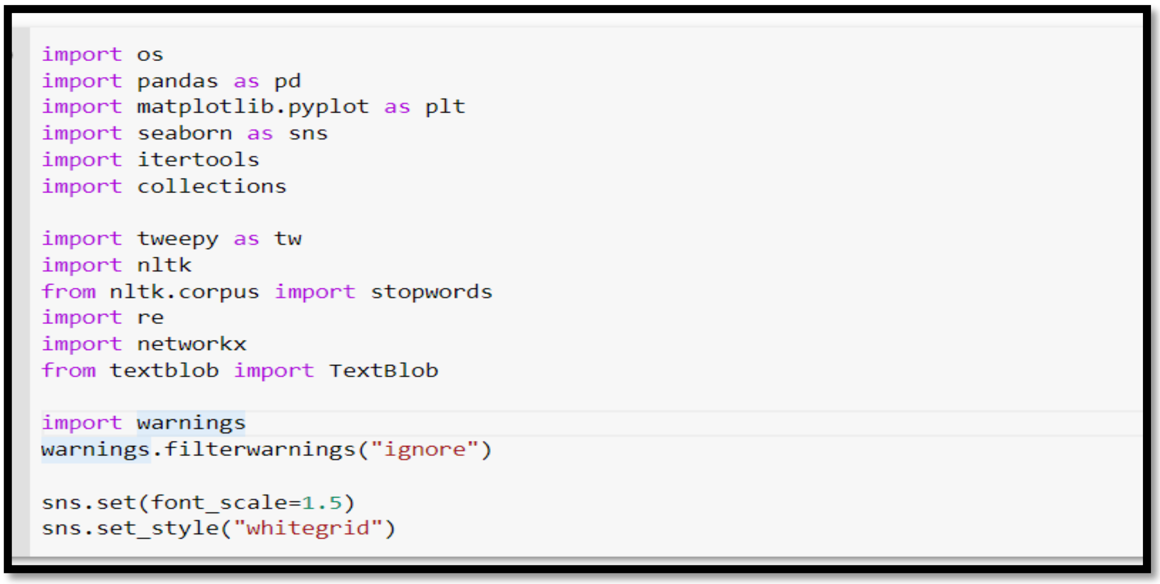
Sentiment Analysis on **Lebanon Explosion**

Name- Anshika Jain

Roll number – 26

PGDM RBA

1.Importing all relevant libraries and Packages

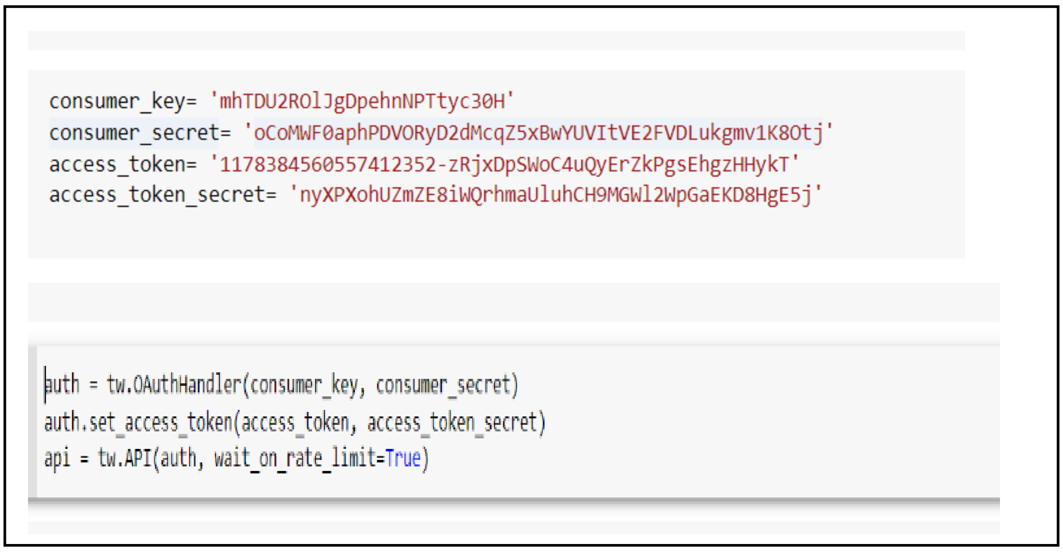


```
import os
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import itertools
import collections

import tweepy as tw
import nltk
from nltk.corpus import stopwords
import re
import networkx
from textblob import TextBlob

import warnings
warnings.filterwarnings("ignore")

sns.set(font_scale=1.5)
sns.set_style("whitegrid")
```

- Imported **Pandas** for importing data and creating python object
- **Matplotlib.pyplot** for plotting various graphs.
- **Itertools** to handle iterators like while and for loop.
- **Collections** are the containers that are used to store collections of data, for example, list, dict, set, tuple etc.
- Import **seaborn** library to create more attractive and informative statistical graphics.
- Imported **Tweepy** library for accessing the Twitter API.
- Imported **NLTK** library for doing tasks on NLP (Natural Language processing).

- Imported **stopwords** from nltk.corpus to remove all stopwords from our data.
- Imported **re (RegEx)** which is used to check if a string contains the specified search pattern.
- Imported **networkX** package for the creation, manipulation, and study of the structure, dynamics, and functions of complex networks.
- Imported **Textblob** library for processing textual data. It provides a simple API for diving into common natural language processing (NLP) tasks
- Imported **warnings** to generate warning signals.

2.Defining the keys

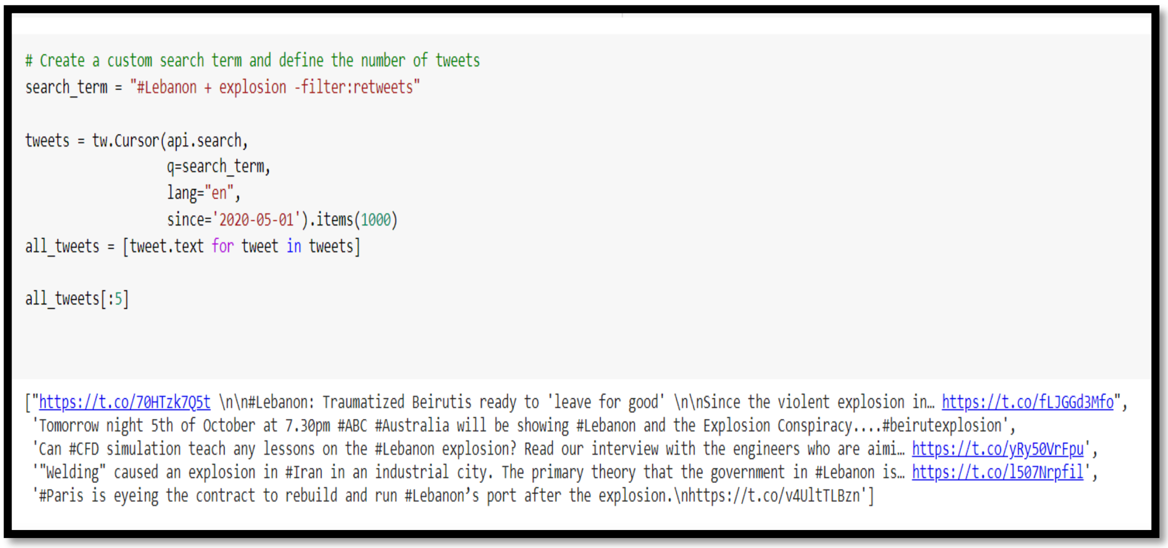


```
consumer_key= 'mhTDU2ROlJgDpehnNPTtyc30H'
consumer_secret= 'oCoMWF0aphPDVORyD2dMcqZ5xBwYUVItVE2FVDLukgmv1K8Otj'
access_token= '1178384560557412352-zRjxDpSWoC4uQyErZkPgsEhgzHHykT'
access_token_secret= 'nyXPXohUZmZE8iWQrhmaUluhCH9MGWl2WpGaEKD8HgE5j'
```

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

- This data is obtained from the developer portal on Twitter application where a unique API key, secret key and authentication tokens are generated for the developer accounts.
- Specifically, keys are unique identifiers that authenticate your App's request, while tokens are a type of authorization for an App to gain specific access to data.

3. Search the term and creating a list



```
# Create a custom search term and define the number of tweets
search_term = "#Lebanon + explosion -filter:retweets"

tweets = tw.Cursor(api.search,
                   q=search_term,
                   lang="en",
                   since='2020-05-01').items(1000)
all_tweets = [tweet.text for tweet in tweets]

all_tweets[:5]
```

```
["https://t.co/70HTzk7Q5t \n\n#Lebanon: Traumatized Beirutis ready to 'leave for good' \n\nSince the violent explosion in… https://t.co/fLJGGd3Mfo",
 'Tomorrow night 5th of October at 7.30pm #ABC #Australia will be showing #Lebanon and the Explosion Conspiracy....#beirutexplosion',
 'Can #CFD simulation teach any lessons on the #Lebanon explosion? Read our interview with the engineers who are aimi… https://t.co/yRy50VrFpu',
 '"Welding" caused an explosion in #Iran in an industrial city. The primary theory that the government in #Lebanon is… https://t.co/l507Nrpfil',
 '#Paris is eyeing the contract to rebuild and run #Lebanon’s port after the explosion.\nhttps://t.co/v4UltTLBzn']
```

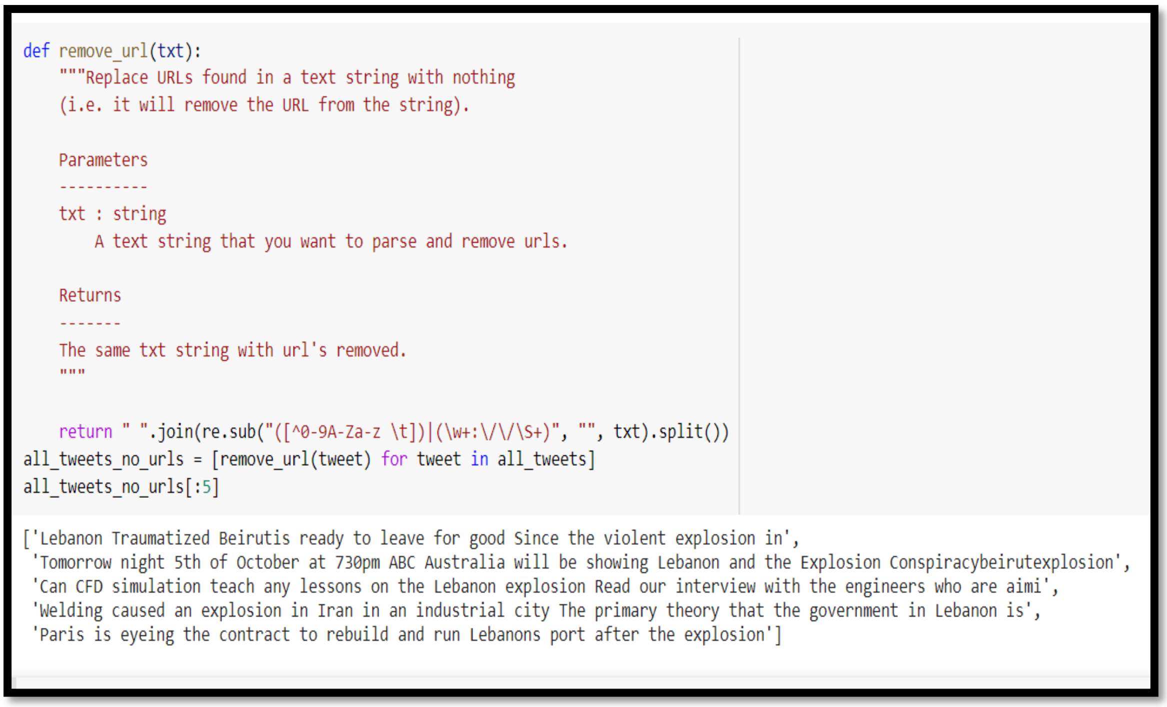
Input:

We have created a search term – Lebanon Explosion, and removed all retweets to get unique data and make our analysis efficient. After that we have created a list of all those "1000" tweets and presented the top 5 tweets.

Output:

The top 5 tweets are shown in the output.

4.Removing the URLs



```
def remove_url(txt):
    """Replace URLs found in a text string with nothing
    (i.e. it will remove the URL from the string).

    Parameters
    ----------
    txt : string
        A text string that you want to parse and remove urls.

    Returns
    -------
    The same txt string with url's removed.
    """

    return " ".join(re.sub("([^0-9A-Za-z \t])|(\w+:\/\/\S+)", "", txt).split())
all_tweets_no_urls = [remove_url(tweet) for tweet in all_tweets]
all_tweets_no_urls[:5]
```

```
['Lebanon Traumatized Beirutis ready to leave for good Since the violent explosion in',
 'Tomorrow night 5th of October at 730pm ABC Australia will be showing Lebanon and the Explosion Conspiracybeirutexplosion',
 'Can CFD simulation teach any lessons on the Lebanon explosion Read our interview with the engineers who are aimi',
 'Welding caused an explosion in Iran in an industrial city The primary theory that the government in Lebanon is',
 'Paris is eyeing the contract to rebuild and run Lebanons port after the explosion']
```

We have created **remove _ url Function** to remove all URLs from 1000 tweets and presented them all by calling that declared function.

Input:

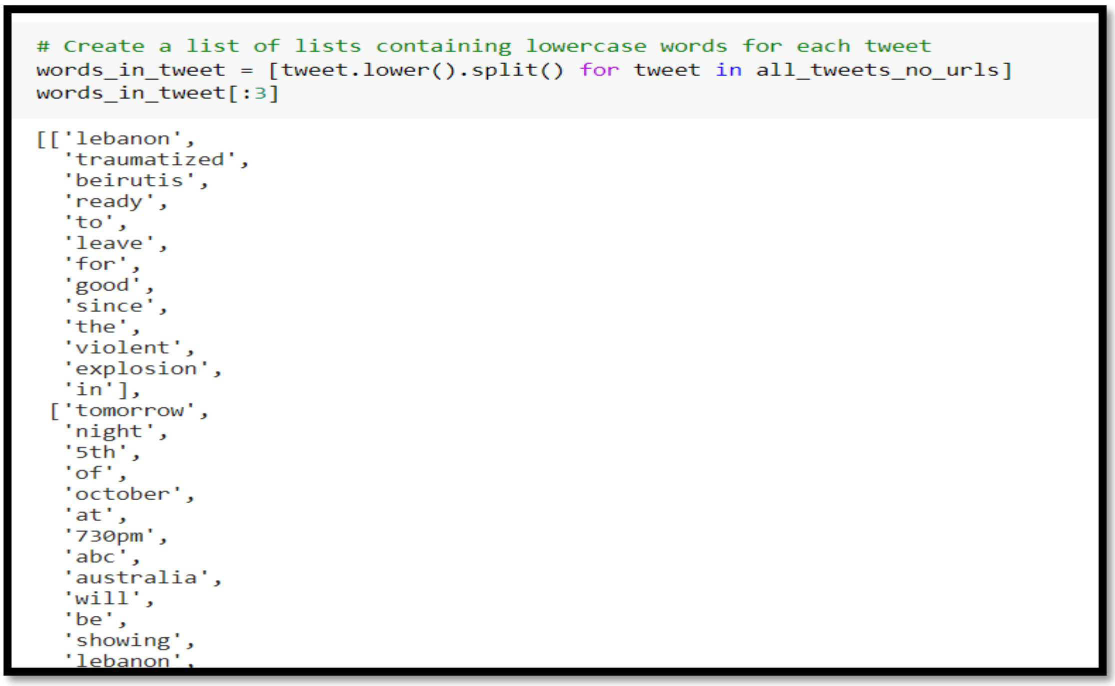
We used the regular expressions package “re” which identifies all string patterns to remove all URLs.

re.sub("([^0-9A-Za-z \t])|(\w+:\/\/\S+)", "", and this line of code says to substitute all strings looking like URLs with “”.

Output:

The top 5 tweets with removed URLs.

5.Lowering the case & splitting the words



```
# Create a list of lists containing lowercase words for each tweet
words_in_tweet = [tweet.lower().split() for tweet in all_tweets_no_urls]
words_in_tweet[:3]
```

```
[['lebanon',
  'traumatized',
  'beirutis',
  'ready',
  'to',
  'leave',
  'for',
  'good',
  'since',
  'the',
  'violent',
  'explosion',
  'in'],
 ['tomorrow',
  'night',
  '5th',
  'of',
  'october',
  'at',
  '730pm',
  'abc',
  'australia',
  'will',
  'be',
  'showing',
  'lebanon',
```

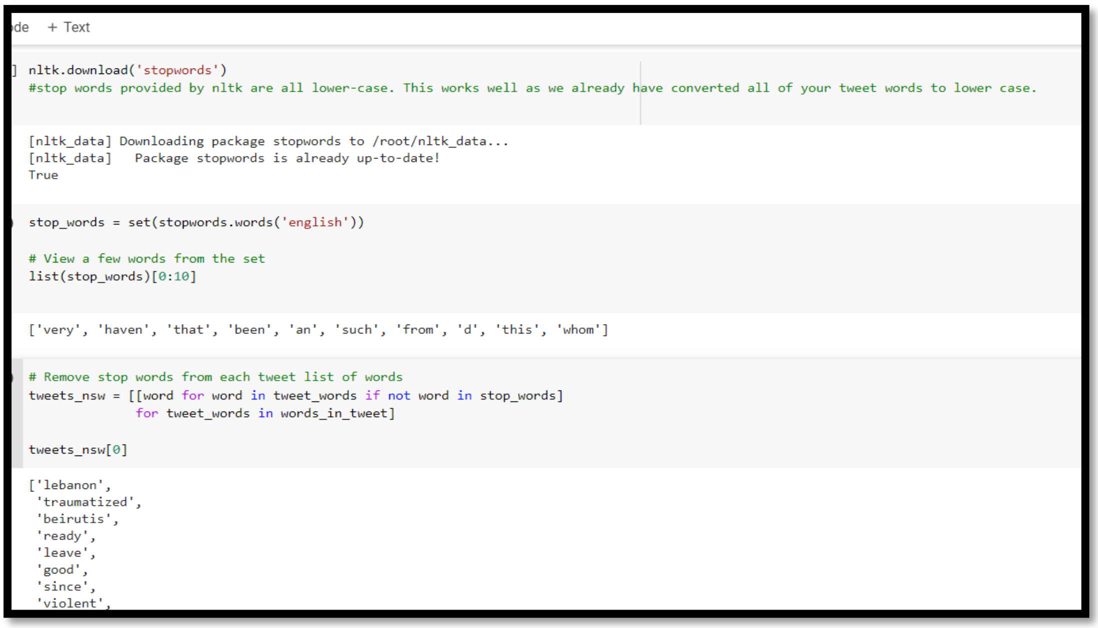
Input:

In order to do frequency analysis ahead we need all unique words list so we have performed lowercase & split function as the same word with capital letters and lowercase letters is treated different in analysis.

Output:

We generated list of unique words converted into lowercase.

6.Removing the stopwords & collection words



```
nltk.download('stopwords')
#stop words provided by nltk are all lower-case. This works well as we already have converted all of your tweet words to lower case.
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
True
```

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

# View a few words from the set
list(stop_words)[0:10]
```

```
['very', 'haven', 'that', 'been', 'an', 'such', 'from', 'd', 'this', 'whom']
```

```
# Remove stop words from each tweet list of words
tweets_nsw = [[word for word in tweet_words if not word in stop_words]
              for tweet_words in words_in_tweet]

tweets_nsw[0]
```

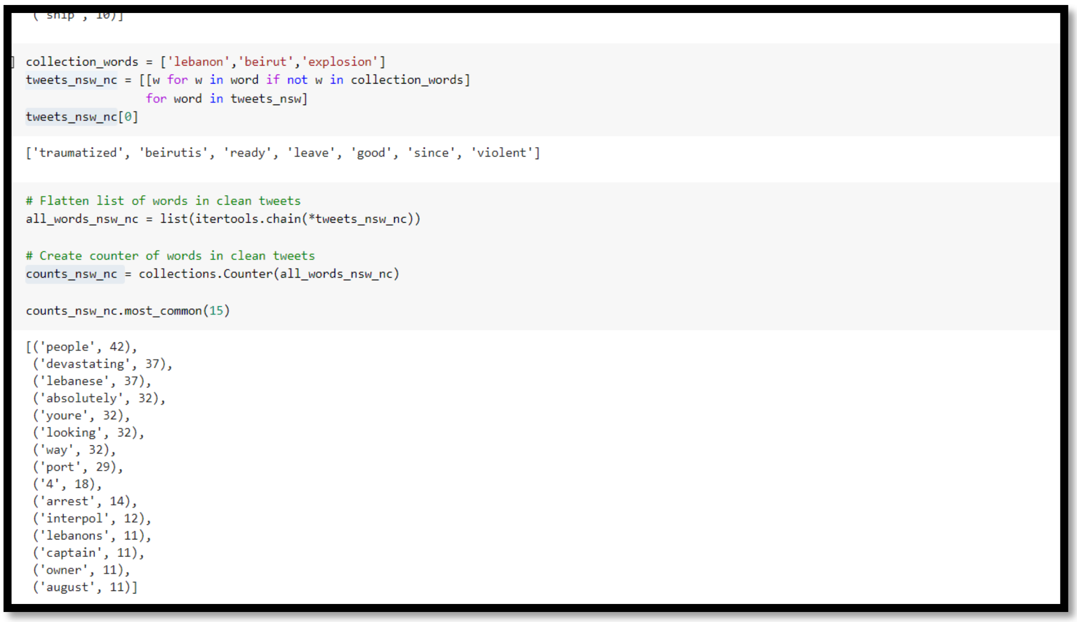
```
['lebanon',
 'traumatized',
 'beirutis',
 'ready',
 'leave',
 'good',
 'since',
 'violent',
```

Input:

The Python package **nltk,** commonly used for text analysis, provides a list of “stop words” that you can use to clean your Twitter data.

Output:

All words list without stopwords in tweet 1.



```
 ('ship', 10)]
```

```
collection_words = ['lebanon','beirut','explosion']
tweets_nsw_nc = [[w for w in word if not w in collection_words]
                 for word in tweets_nsw]
tweets_nsw_nc[0]
```

```
['traumatized', 'beirutis', 'ready', 'leave', 'good', 'since', 'violent']
```

```
# Flatten list of words in clean tweets
all_words_nsw_nc = list(itertools.chain(*tweets_nsw_nc))

# Create counter of words in clean tweets
counts_nsw_nc = collections.Counter(all_words_nsw_nc)

counts_nsw_nc.most_common(15)
```

```
[('people', 42),
 ('devastating', 37),
 ('lebanese', 37),
 ('absolutely', 32),
 ('youre', 32),
 ('looking', 32),
 ('way', 32),
 ('port', 29),
 ('4', 18),
 ('arrest', 14),
 ('interpol', 12),
 ('lebanons', 11),
 ('captain', 11),
 ('owner', 11),
 ('august', 11)]
```

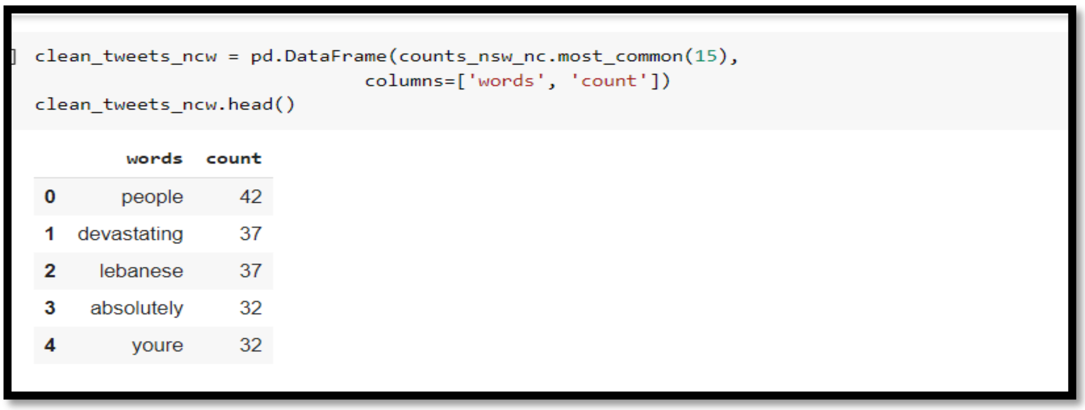
Input:

Collection words are the words that you used to query your data from Twitter. Here our collection words are Lebanon, explosion, Beirut**.** The **collection.Counter** object has a useful built-in method **most_common** that will return the most commonly used words and the number of times that they are used.

Output:

The top 15 most common words used. And tweet 1 with all collection words removed.

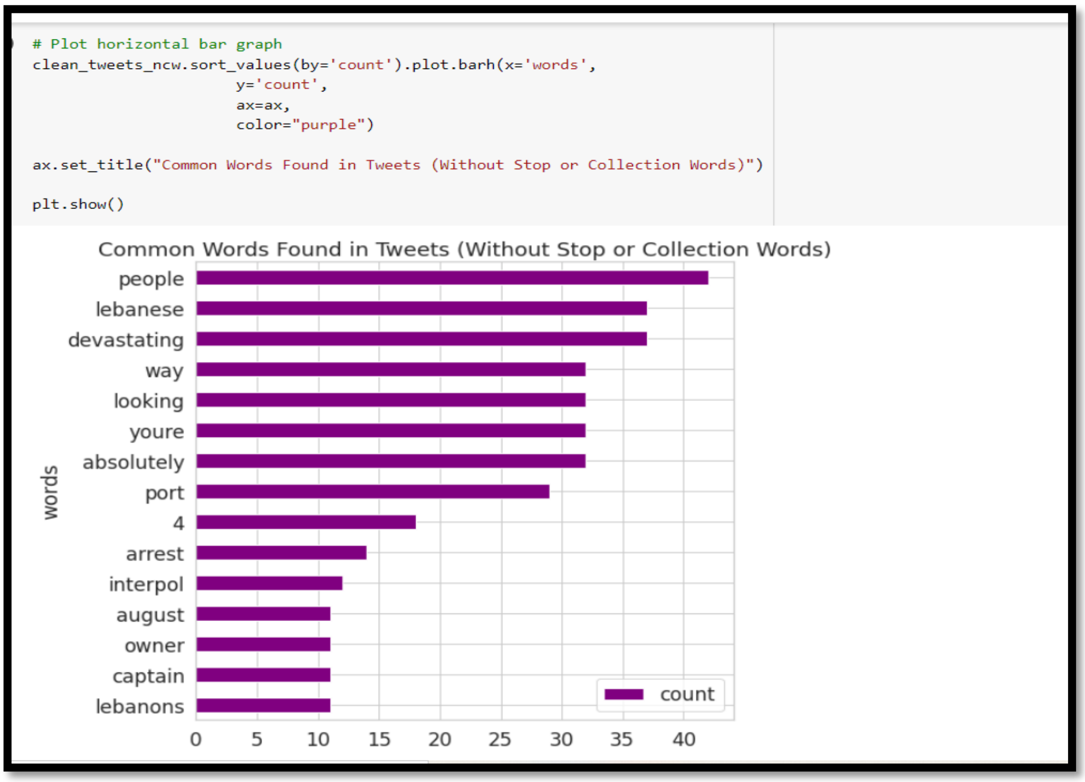
7.Visualization of frequency of words



```
clean_tweets_ncw = pd.DataFrame(counts_nsw_nc.most_common(15),
                                columns=['words', 'count'])
clean_tweets_ncw.head()
```

	words	count
0	people	42
1	devastating	37
2	lebanese	37
3	absolutely	32
4	youre	32

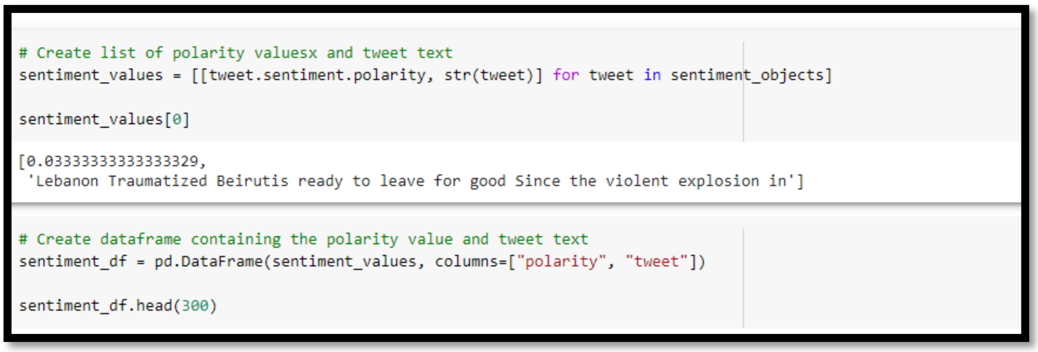
Created the **Pandas Dataframe** of the words and their counts and plot the top 15 most common words from the clean tweets (i.e. no URLs, stop words, or collection words).



```
# Plot horizontal bar graph
clean_tweets_ncw.sort_values(by='count').plot.barh(x='words',
                      y='count',
                      ax=ax,
                      color="purple")

ax.set_title("Common Words Found in Tweets (Without Stop or Collection Words)")

plt.show()
```

This plot displays the frequency of all words in the tweets on climate change, after URLs, stop words, and collection words have been removed

8. Analysing the Sentiments In tweets of Lebanon Explosion



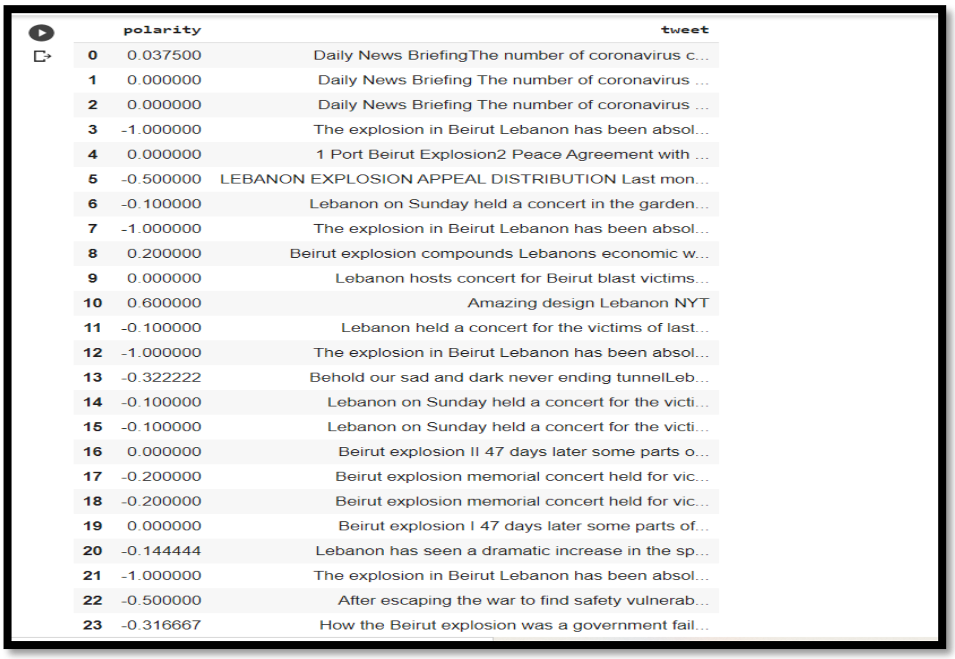
```
# Create list of polarity valuesx and tweet text
sentiment_values = [[tweet.sentiment.polarity, str(tweet)] for tweet in sentiment_objects]

sentiment_values[0]

[0.033333333333333329,
 'Lebanon Traumatized Beirutis ready to leave for good Since the violent explosion in']
```

```
# Create dataframe containing the polarity value and tweet text
sentiment_df = pd.DataFrame(sentiment_values, columns=["polarity", "tweet"])

sentiment_df.head(300)
```

Input: Here we used Python package **Textblob** to calculate the polarity values of individual tweets on climate change. Begin by creating Textblob objects, which assigns polarity values to the tweets. You can identify the polarity value using the attribute polarity of Textblob object.

Polarity is quantification of sentiment with a positive, negative or neutral value. The overall sentiment is often inferred as positive, neutral or negative from the sign of the polarity score which varies from [-1 to 1].

Output:



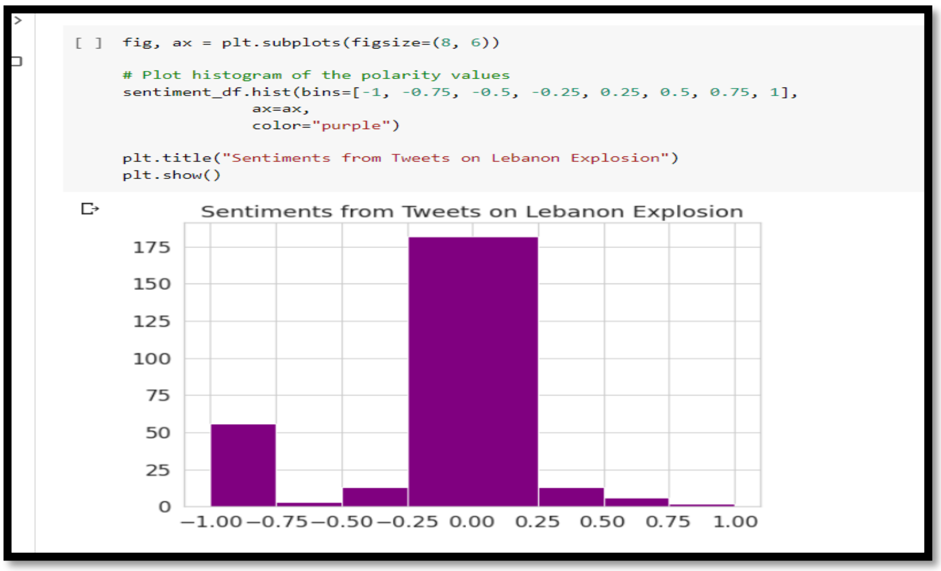
	polarity	tweet
0	0.037500	Daily News BriefingThe number of coronavirus c...
1	0.000000	Daily News Briefing The number of coronavirus ...
2	0.000000	Daily News Briefing The number of coronavirus ...
3	-1.000000	The explosion in Beirut Lebanon has been absol...
4	0.000000	1 Port Beirut Explosion2 Peace Agreement with ...
5	-0.500000	LEBANON EXPLOSION APPEAL DISTRIBUTION Last mon...
6	-0.100000	Lebanon on Sunday held a concert in the garden...
7	-1.000000	The explosion in Beirut Lebanon has been absol...
8	0.200000	Beirut explosion compounds Lebanons economic w...
9	0.000000	Lebanon hosts concert for Beirut blast victims...
10	0.600000	Amazing design Lebanon NYT
11	-0.100000	Lebanon held a concert for the victims of last...
12	-1.000000	The explosion in Beirut Lebanon has been absol...
13	-0.322222	Behold our sad and dark never ending tunnelLeb...
14	-0.100000	Lebanon on Sunday held a concert for the victi...
15	-0.100000	Lebanon on Sunday held a concert for the victi...
16	0.000000	Beirut explosion II 47 days later some parts o...
17	-0.200000	Beirut explosion memorial concert held for vic...
18	-0.200000	Beirut explosion memorial concert held for vic...
19	0.000000	Beirut explosion I 47 days later some parts of...
20	-0.144444	Lebanon has seen a dramatic increase in the sp...
21	-1.000000	The explosion in Beirut Lebanon has been absol...
22	-0.500000	After escaping the war to find safety vulnerab...
23	-0.316667	How the Beirut explosion was a government fail...

This is the output of polarities of individual tweets varying from -1 to 1 where -1 indicates most negative opinion, 0 as neutral and +1 as most positive opinion. We can observe that most of the polarity values are negative as the hashtag tweet selected is concerning to negative emotion i.e. explosion in a city.

9.Visualization of Sentiment Analysis

These polarity values can be plotted in a histogram, which can help to highlight in the overall sentiment (i.e. more positivity or negativity) toward the subject.

I.



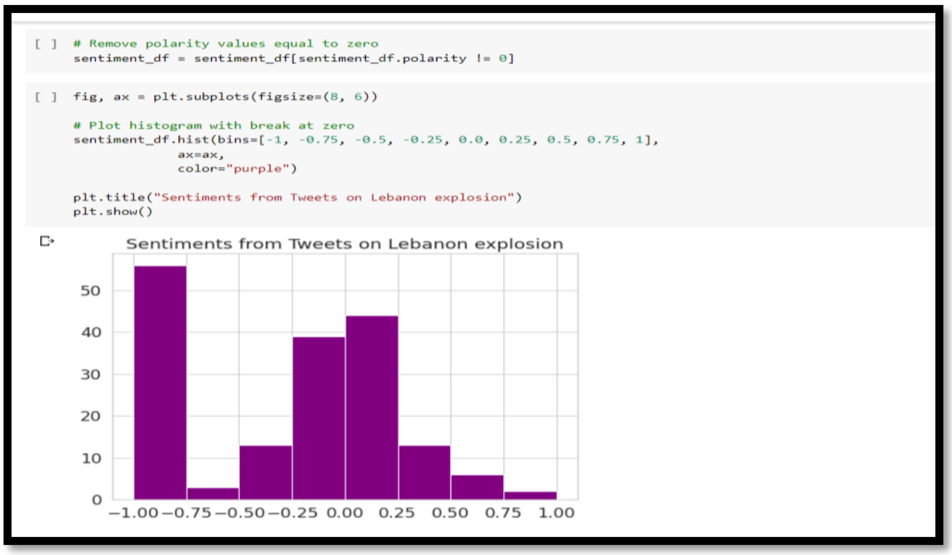
```
[ ]  fig, ax = plt.subplots(figsize=(8, 6))

     # Plot histogram of the polarity values
     sentiment_df.hist(bins=[-1, -0.75, -0.5, -0.25, 0.25, 0.5, 0.75, 1],
                  ax=ax,
                  color="purple")

     plt.title("Sentiments from Tweets on Lebanon Explosion")
     plt.show()
```

So, the above representation has more neutral opinions then negative and then positive.

Approximately 82 tweets are from -1 to -0.25, 190 tweets from -0.25 to +0.25 ,28 tweets from =0.25 to +1.

To get a better visual of the polarity values, it can be helpful to remove the polarity values equal to zero and create a break in the histogram at zero.

II.



```
[ ]  # Remove polarity values equal to zero
     sentiment_df = sentiment_df[sentiment_df.polarity != 0]

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

     # Plot histogram with break at zero
     sentiment_df.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 on Lebanon explosion")
     plt.show()
```

In the above code all the polarities having score of zero is removed and break has been added at zero and again they are plotted on histogram for better distribution of polarity values.

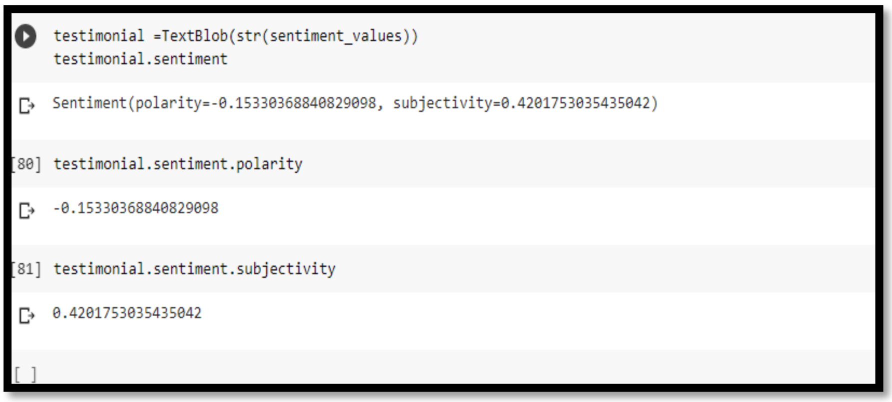
The above total is 180 hence deducting 180 from 300 we get 120 as our neutral comments. From these 180, 115 are negative sentiments and 65 are positive sentiments.

So overall we can conclude that there are

- neutral tweets of about 40%,
- negative sentiment tweets of about 38%
- positive sentiment tweets of about 21%.

Therefore, we can observe that there are more negative sentiments relating to devastation that took place in Beirut in Lebanon and with this incident the government also resigned which escalated the negative emotions among people on twitter.

10.Finding the complete Sentiment



```
testimonial =TextBlob(str(sentiment_values))
testimonial.sentiment

Sentiment(polarity=-0.15330368840829098, subjectivity=0.4201753035435042)

[80] testimonial.sentiment.polarity

-0.15330368840829098

[81] testimonial.sentiment.subjectivity

0.4201753035435042

[ ]
```

Here we find the polarity and subjectivity of the tweets which comes out to be as -0.15 and 0.42 respectively. Here we come to a conclusion clearly that majority of people have negative sentiments towards Lebanon Explosion.