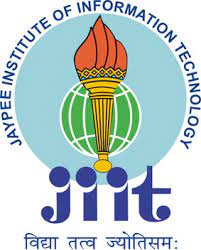
A MINOR PROJECT - I REPORT

ON

**“ISIS TWITTER ANALYSIS”**

****

**SUBMITTED BY:** -

1. DHRUV GOEL-20103067

2. ISHAN CHAUDHARY-20103059

3. SIDDHARTH VISEN-20103080

**UNDER THE SUPERVISION OF**: DEEPTI SINGH

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY, NOIDA**

**1 December, 2022**

**TABLE OF CONTENTS: -**

|  |  |  |
| --- | --- | --- |
| **Chapter No.** | **Topics** | **Page No.** |
|  | **Acknowledgement** | **3** |
|  | **Declaration** | **4** |
|  | **Certificate** | **5** |
| **Chapter-1** | **Introduction** | **6-8** |
| **Chapter-2** | **Software Requirements:** | **8-9** |
| **Chapter-3** | **Key words** | **9-15** |
| **Chapter-4** | **Conclusion** | **16** |
|  | **References** | **17** |

### ACKNOWLEDGEMENT

I would like to place on record my deep sense of gratitude to \_\_\_\_\_\_\_\_\_\_\_\_\_, Designation, Jaypee

Institute of Information Technology, India for his/her generous guidance, help and useful suggestions.

I express my sincere gratitude to \_\_\_\_\_\_\_\_\_\_\_\_\_\_, Dept. of -----------------, India, for his/her

stimulating guidance, continuous encouragement, and supervision throughout the course of present

work.

I also wish to extend my thanks to \_\_\_\_\_\_\_\_\_\_\_\_\_ and other classmates for their insightful comments

and constructive suggestions to improve the quality of this project work.

Signature(s) of Students

1. DHRUV GOEL-20103067

2. ISHAN CHAUDHARY-20103059

3.SIDDHARTH VISEN-20103080

### DECLARATION

We hereby declare that this submission is our own work and that, to the best of our knowledge and beliefs, it contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma from a university or other institute of higher learning, except where due acknowledgment has been made in the text

**Group Details:**

1. DHRUV GOEL-20103067

2. ISHAN CHAUDHARY-20103059

3.SIDDHARTH VISEN-20103080

### CERTIFICATE

This is to certify that the work titled **“ISIS TWITTER ANALYSIS”**

submitted by the group of students of Jaypee Institute of Information Technology, Noida has been carried out under my supervision. This work has not been submitted partially or wholly to any other university or institute for the award of any other degree or diploma.

**Signature of Supervisor**

MR. ANKIT VIDYARTHI

DESIGNATION

DATE

**INTRODUCTION**

* The spread of violent extremism and propaganda is a critical threat both nationally and globally. With the ever-increasing popularity and use of social media, spreading this extremism has never been easier for terrorist organizations and their followers.
* One terrorist organization ISIS (the Islamic State of Iraq and Syria), uses Twitter for the vast majority of their social media interaction.

* It is crucial to have cyber analytics tools developed to combat these extremists' online presence and influence on social media platforms, such as Twitter.
* In this research, we are analyzing their twitting rules and tracking the hidden essential members which helps to predict future attack and avoid them happening.

**PROBLEM STATEMENT**

* As reported, members of ISIS are experts of using social network, especially twitter, a widely-used tool for people all over the world. They mostly use twitter for recruitment and propaganda, attracting thousands of foreigners to join, making the situation of anti-terrorist more severe.
* Understanding the network of pro-ISIS Twitter accounts can help uncover who the primary actors are, how content propagates through the network, and improve the monitoring of ISIS activities.

**MOTIVATION**

As reported, members of ISIS are experts of using social network, especially twitter, a widely-used tool for people all over the world. They mostly use twitter for recruitment and propaganda, attracting thousands of foreigners to join, making the situation of anti-terrorist more severe. Primitively, they created official twitter accounts but are soon banned by internet managers. Now, they are able to be hidden under unofficial accounts and use hashtags for spreading purpose. Analyzing their twitting rules and tracking the hidden essential members are urgent work to predict future attack and avoid them happening.

**SUMMARY OF RESEARCH PAPER STUDIED:**

**PAPER 1:**

In this section, we describe the results of networking experiment and attack prediction, and try to answer the following questions:

1. Who is the Leader in the Network?

In the whole propaganda network, the ranking suggests that which username plays the most important role. The less important users are listed in. As shown in the figure, “ISIS” surpass all the others in sending out and mentioning others, who is the most active one.

1. When and What Do They Mostly Tweet Every Day?

After processing and ranking the hashtags, they frequently used when twitting, we find that, not surprisingly, ‘ISIS’ with its similar form is the most frequent term for attracting people and clustering information. Besides this, ‘Syria’ and ‘Iraq’ are two terms with relative high frequency. That may because most of that attacks and activities happen in their supreme headquarters. We also list some other terms, like ‘Breaking News’, which is mostly followed by a video link in the tweets related to previous attacks. The relative frequency of each tag is depicted in. It shows a seemingly periodic pattern on the tweets number. On each weekend, there seems to be a peak twitting time.

1. Are the Attacks Predictable?

We plot the feature vector, which includes tweets number, tag frequency, and people involvement, also attack severity for visualization purpose, to help us gain some intuitive sense on these limited data. Our result shows the prediction model, especially the dynamic one, with limited data is not perfect, but satisfactory. It successful captures some attack patterns, reminding the public of the time in danger.

**CONCLUSION OF PAPER 1**

In this paper, we analyze the pro-ISIS fanboys network by 17k+ tweets over a year. We firstly construct the propaganda network by extracting user names and their mentioning relationships from tweets, and dig out the most active member by ranking the node importance. To analyze their twitter behaviors, we further summarize the frequency of hashtags and twitting time. Finally, we utilize them to predict potential future attacks by a satisfactory dynamic neural network model. It shows that prediction based on twitter history of ISIS is possible and desirable. In the future work, more emphasis can be put on the twitter contents themselves, and network rapid clustering behaviors across time, to make the prediction more accurate, and save more lives in this planet.

**CHAPTER-2**

**2.1 Software Requirements:**

* PYTHON
* MACHINE LEARNING
* NETWORK X
* KERAS, PANDAS, NUM-PY, SCI-KIT, MATPLOT LIBRARIES
* GOOGLE COLAB

###### **KERAS**

Keras is the most used deep learning framework among top-5 winning teams on [Kaggle](https://www.kaggle.com/). Because Keras makes it easier to run new experiments, it empowers you to try more ideas than your competition, faster. And this is how you win.

###### **PANDAS**

Pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series. It is free software released under the three-clause BSD license.

###### **SCIKIT-LEARN**

Scikit-learn is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support-vector machines

###### **MATPLOTLIB**

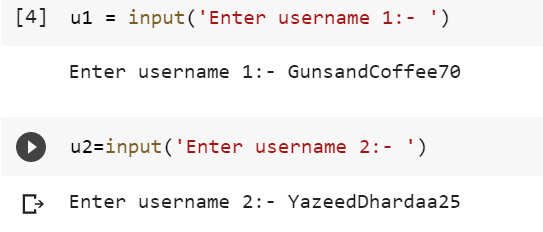
Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK.

**CHAPTER-3**

**METHODOLOGY**

* **COMMON HASHTAGS**

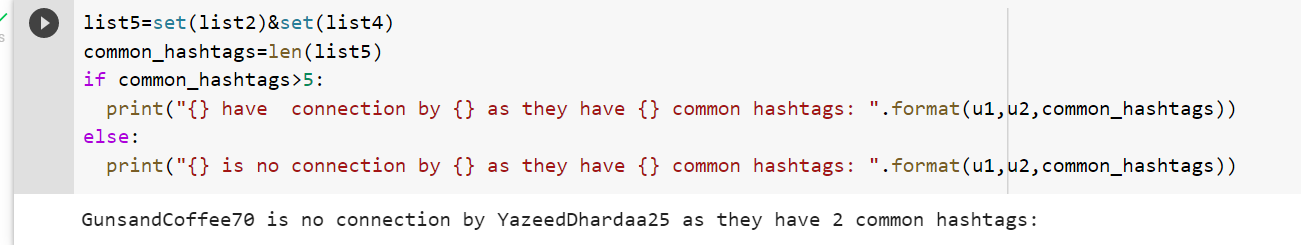
User Name Extraction: In this analysis, we extract the username both in the posting list and the tweets. The method we extract the id, instead of complicated string matching, is localizing the mark of ‘@’ in the tweets. However, some unformatted data, like user names followed by ‘;’ are also filtered in. After extraction, we further guarantee the uniqueness of the username.



Hashtag Extraction: Similar to user name extraction, because of the importance of hashtag when spreading virally, we extract all the hashtags by localizing the mark of ‘#’ in the tweets. Compared to username, the format of hashtags referring to the same content can be more various.



COMMON HASTAGS- LIST 2 EXTRACT THE ALL THE HASHTAGS FROM THE USERNAME 1 AND LIST 4 EXTRACT ALL THE HASHTAGS FROM THE USERNAME 2. HENCE LIST 5 CONTAIN ALL THE COMMON HASHTAGS.

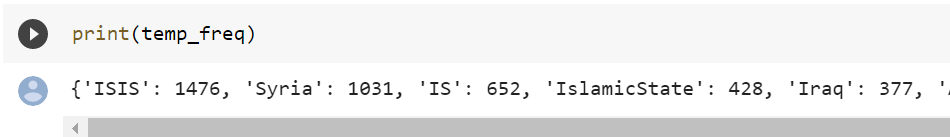


* **Frequency of Hashtags**

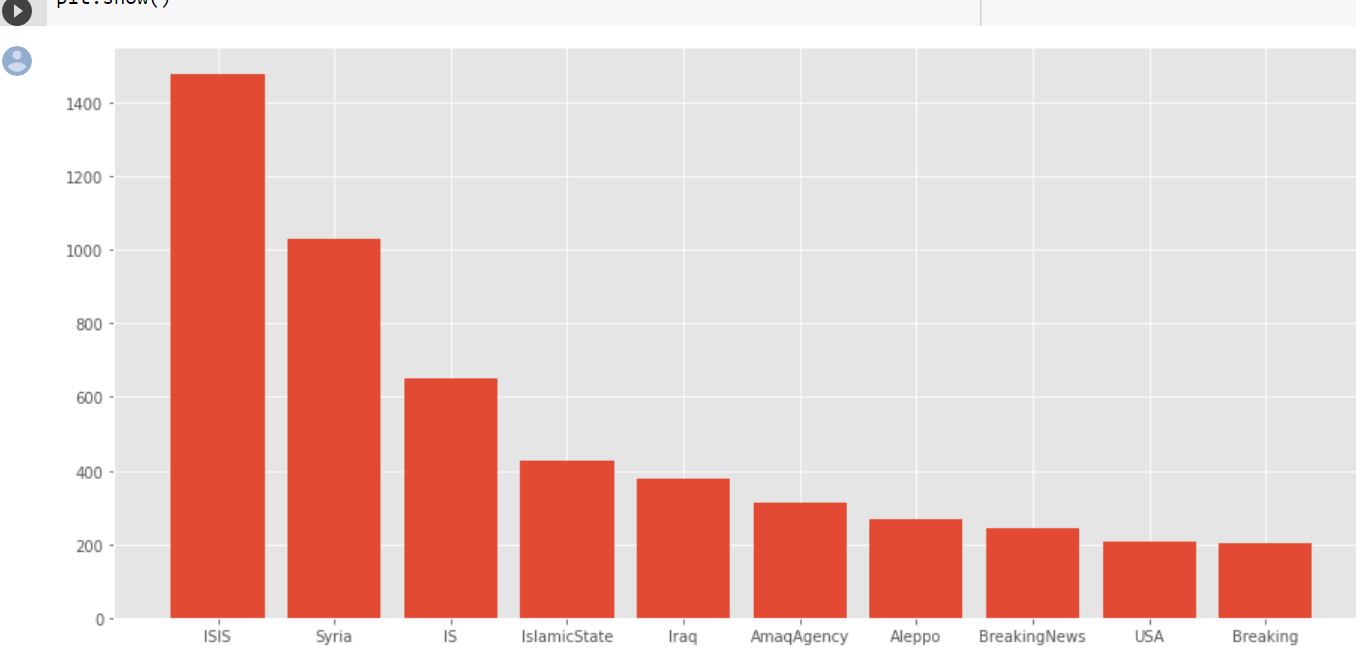
As we discussed, hashtag nowadays become the most effective tools for ISIS members to spread their news and avoid being banned. Hashtags will be able to reveal the potential attack, ISIS ongoing activities and future plans. For example, after nice attack in France, the members of ISIS encourage their followers to hashtag ‘#NiceISIS’ for wide spreading, even skews the twitter tag trend. The method we compute the frequency of hash tag is simply summarizing all the occurrence of unique tag in tweets. The statistics server as, first, an understanding of their tag trend during this time, and secondly, a preparation for analyzing and predicting the attacks.

The information we use for prediction is high-frequency hashtags with their frequency, and high-importance user with their involvement in one-day tweets, as well as the total tweets number for one day. For the hashtags, we rank them by frequency, and select top 7 by thresholding on frequency 7.

Sort them in descending orders.



Plot the graph of top 7 most used hashtags.



* **Word Cloud**

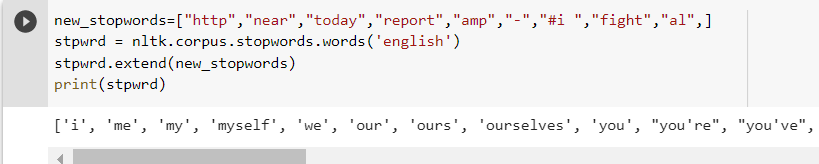
An image composed of words used in a particular text or subject, in which the size of each word indicates its frequency or importance. The visualization of high-frequent hashtags fanboys used for propaganda, showing that ‘ISIS’ is the most frequent one. ‘Syria’ , ‘Iraq’ and ‘IslamicState’ follows, and ‘Breakingnews’ after that. Some important terms like ‘Paris’ are missing in this graph.

Here we have made word cloud we shows of most used hashtags according to their size.



* **Removing Stop words**

A stop word is a commonly used word (such as “the,” “a”, “an”, “in”) that a search engine has been programmed to ignore, both when indexing entries for searching and when retrieving them as the result of a search query.   
We would not want these words to take up space in our database, or taking up valuable processing time. For this, we can remove them easily, by storing a list of words that you consider to stop words. NLTK (Natural Language Toolkit) in python has a list of stopwords stored in 16 different languages.

****

* **Preprocessing**

When doing any Natural Language Processing (NLP) you will need to pre-process your data. There are many things to consider when choosing how to preprocess your text data, but before you do that you will need to familiarize yourself with your data.

In this, we are first removing all English alphabets from all the tweets.

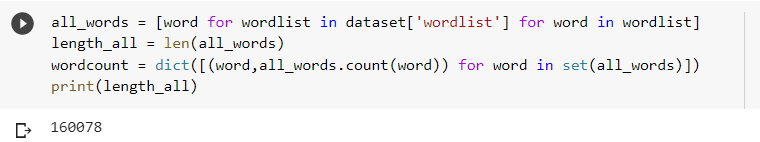
Then we are removing all the https letter word through preprocessing.

**URL links** I did not think URLs would help with sentiment analysis so I wanted to remove them. Removing URLs is not as simple as changing letters to lowercase, it involved using regular expressions (regex). I used expressions, one for URLs with http or https.

**Lowercase**? It is common when doing NLP to lowercase all the words so that a “Hello” is the same as “hello” or “HELLO”. When dealing with tweets, which are not going to follow standard capitalization rules, you should pause before lowercasing everything. For example, it is common to use all caps for emphasis in a tweet when you would rarely do that when writing a formal sentence.

All the words of the tweets are changed to lower case. Hence, all the words are then returned.

Now, the words are counted and hence total number of words are printed.



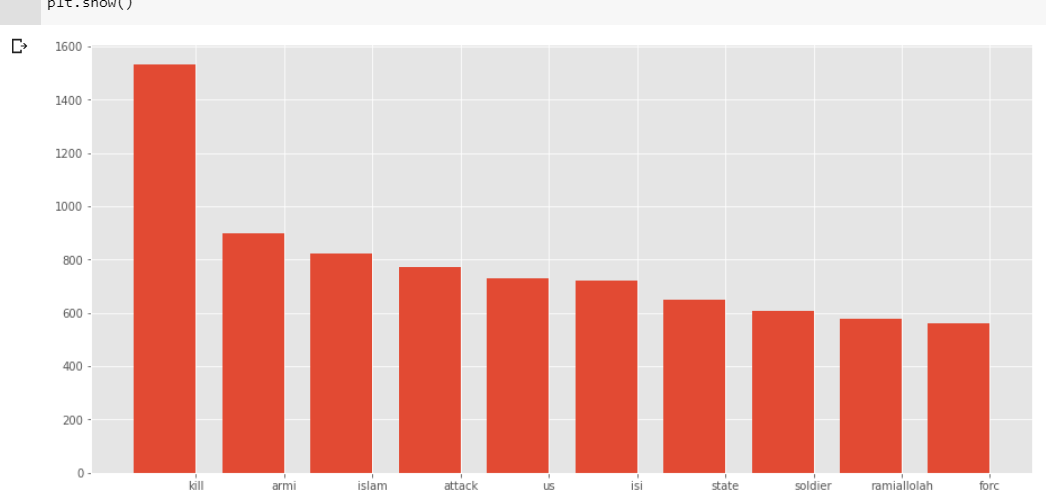
* **Frequency of words**

To get the count of how many times each word appears in the sample, you can use the built-in Python library collections, which helps create a special type of a Python dictonary. 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.

To begin, flatten your list, so that all words across the tweets are in one list. Note that you could flatten your list with another list comprehension like this: all\_words = [item for sublist in tweets\_nsw for item in sublist]

However, it is actually faster to use itertools to flatten the list as follows.

Using this Pandas Dataframe, you can create a horizontal bar graph of the top 10 most common words in the tweets as shown below.

****

* **Popularity based on followers**

Using,

popularity = pd.DataFrame(dataset.groupby('username').followers.max())

we are finding the number of followers each username has.

Hence, we are assigning three data types each of which has some default value input by us.

small=500

mid=2000

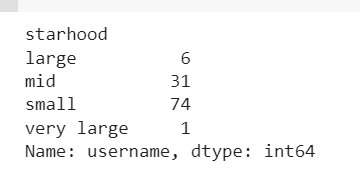
big=10000

Now, from this data types we are finding the popularity of each username

like if username has less than 500 then print “small”.

If it has more than small and less than mid then print “mid.”

Likewise for all username we are traversing and hence finding the popularity.



Now, based on large, mid, small, and very large we are plotting a graph of each username its followers have.

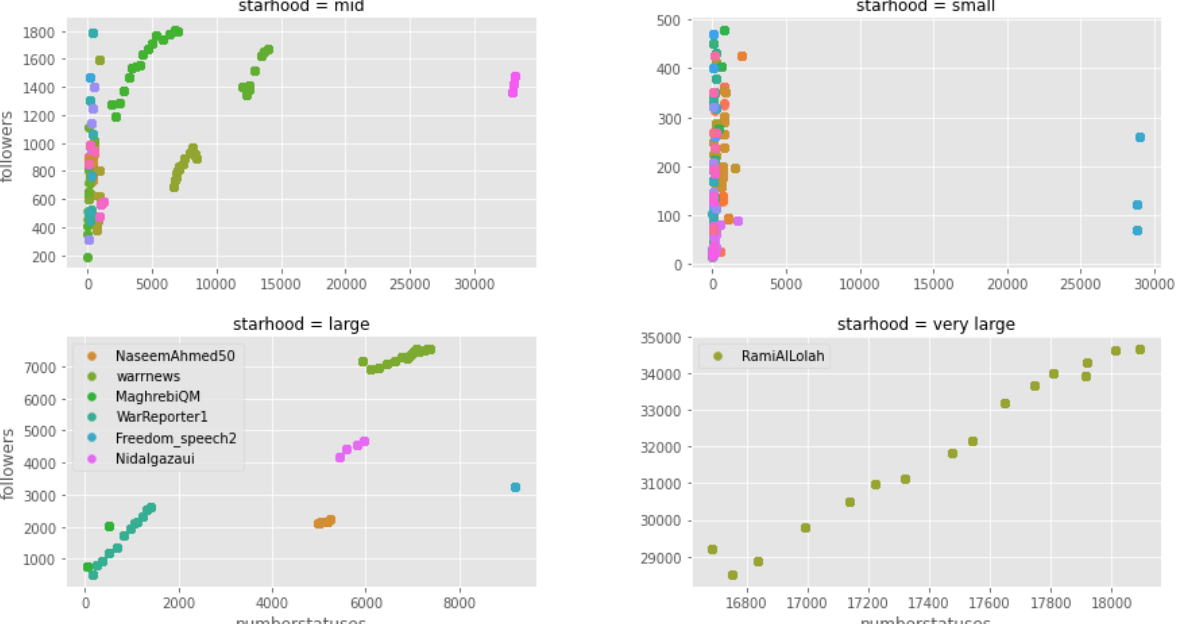
Hence, four graphs are made based on the above category.

Using the below function, we are plotting.

grid = sns.FacetGrid(data=dataset, col='starhood',col\_wrap=2, hue='username', sharex=False, sharey=False)

grid=grid.map(plt.scatter,'numberstatuses','followers')

Below, is the graph for the analyzed data.



* **Locations of tweets**

We are analyzing the location of places from where the tweets are made.

unique\_locations = dataset['location'].unique()

Now, count the locations from how many tweets have been made from that location.

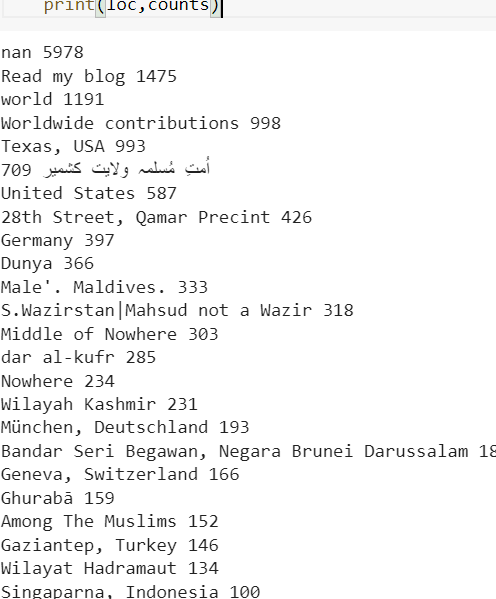
unique\_counts = dict([(loc,list(dataset['location']).count(loc)) for loc in unique\_locations])

unique\_counts = sorted(unique\_counts.items(),key = operator.itemgetter(1))

unique\_counts.reverse()

for (loc,counts) in unique\_counts:

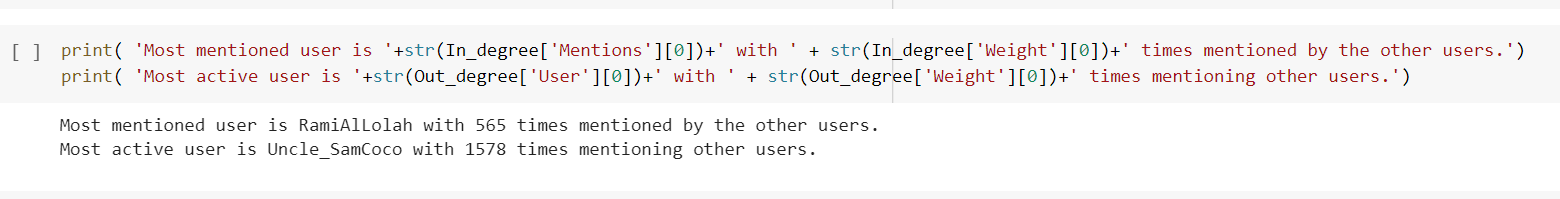
    print(loc,counts)



* **Most Mentioned and Most Active User**

When I search for a tweet containing certain words, I want to check for users mentioned in those tweets that say specifically 'follow @someoneSo' so I could keep track of them and check out their profiles if they seem like relevant accounts, and hopefully, follow some accounts that I like at the end of it.

Hence, we the help of certain libraries we are finding the most mentioned and most active user in dataset.

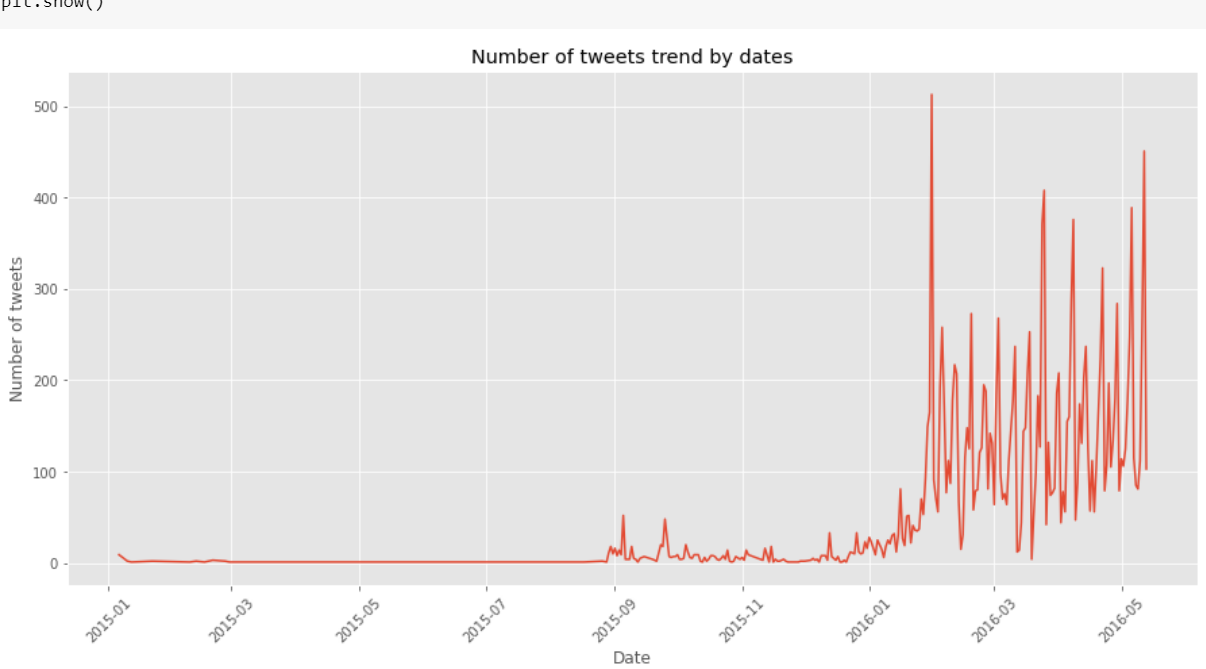


* **No. of tweets trend by dates**

In this no of tweets are plotted on to the following date. We are plotting a graph between no of tweets and date. Date are measured in a gap of month.

plt.xticks(rotation=45)

This function is used to rotate for one month.



We analyze by the graph that number of tweets are basically increasing as we date changes. Since it is not fixed that no of tweets are increasing instead it just gives a idea of tweets. Tweets are not fixed they are fluctuating.

* **Frequency of ISIS tweets in 2016**

In this we are plotting a graph between no of tweets and month of 2016.

Subplot function is used in this graph to plot.

We are also providing two attacks which are happened on that particular date in 2016.

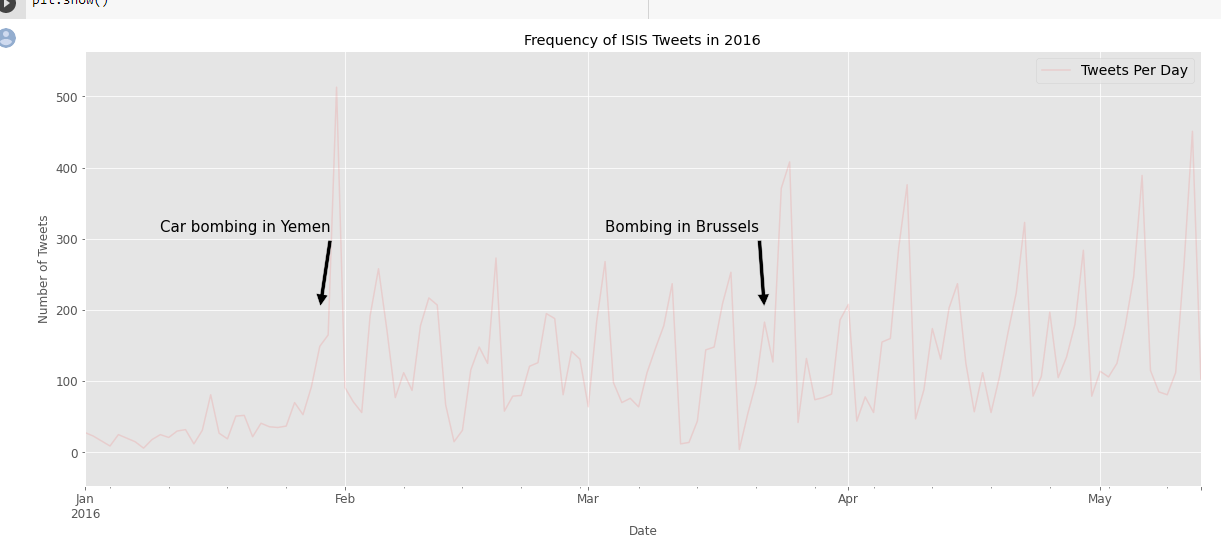
Yemen  = '2016-01-29'

Brussels  = '2016-03-22'

Car bombing in Yemen

Bombing in Brussels

These two attacks get plotted on the graph indicating on this date it happened.



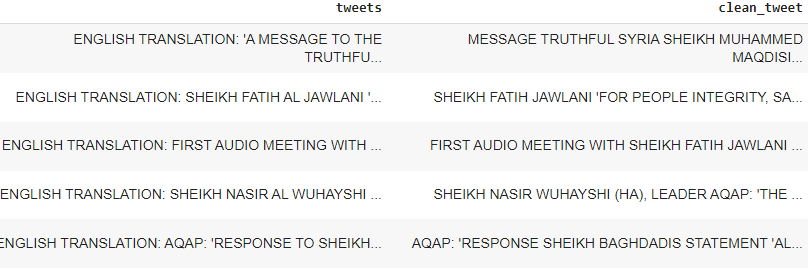
* **Clean Tweets using Preprocessing**

Preprocessing is done which removes all things which are mentioned in the pattern.

In this, we are first removing all mentions start with @someone from all the tweets.

Then we are removing all the https and special characters letter word through preprocessing. After that all English translation words are removed.

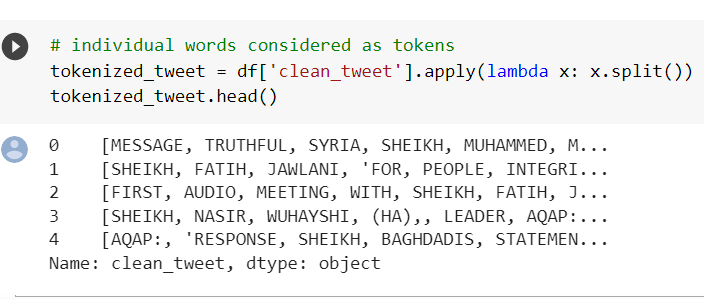
Hence, clear and clean tweet is returned.



* **Tokenization**

In Python tokenization basically refers to splitting up a larger body of text into smaller lines, words or even creating words for a non-English language. The various tokenization functions in-built into the nltk module itself and can be used in programs as shown below.

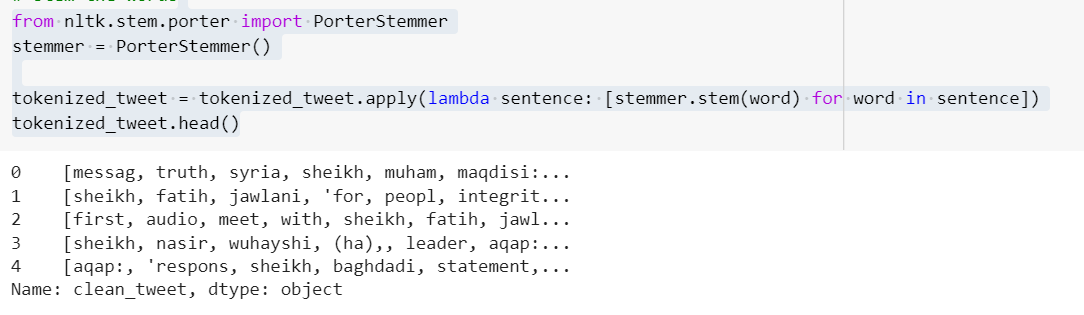
We are tokenizing a tweet into words.

****

* **Porter stemmer**

**Stemming**is the process of producing morphological variants of a root/base word. Stemming programs are commonly referred to as stemming algorithms or stemmers. A stemming algorithm reduces the words “chocolates”, “chocolatey”, and “choco” to the root word, “chocolate” and “retrieval”, “retrieved”, “retrieves” reduce to the stem “retrieve”.

Now we are stremming our words received through tokenization.



* **Combining words**

Now after tokenization and stremming of words received we are combining them into a single sentence for every user.

This is done using join function.

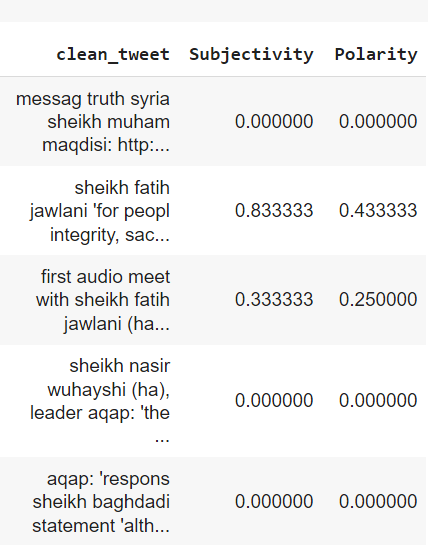


* **Subjectivity and Polarity**

TextBlob is a python library for Natural Language Processing (NLP).TextBlob actively used Natural Language ToolKit (NLTK) to achieve its tasks. NLTK is a library which gives an easy access to a lot of lexical resources and allows users to work with categorization, classification and many other tasks. TextBlob is a simple library which supports complex analysis and operations on textual data.

TextBlob returns **polarity** and **subjectivity** of a sentence. Polarity lies between [-1,1], -1 defines a negative sentiment and 1 defines a positive sentiment. Negation words reverse the polarity. TextBlob has semantic labels that help with fine-grained analysis. For example — emoticons, exclamation mark, emojis, etc. Subjectivity lies between [0,1]. **Subjectivity quantifies the amount of personal opinion and factual information contained in the text. The higher subjectivity means that the text contains personal opinion rather than factual information**. TextBlob has one more parameter — intensity.

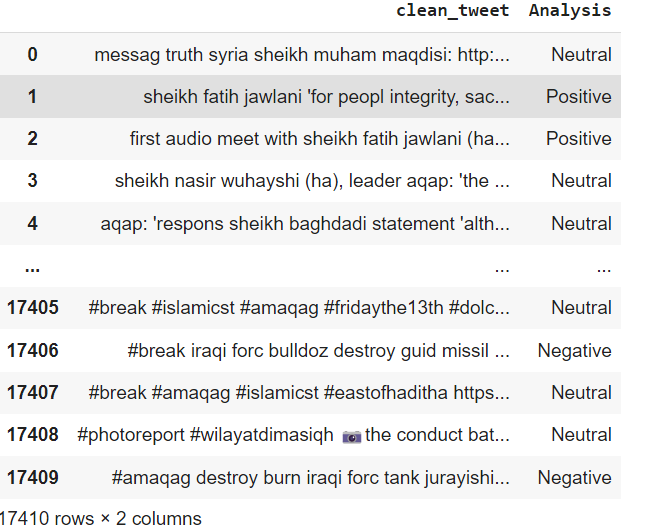
Now we apply both subjectivity and polarity on clean tweets which we have obtained above.

****

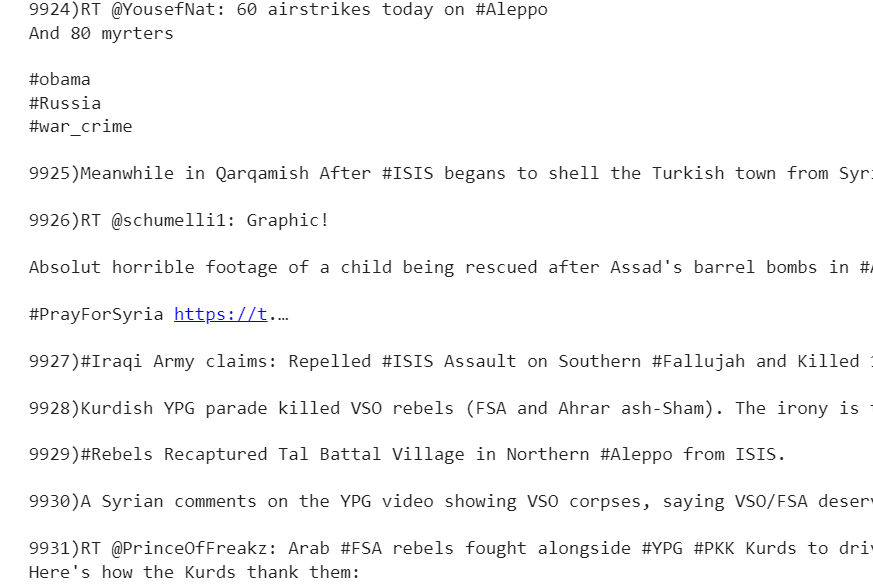
* **Sentiment of Tweet**

Sentiment Analysis can help us decipher the mood and emotions of general public and gather insightful information regarding the context. Sentiment Analysis is a process of analyzing data and classifying it based on the need of the research.

We have to analyze whether the tweet polarity obtained above is positive, negative or neutral. A tweet is positive is when its polarity is more than 0 and negative when less than 0. It is neutral when it is equal to 0.



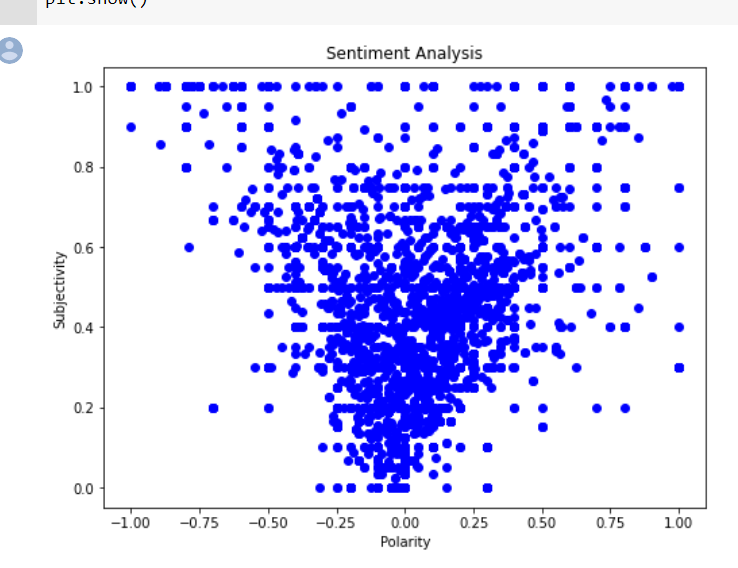
Now we are returning that tweet which is entered by the user whether neutral, positive or negative.



* **Graph based on subjectivity and polarity**

Now we are plotting a graph between subjectivity and polarity of tweet which we already had found above .

Title for which is Sentiment anlaysis.



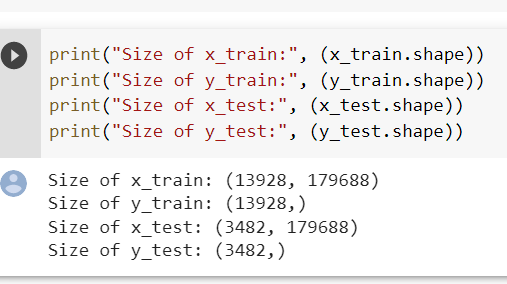
Tweet below 0 shows negative and above 0 shows positive.

* **Count Vectorizer**

**CountVectorizer**is a great tool provided by the scikit-learn library in Python. It is used to transform a given text into a vector on the basis of the frequency (count) of each word that occurs in the entire text. This is helpful when we have multiple such texts, and we wish to convert each word in each text into vectors (for using in further text analysis).

We are finding the values of x\_train, y\_train, x\_test and y\_test using countvectorizer.

Using [train\_test\_split()](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html) from the data science library [scikit-learn](https://scikit-learn.org/stable/index.html), you can split your dataset into subsets that minimize the potential for bias in your evaluation and validation process.



* **Confusion Matrix**

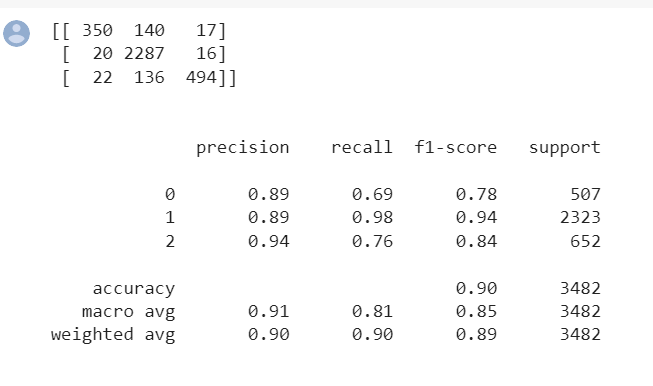
It is a table that is used in classification problems to assess where errors in the model were made.

The rows represent the actual classes the outcomes should have been. While the columns represent the predictions we have made. Using this table it is easy to see which predictions are wrong.

**Precision**

precision=(TP)/(TP+FP)  
TP is the number of true positives, and FP is the number of false positives.   
A trivial way to have perfect precision is to make one single positive prediction and ensure it is correct (precision = 1/1 = 100%).

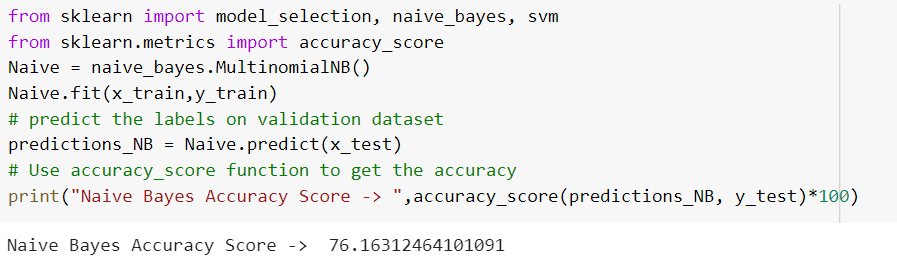
**Recall**  
recall = (TP) / (TP+FN)



**Naïve Base Accuracy Technique**

Naive Bayes is a statistical classification technique based on Bayes Theorem. It is one of the simplest supervised learning algorithms. Naive Bayes classifier is the fast, accurate and reliable algorithm. Naive Bayes classifiers have high accuracy and speed on large datasets.

Naive Bayes classifier assumes that the effect of a particular feature in a class is independent of other features.



* **SVM**

**Support Vector Machine (SVM)** is a simple, supervised machine learning algorithm. SVMs are used for both classification and regression problems.

In the SVM algorithm, each point is represented as a data item within the n-dimensional space where the value of each feature is the value of a specific coordinate.

After plotting, classification has been performed by finding hype-plane which differentiates two classes. Refer below image to understand this concept.

