TITLE : NATURAL LANGUAGE PROCESSING SENTIMENT ANALYSIS OF TWEETS

By Uditi Namdev

TE – Computers

INTRODUCTION

Natural language processing (NLP) is a subfield of linguistics, computer science, and artificial intelligence concerned with the interactions between computers and human language, in particular how to program computers to process and analyse large amounts of natural language data.

It is an area of growing attention due to increasing number of applications like chatbots, machine translation etc. In some ways, the entire revolution of intelligent machines in based on the ability to understand and interact with humans.

Sentiment analysis is the automated process of identifying and extracting the subjective information that underlies a text. This can be either an opinion, a judgment, or a feeling about a particular topic or subject. The most common type of sentiment analysis is called 'polarity detection' and involves classifying a statement as 'positive', 'negative', or 'neutral'. It is also the sub-field of Natural Language Processing (NLP) and has been getting a lot of attention in recent years due to its diverse Business applications.

In the past decade, new forms of communication, such as microblogging and text messaging have emerged and become ubiquitous. While there is no limit to the range of information conveyed by tweets and texts, often these short messages are used to share opinions and sentiments that people have about what is going on in the world around them. Nearly 80% of the world's digital data is unstructured, and a large portion of that includes social media data. By analysing social media posts, product reviews, customer feedback, and NPS responses (among other unstructured data), businesses can understand how their customers feel about their product or service. Using sentiment analysis tools to analyse opinions in Twitter data can help companies understand how people are talking about their brand.

Many traditional approaches in sentiment analysis uses the bag of words method. The bag of words technique does not consider language morphology, and it could incorrectly classify two phrases of having the same meaning because it could have the same bag of words. The relationship between the collection of words is considered instead of the relationship between individual words. When determining the overall sentiment, the sentiment of each word is determined and combined using a function. Bag of words also ignores word order, which leads to phrases with negation in them to be incorrectly classified. Other techniques discussed in sentiment analysis include Naive Bayes, Maximum Entropy, and Support Vector Machines.

How is Sentiment analysis useful?

Sentiment analysis has been useful for companies to get their customer's opinions on their products predicting outcomes of elections, and getting opinions from movie reviews. The information gained from sentiment analysis is useful for companies making future decisions.

According to some researchers, Sentiment Analysis of Twitter data can help in the prediction of stock market movements. Research show that news articles and social media can hugely influence the stock market. News with overall positive sentiment has been observed to relate to a large increase in price albeit for a short period of time. On the other hand, negative news is seen to be linked to a decrease in price – but with more prolonged effects.

OBJECTIVE

This project aims to perform Sentiment Analysis on Tweets that we can fetch from twitter API. Here we will try to implement this by using the TextBlob python library, which is built on the shoulders of NLTK (Natural Language Toolkit) and Pattern. TextBlob makes text processing simple by providing an intuitive interface to NLTK. TextBlob is a python library that offers a simple API to access its methods and perform basic NLP tasks. A big advantage of this is, it is easy to learn and offers a lot of features like sentiment analysis, POS-tagging, noun phrase extraction, etc

APPLICATIONS

There are various applications of Sentiment analysis, two of them are:

Social Media Monitoring: In today's world, the vast majority of social media tools have some kind of sentiment analysis capabilities. Companies also use automated sentiment analysis based on word lists, with each word being given a pre-defined sentiment value. The value of the text is then determined based on the words it contains. This has a number of different uses. For example, a restaurant might engage in social media monitoring in order to gauge how people feel about their menu, figure out whether or not people enjoyed their food, and ascertain what feelings people associated with their overall experience at the restaurant. And the good news is that the accuracy rates of sentiment analysis for social media monitoring keep getting better. Sentiment analysis companies achieve accuracy rates of over 75 percent with its automated sentiment analysis capabilities.

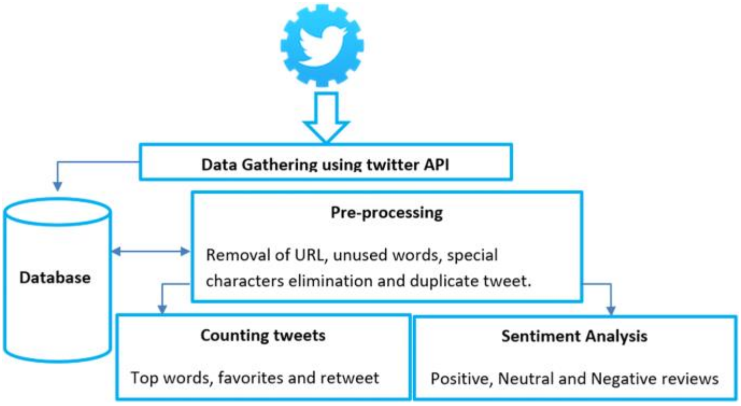
Data Mining: Sentiment analysis can also be used for data mining, or gathering competitive intelligence about your competitors. For example, a brand could easily track social media mentions or mentions of competitors in other places across the web, and analyse how consumers feel about the competitors and their products. This is an excellent way to gain a competitive edge in today's highly competitive marketplace.

METHOD

Twitter Sentiment Analysis can be carried out by various methods: Lexicon or Machine Learning methods.

There are several steps involved for the same:

1. Gathering Twitter Data
2. Preparing the Data
3. Creating a Sentiment Analysis Model
4. Plotting graphs and visualising the Results accordingly


Data Gathering using twitter API
Database
Pre-processing
Removal of URL, unused words, special characters elimination and duplicate tweet.
Counting tweets
Top words, favorites and retweet
Sentiment Analysis
Positive, Neutral and Negative reviews

To gather the data, we will be incorporating the use of Twitter APIs. The Twitter API (the term stands for Application Programming Interface) enables software developers to access and interact with public Twitter data. Developers can interact with this API by writing their own scripts or by using one of the open source libraries available in different programming languages. To make any request to the Twitter API, you must first apply for a developer account, and have your use case approved. Once approved, you can create a Project and connect an associated developer App which will provide you a set of credentials that you will use to authenticate all requests to the API. There are many open source libraries that you can use to connect with the Twitter API. Tweepy is one of the the famous ones.

Basic methodology for this project would be:

1. Import Tweepy for creating the connection with Twitter API.

2. Fetch tweets as dataset or type the keyword related to some tweet.

3. Pre-processing of tweets by removing the stop words, punctuations, hashtags, etc.

4. Tokenize each word in the dataset and save it into the dataset.

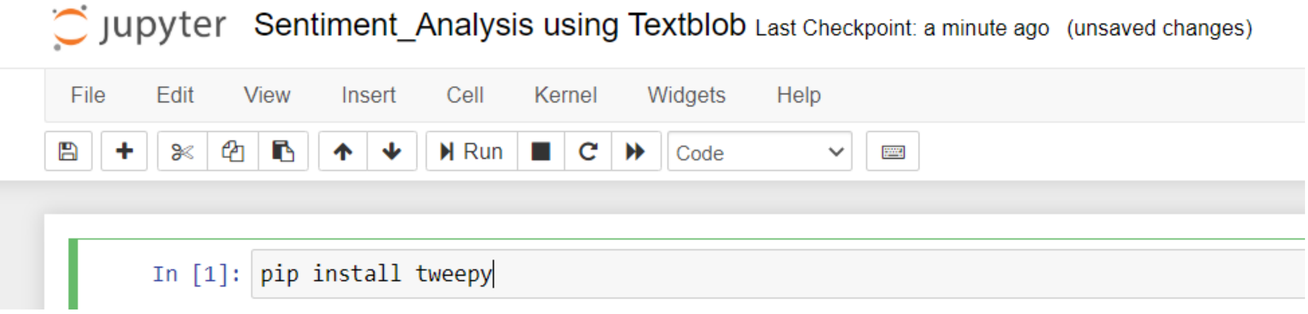
5. For each word, compare it with positive, negative and neutral sentiments word in the dictionary. Then increment the positive, negative and neutral count.

6. Finally, based on the positive, negative and neutral count, we can get the result percentage about sentiment to decide the polarity.

IMPLEMENTATION

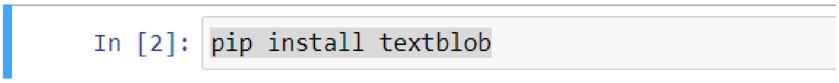
The required libraries are installed by following commands:

Tweepy: This package will be used for handling the Twitter API.

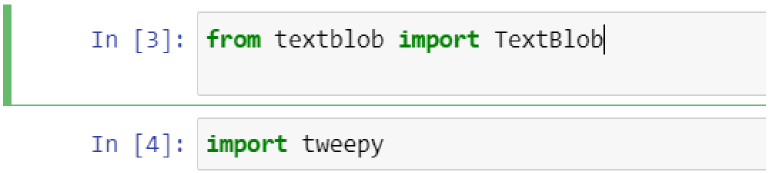

jupyter
Sentiment_Analysis using Textblob Last Checkpoint: a minute ago (unsaved changes)
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Run
Code

```
In [1]: pip install tweepy
```

Textblob : This package will be used for the sentiment analysis.



```
In [2]: pip install textblob
```

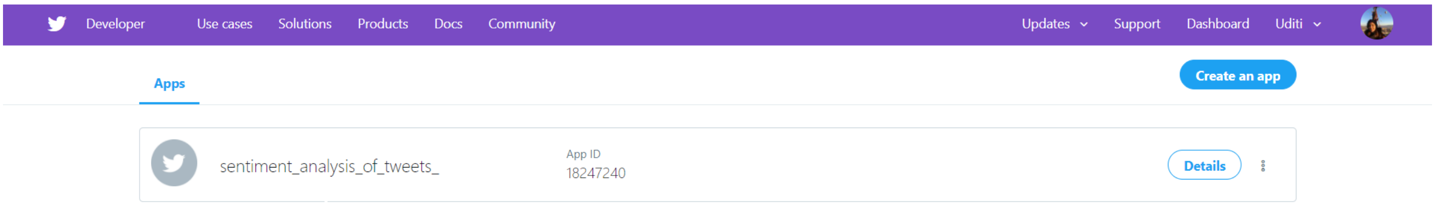


```
In [3]: from textblob import TextBlob

In [4]: import tweepy
```

In order to fetch tweets through Twitter API, one needs to register an App through their twitter account.

The steps I followed are:

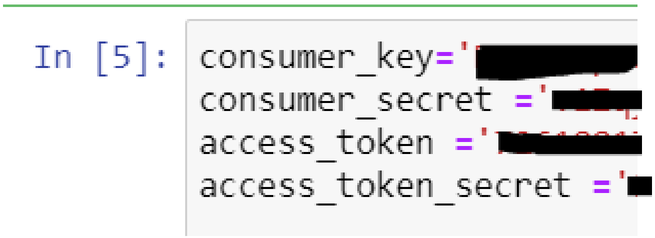
1) Open this link: https://developer.twitter.com/en/apps and click on the button: 'Create New App'
2) Fill the application details.
3) Once the app is created, we'll be redirected to the app page.
4) Open the 'Keys and Access Tokens' tab.
5) Copy 'Consumer Key', 'Consumer Secret', 'Access token' and 'Access Token Secret'. There is also an option to regenerate the keys.



sentiment_analysis_of_tweets_

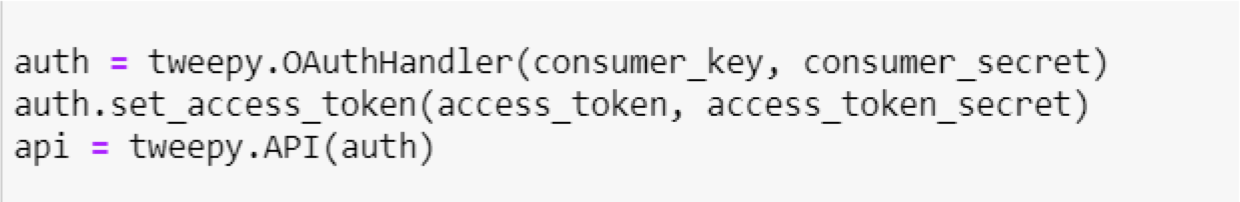
App ID
18247240

We need to declare the variables to store the various keys associated with the Twitter API.



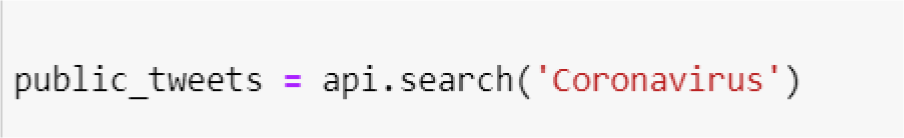
```
In [5]: consumer_key='[redacted]
        consumer_secret ='[redacted]
        access_token ='[redacted]
        access_token_secret ='[redacted]
```

The next step is to create a connection with the Twitter API using tweepy with these tokens. Twitter API uses OAuth, which is an open authorization protocol to authenticate requests.
Tweepy supports OAuth authentication. Authentication is handled by the tweepy.OAuthHandler class.
An OAuthHandler instance must be created by passing a consumer token and secret.
On this auth instance, we will call a function set_access_token by passing the access_token and access_token_secret. Finally, we create our tweepy API instance by passing this auth instance into the API function of tweepy.



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

We can now search Twitter for any topic using the search method of the API. This method returns a collection of relevant Tweets matching a specified query for all public tweets.



```
public_tweets = api.search('Coronavirus')
```

Hence, we will get all the tweets related to 'Coronavirus' and can perform sentiment analysis on them.

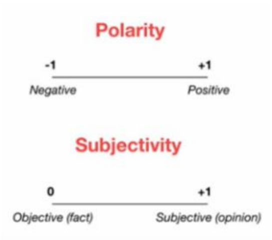
TextBlob provides a simple API for diving into common natural language processing (NLP) tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and more.

The sentiment property returns a named tuple of the form Sentiment (polarity, subjectivity). The polarity score is a float within the range [-1.0, 1.0]. The subjectivity is a float within the range [0.0, 1.0] where 0.0 is very objective or like a fact and 1.0 is very subjective and based on personal opinion.

Basically,

Polarity: How positive or negative a word is. -1 is very negative. +1 is very positive.

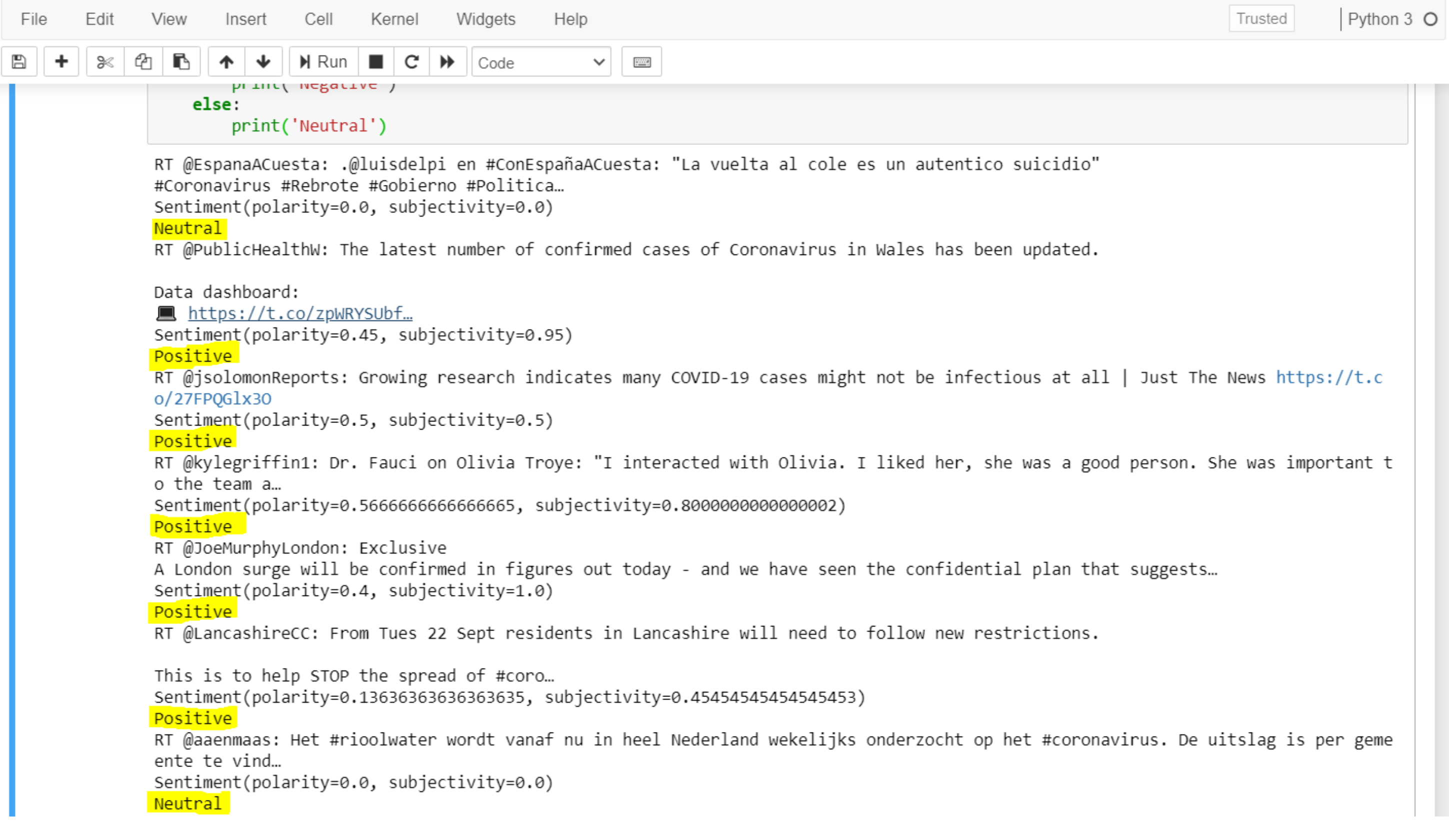
Subjectivity: How subjective, or opinionated a word is. 0 is fact. +1 is very much opinion.



Now, we can iterate the public_tweets array and find the polarity of the tweets and classify them as positive, negative or neutral.

```
for tweet in public_tweets:
    print(tweet.text)
    analysis = TextBlob(tweet.text)
    print(analysis.sentiment)
    if analysis.sentiment[0]>0:
        print('Positive')
    elif analysis.sentiment[0]<0:
        print('Negative')
    else:
        print('Neutral')
```

Therefore, when we run this code, we get the following results:



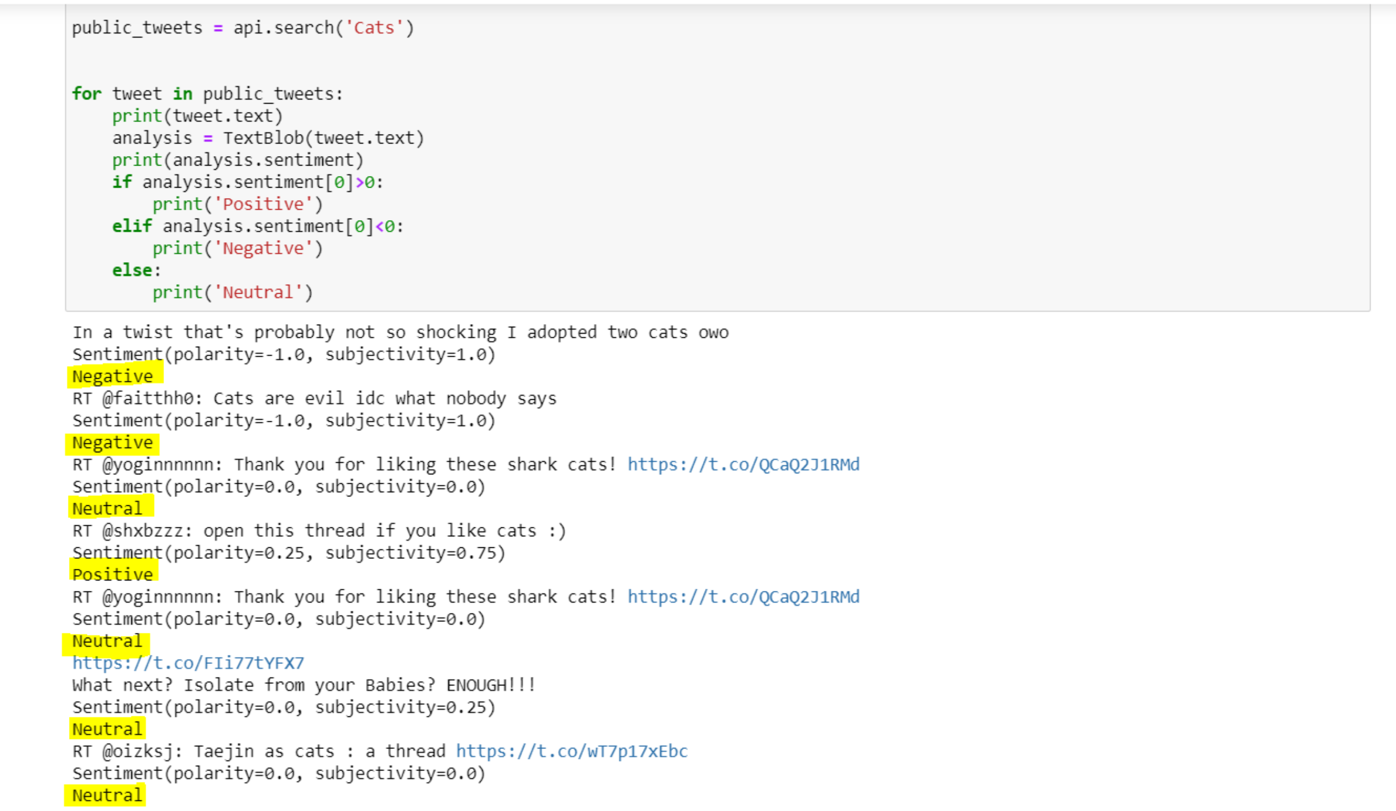
```
        print('Negative')
    else:
        print('Neutral')
```

```
RT @EspanaACuesta: .@luisdelpi en #ConEspañaACuesta: "La vuelta al cole es un autentico suicidio"
#Coronavirus #Rebrote #Gobierno #Politica…
Sentiment(polarity=0.0, subjectivity=0.0)
Neutral
RT @PublicHealthW: The latest number of confirmed cases of Coronavirus in Wales has been updated.

Data dashboard:
💻 https://t.co/zpWRYSUbf…
Sentiment(polarity=0.45, subjectivity=0.95)
Positive
RT @jsolomonReports: Growing research indicates many COVID-19 cases might not be infectious at all | Just The News https://t.c
o/27FPQGlx3O
Sentiment(polarity=0.5, subjectivity=0.5)
Positive
RT @kylegriffin1: Dr. Fauci on Olivia Troye: "I interacted with Olivia. I liked her, she was a good person. She was important t
o the team a…
Sentiment(polarity=0.5666666666666665, subjectivity=0.8000000000000002)
Positive
RT @JoeMurphyLondon: Exclusive
A London surge will be confirmed in figures out today - and we have seen the confidential plan that suggests…
Sentiment(polarity=0.4, subjectivity=1.0)
Positive
RT @LancashireCC: From Tues 22 Sept residents in Lancashire will need to follow new restrictions.

This is to help STOP the spread of #coro…
Sentiment(polarity=0.13636363636363635, subjectivity=0.45454545454545453)
Positive
RT @aaenmaas: Het #rioolwater wordt vanaf nu in heel Nederland wekelijks onderzocht op het #coronavirus. De uitslag is per geme
ente te vind…
Sentiment(polarity=0.0, subjectivity=0.0)
Neutral
```

So here, there is a tweet: 'Growing research indicates many Covid-19 cases might not be infectious at all. This tweet was rightly analysed as 'Positive'.

One more example:

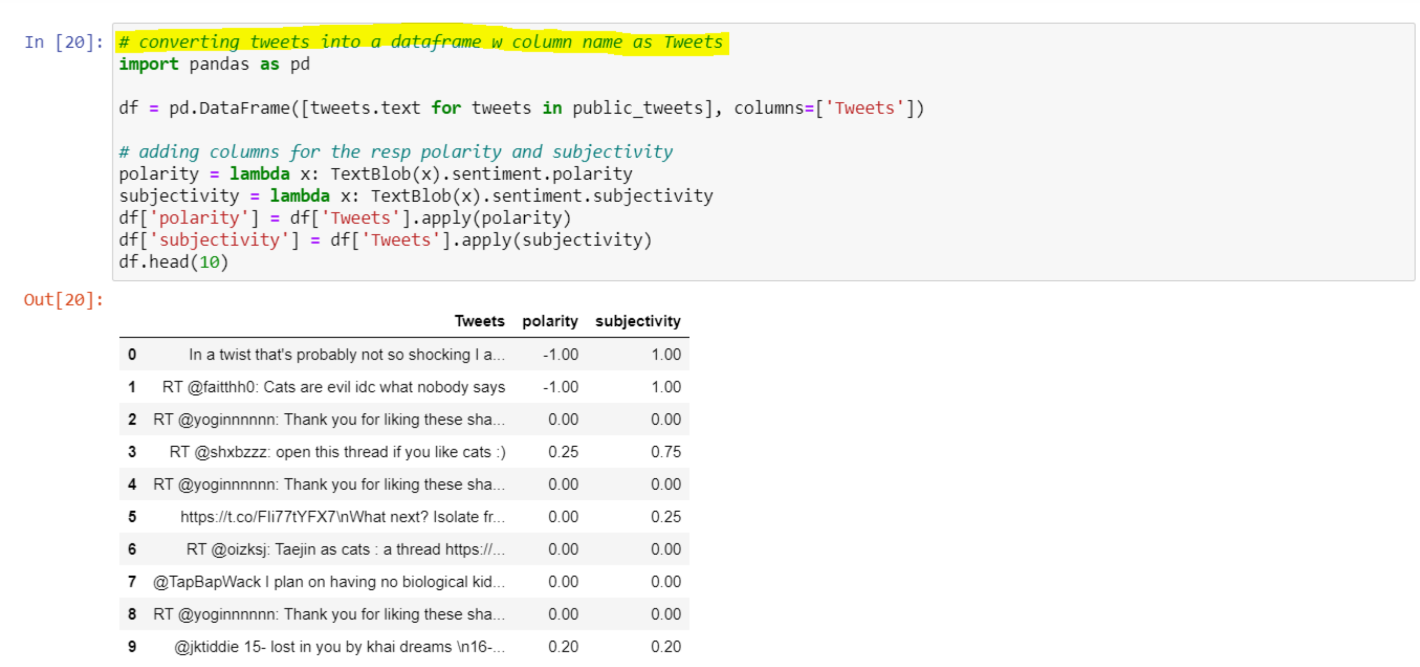
When 'Cats' is searched the relevant tweets that come up are :



```
public_tweets = api.search('Cats')


for tweet in public_tweets:
    print(tweet.text)
    analysis = TextBlob(tweet.text)
    print(analysis.sentiment)
    if analysis.sentiment[0]>0:
        print('Positive')
    elif analysis.sentiment[0]<0:
        print('Negative')
    else:
        print('Neutral')
```

```
In a twist that's probably not so shocking I adopted two cats owo
Sentiment(polarity=-1.0, subjectivity=1.0)
Negative
RT @faitthh0: Cats are evil idc what nobody says
Sentiment(polarity=-1.0, subjectivity=1.0)
Negative
RT @yoginnnnnn: Thank you for liking these shark cats! https://t.co/QCaQ2J1RMd
Sentiment(polarity=0.0, subjectivity=0.0)
Neutral
RT @shxbzzz: open this thread if you like cats :)
Sentiment(polarity=0.25, subjectivity=0.75)
Positive
RT @yoginnnnnn: Thank you for liking these shark cats! https://t.co/QCaQ2J1RMd
Sentiment(polarity=0.0, subjectivity=0.0)
Neutral
https://t.co/FIi77tYFX7
What next? Isolate from your Babies? ENOUGH!!!
Sentiment(polarity=0.0, subjectivity=0.25)
Neutral
RT @oizksj: Taejin as cats : a thread https://t.co/wT7p17xEbc
Sentiment(polarity=0.0, subjectivity=0.0)
Neutral
```

We can convert this into and dataframe and visualise the analysis as well:

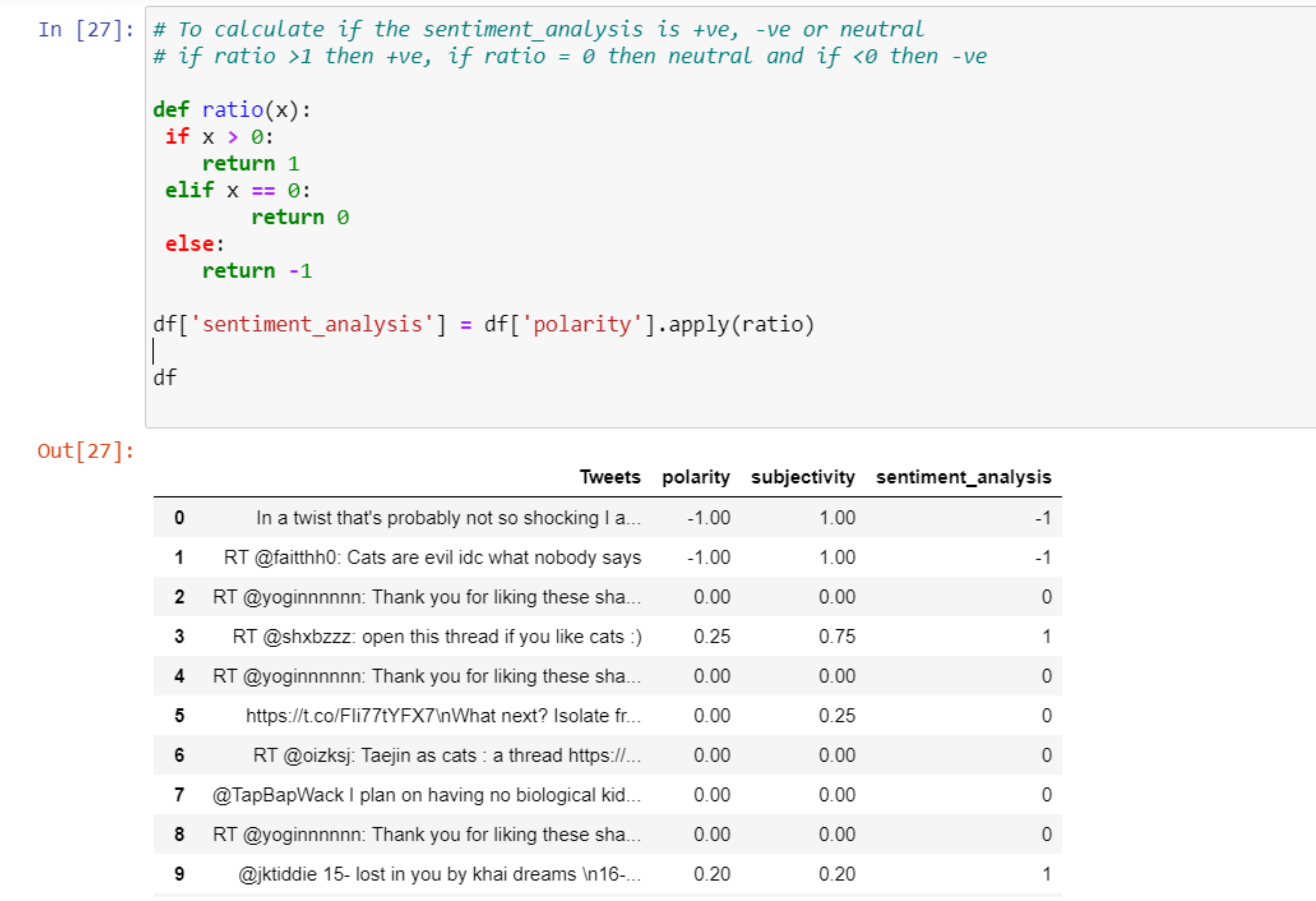


```
In [20]: # converting tweets into a dataframe w column name as Tweets
         import pandas as pd

         df = pd.DataFrame([tweets.text for tweets in public_tweets], columns=['Tweets'])

         # adding columns for the resp polarity and subjectivity
         polarity = lambda x: TextBlob(x).sentiment.polarity
         subjectivity = lambda x: TextBlob(x).sentiment.subjectivity
         df['polarity'] = df['Tweets'].apply(polarity)
         df['subjectivity'] = df['Tweets'].apply(subjectivity)
         df.head(10)
```

Out[20]:

	Tweets	polarity	subjectivity
0	In a twist that's probably not so shocking I a...	-1.00	1.00
1	RT @faitthh0: Cats are evil idc what nobody says	-1.00	1.00
2	RT @yoginnnnnn: Thank you for liking these sha...	0.00	0.00
3	RT @shxbzzz: open this thread if you like cats :)	0.25	0.75
4	RT @yoginnnnnn: Thank you for liking these sha...	0.00	0.00
5	https://t.co/FIi77tYFX7\nWhat next? Isolate fr...	0.00	0.25
6	RT @oizksj: Taejin as cats : a thread https://...	0.00	0.00
7	@TapBapWack I plan on having no biological kid...	0.00	0.00
8	RT @yoginnnnnn: Thank you for liking these sha...	0.00	0.00
9	@jktiddie 15- lost in you by khai dreams \n16-...	0.20	0.20

Adding the sentiment_analysis column to classify the tweets into positive, negative or neutral.



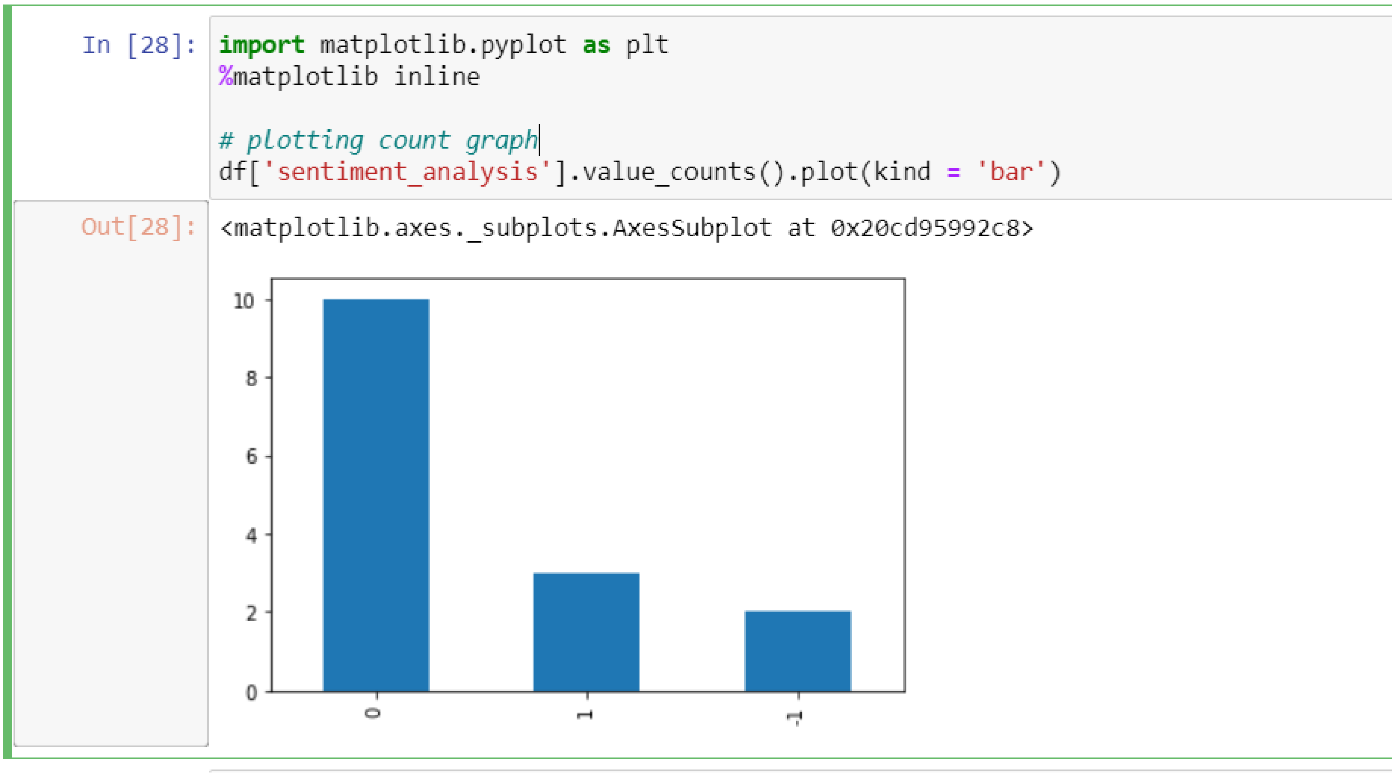
In [27]:

```
# To calculate if the sentiment_analysis is +ve, -ve or neutral
# if ratio >1 then +ve, if ratio = 0 then neutral and if <0 then -ve

def ratio(x):
 if x > 0:
    return 1
 elif x == 0:
        return 0
 else:
    return -1

df['sentiment_analysis'] = df['polarity'].apply(ratio)

df
```

Out[27]:

	Tweets	polarity	subjectivity	sentiment_analysis
0	In a twist that's probably not so shocking I a...	-1.00	1.00	-1
1	RT @faitthh0: Cats are evil idc what nobody says	-1.00	1.00	-1
2	RT @yoginnnnnn: Thank you for liking these sha...	0.00	0.00	0
3	RT @shxbzzz: open this thread if you like cats :)	0.25	0.75	1
4	RT @yoginnnnnn: Thank you for liking these sha...	0.00	0.00	0
5	https://t.co/FIi77tYFX7\nWhat next? Isolate fr...	0.00	0.25	0
6	RT @oizksj: Taejin as cats : a thread https://...	0.00	0.00	0
7	@TapBapWack I plan on having no biological kid...	0.00	0.00	0
8	RT @yoginnnnnn: Thank you for liking these sha...	0.00	0.00	0
9	@jktiddie 15- lost in you by khai dreams \n16-...	0.20	0.20	1

Plotting a count graph using matplotlib:



In [28]:

```
import matplotlib.pyplot as plt
%matplotlib inline

# plotting count graph
df['sentiment_analysis'].value_counts().plot(kind = 'bar')
```

Out[28]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x20cd95992c8>
```

Hence we find that most of the tweets related to the word 'Cats' are Neutral in nature. This is how sentiment analysis can be done easily using Textblob.

CONCLUSION:

The Sentiment Analysis of Tweets was implemented successfully using the Textblob python library in Jupyter Notebook. A certain word was typed in the code and several tweets containing that word, were analysed.

REFERENCES:

https://www.analyticsvidhya.com/blog/2018/02/natural-language-processing-for-beginners-using-textblob/#:~:text=Sentiment%20analysis%20is%20basically%20the,properties%2C%20polarity%2C%20and%20subjectivity.

https://textblob.readthedocs.io/en/dev/

http://docs.tweepy.org/en/latest/

https://developer.twitter.com/en/apps

https://monkeylearn.com/blog/sentiment-analysis-of-twitter/#:~:text=Sentiment%20analysis%20is%20the%20automated,are%20talking%20about%20their%20brand.

https://www.freecodecamp.org/news/how-to-build-a-twitter-sentiments-analyzer-in-python-using-textblob-948e1e8aae14/

https://www.earthdatascience.org/courses/use-data-open-source-python/intro-to-apis/analyze-tweet-sentiment-in-python/