Advanced Twitter Sentiment Analysis using Naïve Bayes Classification with Word Cloud Visualization

Bala Naveena @ Nivetha.M
Computer Science and Engineering,
National Engineering College,
Kovilpatti – 628503
balanive1217@gmail.com

Kavusalya.M.R
Computer Science and Engineering,
National Engineering College,
Kovilpatti – 628503
kavusalyamurugank@gmail.com

Mr.K.Raj Kumar.,ME
Assistant Professor,
Computer Science and Engineering,
National Engineering College,
Kovilpatti-628503
rajkathir08@gmail.com

Abstract*: In this world of growing technology, microblogging has become a trend in Social networking. People express their thoughts, opinions, emotions through such microblogging in Twitter, Facebook, Tumblr etc. Twitter is online news and social networking site where people communicate in short messages called tweets. Tweeting is sending short messages to anyone who follows you on Twitter. Sentiment Analysis is a process by which the information is extracted, processed and classified using classifier algorithms, to derive the sentiment of the given text whether it is a positive, negative or neutral one. In this paper, the Twitter tweets will be sentimentally classified using the most efficient supervised Machine learning classifier algorithm "Naïve Bayes Classification algorithm"(NBC). The reason for using NBC is that it is fast processing algorithm with quick converges. And also, less training set is sufficient for the classification. The input tweets will be collected using Tweepy and the text processing will be done by TextBlob and authentication needed for twitter will be taken responsibility by OAuthHandler. The advancement made in our paper is that, the input could be fed as keywords and also as the link/website of the twitter page. Thus either a single word or the whole page of the twitter could be used as input for the sentiment analysis. For all these, Python will be used and Anaconda Spider will be the platform. The Natural Language Tool Kit (NLTK) plays a vital role in this classification. The classified data with its sentiment will be displayed in percentage and chart representation. Proposal is to use Word cloud for the Data Visualization. Thus, it will be user friendly and impressive. When this analysis method is used for analyzing the sentiment of Social Medias, good solutions for any such kind of social issues can be attained and this helps to visualize the response of people and their sentiments.

Keywords—Tweets, Twitter, Naïve Bayes classifier, sentiment, positive, negative, neutral, nltk, data visualization, word cloud.

I. INTRODUCTION

Twitter is a popular microblogging website. With over 317 million active users a month, Twitter has become a wealth of data for those trying to understand how people feel about brands, topics, opinions and more. Each tweet is 140 characters in length. They are used to express a tweeter's emotion ,opinion on a particular subject. Mining Twitter data for insights is one of the most common natural language processing tasks. Generally, an opinion is "simply a positive or negative sentiment, view, attitude, emotion, or appraisal about an entity or an aspect of the entity from an opinion holder at a specific time ". The entity can be a product/service, event, person, organization, or topic consisting of aspects (features/attributes) that represents both components and attributes of the entity. With the explosion of user generated opinions there is the need by companies, politicians, service providers, social psychologists, researchers and other actors to analyze them in order to implement better decision choices.

II. SCOPE OF SENTIMENT ANALYSIS

Twitter sentiment analysis is a very novel problem from an academic standpoint. For a long time sentiment analysis was largely based on text samples that were a paragraph or larger. With the large amount of information in a paragraph of text it's much easier to achieve good performance. When twitter first came out, it utterly broke most existing sentiment analysis approaches, mostly because it was based around very short, informal text. It became a very interested problem, not just from a business perspective, but also from an academic perspective. The other nice piece was that the Twitter dataset is massive. The huge amount of data that twitter makes available made it possible to explore a number of machine learning approaches, specifically in the realm of neural networks that were largely intractable on smaller datasets. Ultimately, the synergy between academic curiosity and business interest led to the large amount of interest in twitter sentiment analysis seen publicly. Sentiment analysis was chosen largely because it is a simple binary classification problem that does away with many confounding factors like class imbalance that would impede more complex problems. Social media sentiment analysis can be an excellent source of information that provides insights that can [1] determine marketing strategy, [2] improve campaign success, [3] product messaging, [4] customer service, [5] test business KPIs, and [6] generate leads.

A. Adjust marketing strategy

Most companies, if not all, are active in social media, and use the public forum to promote their brands and services. It is a place where the customers chit chat about the brand and is full of information about how brand is being perceived by the target customers. The information people get from sentiment analysis provides us with means to optimize one's marketing strategy. By listening to what the customers feel and think about the brand, one can adjust their high-level messaging to meet their needs. From the tactical point of view, one can build a short-term marketing campaign to provide customers with what they want. By continuously having sentiment analysis in place, they can adjust their campaign to fit even more to their target audience. Hours of work typically goes into preparing marketing campaigns.

B. *Develop product quality*

Sentiment analysis helps to complete the market research by getting to know what the customers' opinions are about the products/services and how one can align the products/services' quality and features with their tastes. Since people freely express their true opinions about their experience with various products and brands in social media, it is a better way to gather the data from social media and analyze.

C. *Lead generation*

By having accurate sentiment analysis in place and as a result of adjusting the marketing campaigns, having great customer service, and improving the product quality to meet the needs of the market, one will be able to increase leads. Loyal and happy customers, who will act as your brand ambassadors, will bring you new customers.

D. *Sales Revenue*

The biggest benefit of doing sentiment analysis is to boost sales revenue. The increase in sales revenue is the final outcome of successful marketing campaigns, improved products/service quality, and customer service, which can be achieved with sentiment analysis. When there are more positive discussions going on, the sales revenue will increase, and when there are more negative discussions going on, the sales revenue will decrease. This is proved by Public Response Analysis.

III. LITERATURE SURVEY

There are numerous research papers and studies that focus on Sentiment classification for Twitter. These studies describe some interesting methodologies of detecting and identifying sentiment from twitter data. Classifying tweets into positive and negative classes using distant supervision was presented by Alec Go, Richa Bhayani and Lei Huang , 2009 .They presented an approach for automatically classifying the sentiment of twitter messages with respect to a query term . They presented results of machine learning algorithms (Naive Bayes , maximum Entropy , and SVM) for classification . Use of linguistic features for detecting sentiment of twitter messages was investigated by Efthymios Kouloumis , Theresa Wilson and Johanna Moore (2011). They used hashtagged dataset(HASH) for development and training. Hassan Saif , Yulan He and Harith Alani , 2012 explained an approach of adding semantics as additional features into training set for sentiment analysis.

A. *Sentiment Analysis of Twitter Data using Machine Learning Approaches* by Ankit Pradeep Patel et al., published their work in IJIRST –International Journal for Innovative Research in Science & Technology-March 2017.They used SVM(Support Vector Machine)algorithm for the classification of tweets. SVM has been shown to be highly effective in traditional text categorization.SVM measure the complexity of hypothesis based on the margin with which they separate the data instead of the number of features. One remarkable property of SVM is that their ability to learn can be independent of the dimensionality of the feature space. To construct a feature vector of the document stop words are removed first and then each distinct word in the document is used to represent a feature. Support Vector Machines (SVMs) are supervised learning methods used for classification

B. *Twitter sentiment analysis with Machine Learning in R using doc2vec approach* by Sergey Bryl‘, a Data Scientist at MacPaw Inc., published his work-February, 2017. Deep learning algorithm was used. Deep learning allows algorithms to understand sentence structure and semantics. They represent every word as a vector and an operator (a matrix) which seems very intuitive. Thus the word "not" can be a rotation matrix that acts on the next word (for eg. good) and changes it's polarity by rotating the vector of good to now mean not good. This is a very powerful concept. However these networks require a lot of training data (parse trees are required to train these networks). Word2vec and Paragraph vectors have been shown to work very well in sentiment analysis

C. *Sentiment Analysis and Summarization of Twitter Data* by Seyed-Ali Bahrainian, Andreas Dengel of University Of Kaiserslautern,Germany, published his work in IEEE 16th International Conference. Various machine learning algorithms were used for analyzing the tweets and to derive the sentiments. In that work, they used two types of feature sets and three base classifiers to form the ensemble framework. Two types of feature sets are created using Part-of-speech information and Word-relations. Naive Bayes, Maximum Entropy and Support Vector Machines are selected as base classifiers. They applied different ensemble methods like Fixed combination, Weighted combination and Meta-classifier combination for sentiment classification and obtained better accuracy.

D. *Real Time Sentiment Analysis of Tweets Using Naive Bayes* byAnkur Goel et al.,published their work in IEEE October 2016.The paper contained implementation of Naive Bayes using sentiment140 training data using twitter database and propose a method to improve classification . Use of SentiWordNet along with Naive Bayes can improve accuracy of classification of tweets, by providing positivity, negativity and objectivity score of words present in tweets.

IV. PROPOSED SYSTEM

In the proposed system, the input is fed either as a single keyword or as the link of the twitter page. Those tweets were retrieved/ extracted from twitter using twitter API called Tweepy based on the query. The collected tweets will be subjected to pre-processing. This system uses TextBlob for this pre-processing and OAuthHandler for authentication purpose.Then the supervised algorithm was applied on the stored data. The supervised algorithm used in our system is Naïve Bayes Classification (NBC). The classified data will be analyzed to find out the sentiment of the text. The results of the algorithm i.e. the sentiment (whether it is positive or negative or neutral) will be represented in easy mathematical representation (percentage) and also in many user friendly representations. The representation will be in chart and graphs. Further impressive visualization of the result will be the Word Cloud Visualization technique. This technique displays the words frequently used in the tweets with varying colours, depending upon the sentiment of the text and also in varying sizes depending upon the frequency of the word (number of times used) in the tweets. The proposed system is more effective than the existing one. This is because NBC is easy to implement. It needs only less training set. It leads to good results most of the time. And it will be more desirable because of its high speed. The language used for execution of the project work is Python. The execution will be carried out in both Windows and Linux platform. Thus it reduces the maintenance cost.

V. PLATFORM FOR PROJECT

Software: Anaconda-Spyder(3.6)
API: Tweepy, TextBlob, OAuthHandler
Language: Python

VI. SYSTEM MODEL

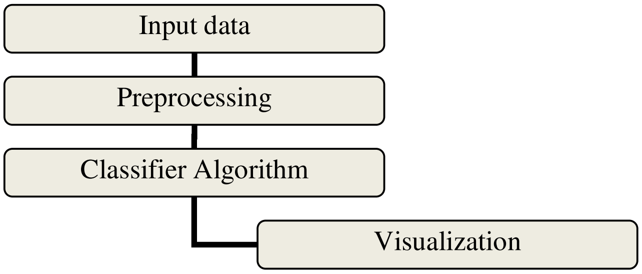


Fig: 1- Flow Diagram for the system

The input data and the training set are fed into the system. This data is preprocessed or cleaned. Then the classification algorithm is applied to that data and then the result is visualized. Refer *Fig: 1* for the flow diagram.

VII. SYSTEM ARCHITECTURE

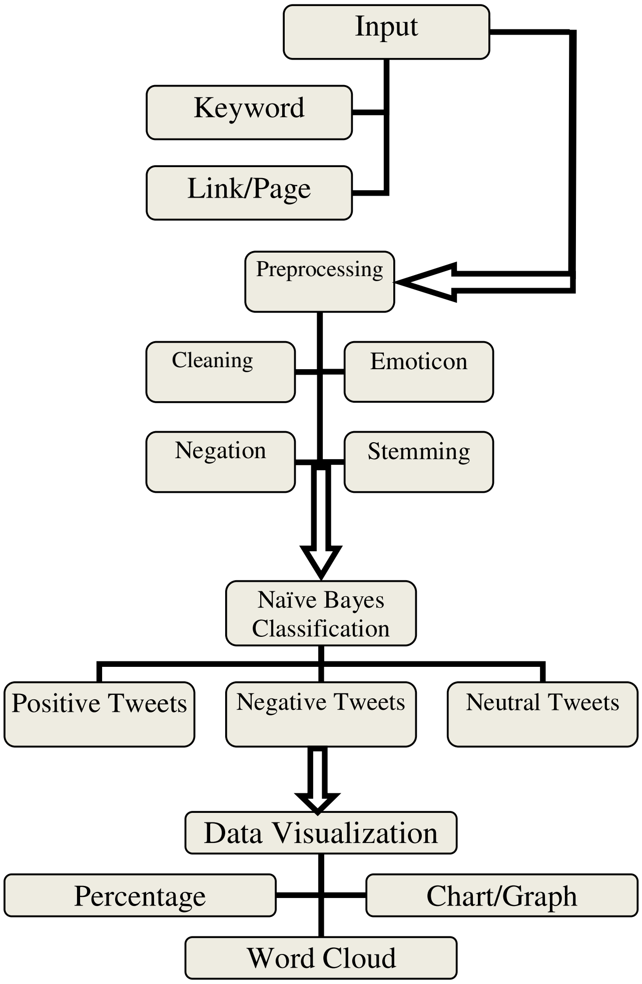


Fig: 2- System Architecture

A. *Input:*

All the input either the single keyword or the link/URL of the entire page is fed as the input as said in *Fig:2*. All the tweets and the data are collected from the Twitter website using the Tweepy API. For this, just import this tweepy in the Anaconda platform via the Python coding. And the important thing needed to collect the tweets from the twitter would be the authentication. For this purpose, OAuthHandler is also imported in the Spyder.

B. *Training Dataset:*

Another input is the Training set needed for the classification purpose. The system will compare and analyze the test data with the training set data. So this plays a vital role in the system's accuracy. Training data is most important part of the whole system as training of the system is wholly depends on it and classification of testing data is done on the basis of this result only. While choosing training data, type of problem should be taken into consideration as similar type of training data should be taken so that it can provide more efficient and accurate results like if problem is related to movie review then training data can be taken from IMDb or if problem is related to food review then it is better to use review of zomato or if problem is related to product review then data can be taken from amzon.com because this type of training data is more associated to the type of problem statement.

C. Preprocessing:

The huge amount of input data may contain many irrelevant things which make it tough to handle. So it is always suggested to remove irrelevant portion of data or make it relevant which can further increase efficiency of the system. Taking example of twitter data, tweets contains so many data types such as User ID (staring with @), URLs, text, date and time, location, multimedia files (images, videos etc), emoticons, hashtags (staring with #). Each of these have their own significance during the sentiment analysis and some are irrelevant which do not have any significant effect so it is suggested to omit these data while sentiment analysis. User name in tweets always starts with @ symbol, this is to tell who tweeted this particular tweet , while doing sentiment analysis there is no significant effect of user name so by applying filters, user name is excluded from training as well from testing data. There is character limitation in tweets so users includes some URL links to explain it better, these URLs (generally start with http://) are not needed during training and testing of data as they do not contain any useful information in it which can be used during sentiment analysis. Text is actual body of tweet which can be maximum 140 character longer and contains everything that any user want to tweet. This text is mainly used for sentiment analysis but it should be free of irrelevant data. Date and time stamp is attached with every tweet which tells that about when a particular tweet is tweeted. This feature is very useful to find how frequently any user tweets and to find most buzzing word of twitter for any particular time interval. Location can also be traced in tweets but this is optional feature of user whether he/she wants to share location or not, this feature helps in to find the trends in any particular demographical area. Here in above mentioned system this feature is not used and omitted. Emoticons are very useful symbols present in the text of tweets, they emphasize the sentiments of tweets and also help to find out the true sentiment of tweets.

Stages of Preprocessing:

1) Basic cleaning:

In order to provide only significant information, in general a clean tweet should not contain URLs, hashtags (i.e. #happy) or mentions (i.e. @ModiJi). Refer *Fig:2.* Furthermore, tabs and line breaks should be replaced with a blank and quotation marks with apexes. Finally, all the text is converted to lower case, and extra blank spaces are removed.

2) Emoticon:

The last step is to convert many types of emoticons into tags that express their sentiment (i.e. :) ! smile happy).

Examples

Positive :-j =p :] :-P ;) :p :3 =] :b :-) 8) _/ :') ;-) :-p :S
Negative :-/ : :'(:[= :/ :@ :'-(:c ;(=/

3) Negation:

Dealing with negations (like \not good") is a critical step in Sentiment Analysis. A negation word can influence the tone of all the words around it, and ignoring negations is one of the main causes of misclassification. In this phase, all negative constructs (can't, don't, isn't, never etc) are re- placed with \not". This technique allows the classifier model to be enriched with a lot of negation bigram constructs that would otherwise be excluded due to their low frequency

4) Stemming:

Stemming techniques put word variations like \great",\greatly", \greatest", and \greater" all into one bucket, effectively decreasing entropy and increasing the relevance of the concept of \great". In other words, Stemming allows us to consider in the same way nouns, verbs and adverbs that have the same radix. As in the case of emoticons, with the use of this technique it is possible to combine features with the same meaning and reduce the entropy of the model.

D. Classification:

Classification basically means categorizing data into different classes based on some computation which determines the sentiment behind the data. Many classifiers can be used for classification process like Naive Bayes classifier, Support vector machine, Baseline etc. Here Naive Bayes classifier is being used for the classification process. Naive Bayes is mostly preferred for classification due to its speed and simplicity. Many classifiers can be used for classification process like Naive Bayes classifier, Support vector machine , Baseline etc. Refer *Fig:2.* Here Naive Bayes classifier is being used for the classification process.

1. Naïve Bayes Classifier:

Naive Bayes is mostly preferred for classification due to its speed and simplicity. Naive Bayes classifier assumes that the presence of a particular feature in a class is not related to the presence of any other feature. For example, a fruit may be considered to be an apple if it is yellow, round and about 3 inches in diameter. Even if these features depend on each other, all of these properties contribute to the property that this fruit is orange and thus the name "Naive". Mathematically, for a word w and class c , by Bayes Theorem

$P(c/w) = [P(w/c)P(c)]/P(w)$ ……………………………....(1)

Where P(c/w) is probability of class c given word is w. P(c) is probability of class c and P(w) is probability of word w.
Naive Bayes classifier will be

$c^{*}=\arg\max c\ P(c/w)$ ....(2)

*The algorithm: (*Naïve Bayes Classifier)

Naive Bayes Classifier makes use of all the features in the feature vector and analyzes them individually as they are equally independent of each other. The conditional probability for Naive Bayes can be defined as

$P(X \mid yj) = \pi^{m}{}_{i=1}P(xi \mid yj)$ ……….…………………………(1)

'X' is the feature vector defined as X=fx1,x2,....xmg and yj is the class label. Here, in our work there are different independent features like emoticons, emotional keyword, count of positive and negative keywords, and count of positive and negative hash tags which are effectively utilized by Naïve Bayes classifier for classification. Naive Bayes does not consider the relationships between features. So it cannot utilize the relationships between part of speech tag, emotional keyword and negation. By using the Classifier, the tweets are classified as positive, negative and neutral tweets.

E. Visualization:

The resulting data i.e. the tweets that were categorized as positive, negative and neutral has to be displayed to the user. Sample output attained:

Positive tweets:
b'RT @malviyamit: Modi sarkaar making lives easy for the middle class... https://t.co/pJHXg0AwF8'
b"RT @HartoshSinghBal: how easily those looking for hope in anyone against modi delude themselves. hardik shares much with modi's worldview,\xe2\x80\xa6"
b'RT @Ish_Bhandari: So RaghuRam Rajan criticising PM Modi &amp; economic slowdown as due to his policy at #Davos \n\nOnly @Swamy39 had identified h\xe2\x80\xa6'

Negative tweets:
b"RT @24x7Politics: Few days ago, Mr. Modi asked for his Govt's performance to be compared with that of the UPA. \n\nRequest Granted.\n\nIn no ma\xe2\x80\xa6"
b'RT @VidyaNandMishr8: Indians Are Worse Off Under Modi via @forbes https://t.co/6VNXWJWV0l Country has nose dived under BJP rule @OfficeOfRG\xe2\x80\xa6'
b'RT @waglenikhil: Shiv Sena\xe2\x80\x99s decision is not unexpected. But will they stick to it? When will SS ministers resign from Modi/ Fadnavis gover\xe2\x80\xa6'

Further Visualization is done in many ways.

1. *The Basic Visualization-Percentage:*

The percentage of positive, negative and neutral tweets is displayed. This is an easy way to represent the result in mathematical form. Refer *Fig:3*

```
Twitter Sentimental Analysis result for modi
Positive tweets percentage: 32.87671232876713 %
Negative tweets percentage: 13.698630136986301 %
Neutral tweets percentage: 53.42465753424658 %
```

2. *Charts/ Graphs:*

User friendly approach of data visualization is Charts and graphs. The resulting tweets' percentage is displayed as charts and graphs for easy understanding. Refer *Fig:4*

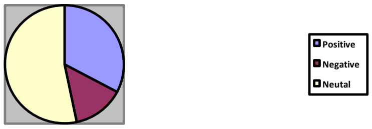


Fig 3: Data Visualization-Chart

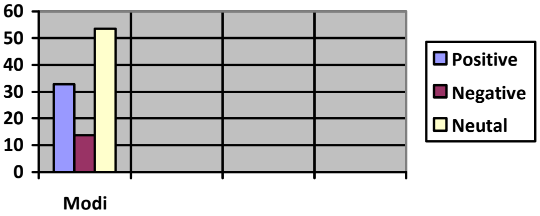
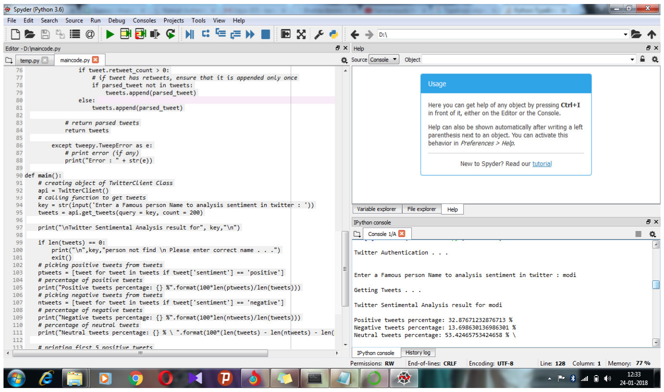


Fig 4: Data Visualization-Graph

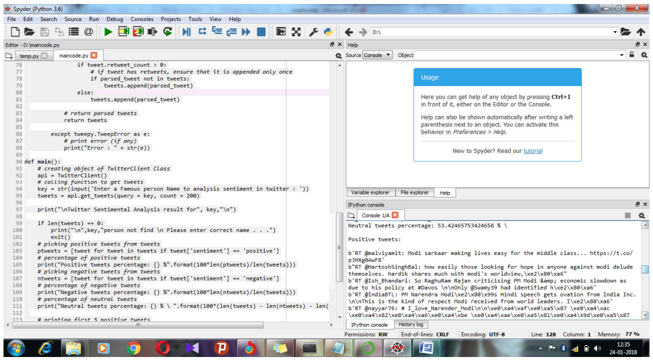
Screenshots:


Spyder (Python 3.6)
File Edit Search Source Run Debug Consoles Projects Tools View Help
D:\
Editor - D:\maincode.py
temp.py
maincode.py
Help
Source Console
Object
Usage
Here you can get help of any object by pressing Ctrl+I in front of it, either on the Editor or the Console.
Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in Preferences > Help.
New to Spyder? Read our tutorial
Variable explorer
File explorer
Help
IPython console
Console 1/A
IPython console
History log
Permissions: RW
End-of-lines: CRLF
Encoding: UTF-8
Line: 128
Column: 1
Memory: 77 %
12:33
24-01-2018

```
                if tweet.retweet_count > 0:
                    # if tweet has retweets, ensure that it is appended only once
                    if parsed_tweet not in tweets:
                        tweets.append(parsed_tweet)
                else:
                    tweets.append(parsed_tweet)

            # return parsed tweets
            return tweets

        except tweepy.TweepError as e:
            # print error (if any)
            print("Error : " + str(e))

def main():
    # creating object of TwitterClient Class
    api = TwitterClient()
    # calling function to get tweets
    key = str(input('Enter a Famous person Name to analysis sentiment in twitter : '))
    tweets = api.get_tweets(query = key, count = 200)

    print("\nTwitter Sentimental Analysis result for", key,"\n")

    if len(tweets) == 0:
        print("\n",key,"person not find \n Please enter correct name . . .")
        exit()
    # picking positive tweets from tweets
    ptweets = [tweet for tweet in tweets if tweet['sentiment'] == 'positive']
    # percentage of positive tweets
    print("Positive tweets percentage: {} %".format(100*len(ptweets)/len(tweets)))
    # picking negative tweets from tweets
    ntweets = [tweet for tweet in tweets if tweet['sentiment'] == 'negative']
    # percentage of negative tweets
    print("Negative tweets percentage: {} %".format(100*len(ntweets)/len(tweets)))
    # percentage of neutral tweets
    print("Neutral tweets percentage: {} % \ ".format(100*(len(tweets) - len(ntweets) - len(

    # printing first 5 positive tweets
```

```
Twitter Authentication . . .


Enter a Famous person Name to analysis sentiment in twitter : modi

Getting Tweets . . .

Twitter Sentimental Analysis result for modi

Positive tweets percentage: 32.87671232876713 %
Negative tweets percentage: 13.698630136986301 %
Neutral tweets percentage: 53.42465753424658 % \
```

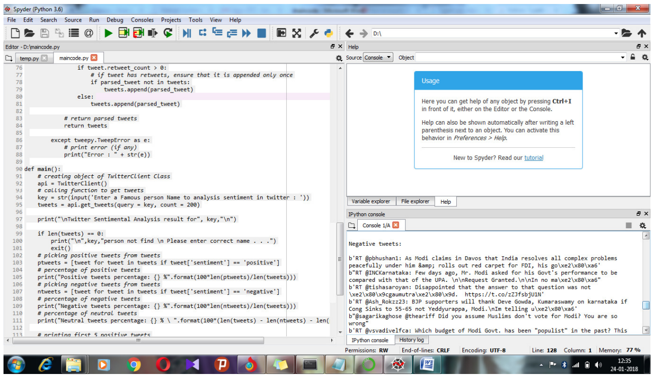
Fig 5: Output


Spyder (Python 3.6)
File Edit Search Source Run Debug Consoles Projects Tools View Help
D:\
Editor - D:\maincode.py
temp.py
maincode.py
Help
Source Console
Object
Usage
Here you can get help of any object by pressing Ctrl+I in front of it, either on the Editor or the Console.
Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in Preferences > Help.
New to Spyder? Read our tutorial
Variable explorer
File explorer
Help
IPython console
Console 1/A
IPython console
History log
Permissions: RW
End-of-lines: CRLF
Encoding: UTF-8
Line: 128
Column: 1
Memory: 77 %
12:35
24-01-2018

```
                if tweet.retweet_count > 0:
                    # if tweet has retweets, ensure that it is appended only once
                    if parsed_tweet not in tweets:
                        tweets.append(parsed_tweet)
                else:
                    tweets.append(parsed_tweet)

            # return parsed tweets
            return tweets

        except tweepy.TweepError as e:
            # print error (if any)
            print("Error : " + str(e))

def main():
    # creating object of TwitterClient Class
    api = TwitterClient()
    # calling function to get tweets
    key = str(input('Enter a Famous person Name to analysis sentiment in twitter : '))
    tweets = api.get_tweets(query = key, count = 200)

    print("\nTwitter Sentimental Analysis result for", key,"\n")

    if len(tweets) == 0:
        print("\n",key,"person not find \n Please enter correct name . . .")
        exit()
    # picking positive tweets from tweets
    ptweets = [tweet for tweet in tweets if tweet['sentiment'] == 'positive']
    # percentage of positive tweets
    print("Positive tweets percentage: {} %".format(100*len(ptweets)/len(tweets)))
    # picking negative tweets from tweets
    ntweets = [tweet for tweet in tweets if tweet['sentiment'] == 'negative']
    # percentage of negative tweets
    print("Negative tweets percentage: {} %".format(100*len(ntweets)/len(tweets)))
    # percentage of neutral tweets
    print("Neutral tweets percentage: {} % \ ".format(100*(len(tweets) - len(ntweets) - len(

    # printing first 5 positive tweets
```

```
Neutral tweets percentage: 53.42465753424658 % \

Positive tweets:

b'RT @malviyamit: Modi sarkaar making lives easy for the middle class... https://t.co/
pJHXg0AwF8'
b"RT @HartoshSinghBal: how easily those looking for hope in anyone against modi delude
themselves. hardik shares much with modi's worldview,\xe2\x80\xa6"
b'RT @Ish_Bhandari: So RaghuRam Rajan criticising PM Modi &amp; economic slowdown as
due to his policy at #Davos \n\nOnly @Swamy39 had identified h\xe2\x80\xa6'
b'RT @IndiaBTL: PM Narendra Modi\xe2\x80\x99s Hindi speech gets ovation from India Inc.
\n\nThis is the kind of respect Modi received from world leaders. I\xe2\x80\xa6'
b'RT @nayyar76: # I_love_Narender_Modi\n\n\xe0\xa4\xaf\xe0\xa5\x87 \xe0\xa4\xac
\xe0\xa4\x82\xe0\xa4\xa6\xe0\xa4\xbe \xe0\xa4\xae\xe0\xa5\x81\xe0\xa4\x9d\xe0\xa5\x87
```

Fig 6: Output



```
            if tweet.retweet_count > 0:
                # if tweet has retweets, ensure that it is appended only once
                if parsed_tweet not in tweets:
                    tweets.append(parsed_tweet)
            else:
                tweets.append(parsed_tweet)

        # return parsed tweets
        return tweets

    except tweepy.TweepError as e:
        # print error (if any)
        print("Error : " + str(e))

def main():
    # creating object of TwitterClient Class
    api = TwitterClient()
    # calling function to get tweets
    key = str(input('Enter a Famous person Name to analysis sentiment in twitter : '))
    tweets = api.get_tweets(query = key, count = 200)

    print("\nTwitter Sentimental Analysis result for", key,"\n")

    if len(tweets) == 0:
        print("\n",key,"person not find \n Please enter correct name . . .")
        exit()
    # picking positive tweets from tweets
    ptweets = [tweet for tweet in tweets if tweet['sentiment'] == 'positive']
    # percentage of positive tweets
    print("Positive tweets percentage: {} %".format(100*len(ptweets)/len(tweets)))
    # picking negative tweets from tweets
    ntweets = [tweet for tweet in tweets if tweet['sentiment'] == 'negative']
    # percentage of negative tweets
    print("Negative tweets percentage: {} %".format(100*len(ntweets)/len(tweets)))
    # percentage of neutral tweets
    print("Neutral tweets percentage: {} % \ ".format(100*(len(tweets) - len(ntweets) - len(

    # printing first 5 positive tweets
```

Usage

Here you can get help of any object by pressing **Ctrl+I** in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in *Preferences > Help*.

New to Spyder? Read our tutorial

```
Negative tweets:

b'RT @pbhushan1: As Modi claims in Davos that India resolves all complex problems
peacefully under him &amp; rolls out red carpet for FDI, his go\xe2\x80\xa6'
b"RT @INCKarnataka: Few days ago, Mr. Modi asked for his Govt's performance to be
compared with that of the UPA. \n\nRequest Granted.\n\nIn no ma\xe2\x80\xa6"
b'RT @tishasaroyan: Disappointed that the answer to that question was not
\xe2\x80\x9cgaumutra\xe2\x80\x9d.  https://t.co/zZJfsbjU1N'
b'RT @Ash_Rokzz23: BJP supporters will thank Deve Gowda, Kumaraswamy on karnataka if
Cong Sinks to 55-65 not Yeddyurappa, Modi.\nIm telling u\xe2\x80\xa6'
b"@sagarikaghose @theariff Did you assume Muslims don't vote for Modi? You are so
wrong"
b'RT @vsvadivelfca: Which budget of Modi Govt. has been "populist" in the past? This
```

Permissions: RW End-of-lines: CRLF Encoding: UTF-8 Line: 128 Column: 1 Memory: 77 %

12:35 24-01-2018

Fig 7: Output

3. *Word Cloud:*

A word cloud represents word usage in a document by resizing individual words proportionally to its frequency, and then presenting them in random arrangement. *Refer Fig 8.* Some of the concerns over word cloud are that, it supports only the crudest sorts of textual analysis, and it is often applied to situations where textual analysis is not appropriate, and it leaves viewers to figure out the context of the data by themselves without providing the narrative. But in the case of tweets, textual analysis is the most important analysis, and it provides a general idea of what kind of words are frequent in the corpus, in a sort of quick way. For the word cloud, use the python library word cloud.

Sample output:

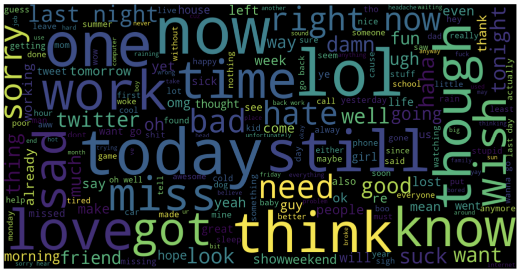


Fig 8: Data Visualization-Word Cloud

Thus, from the above word cloud, the words with high frequency are displayed in large font text and low number of frequency is displayed in small size. And it is also displayed in various colors for positive negative and neutral keywords. This is user friendly and impressive way of data visualization.

VIII. CONCLUSION

The Sentiment Analysis of twitter data can be used in many ways like customer's review, marketing, and in all such fields. Although many classifiers are available but Naive Bayes have been used because of its speed and the less amount of training data set that is needed for the classification. Thus this provides to be quite simple and accurate.

References

[1] Ajay Deshwal, Sudhir Kumar Sharma, "Twitter Sentiment Analysis using Various Classification Algorithms" *KIIT College of Engineering, Gurgaon, India-IEEE 2016.*

[2] Alec Go, Richa Bhayani and Lei Huaug, Stanford university(2009), "Twitter Sentiment Classification using Distant Supervision," in:The Third International Conference on Data Analytics.

[3] E. Boiy, P. Hens, K. Deschacht, and M.-F. Moens, "Automatic sentiment analysis in on-line text," in Proceedings of the 11th International Conference on Electronic Publishing, pp. 349–360, 2007.

[4] A. Kumar qnd T.M. Sebastian, "Machine Learning assisted Sentiment Analysis". Proceedings of International Conference on computer science and engineering (ICCSE'2012), 2012.

[5] Bifet and E. Frank, "Sentiment Knowledge Discovery In Twitter Streaming Data", In proceedings of 13th International Conference of Discovery Science, Berlin, Germany : Springer, 2010.

[6] Ankit Pradeep Patel et al., "*Sentiment Analysis of Twitter Data using Machine Learning Approaches"* in IJIRST –International Journal for Innovative Research in Science & Technology-March 2017.

[7] Sergey Bryl', a Data Scientist at MacPaw Inc., "*Twitter sentiment analysis with Machine Learning in R using doc2vec approach"*-February, 2017.

[8] Seyed-Ali Bahrainian, Andreas Dengel of University Of Kaiserslautern,Germany, "*Sentiment Analysis and Summarization of Twitter Data"* in IEEE 16th International Conference.

[9] Ankur Goel et al.,"*Real Time Sentiment Analysis of Tweets Using Naive Bayes* in IEEE October 2016.