International Journal of Computing and Corporate Research
ISSN (Online) : 2249-054X
Volume 5 Issue 5 September 2015
International Manuscript ID : 2249054XV5I5092015-02

# SENTIMENT ANALYSIS FROM SOCIAL MEDIA LIVE FEEDS USING UNSTRUCTURED DATA MINING

*Manu Bansal*
*Assistant Professor*
*Department of IT*
*University Institute of Engineering & Technology*
*Panjab University, Chandigarh*

## ABSTRACT

Web Monitoring, Scraping and digital forensic is one of the prominent areas in the domain of Big Data and Sentiment Analysis. A number of software products and tools are available in the technology market which are used to guards the network infrastructure and confidential data against cyber threats and attacks. From long time, the monitoring of servers and forensic analysis of network infrastructure is done using packet capturing (PCAP) tools. These activities are performed using PCAP and related tools available in the market which includes open source software as well as commercial products. As far as the fame and usage of the software suites is concerned, the open source market is getting popularity because of the scope of customization and organization specific personalization the software products. In this research paper, an approach is depicted for the fetching and analysis of live data from social media portals and using such approaches the sentiment data analysis can be implemented effectively.

Keywords – Big Data Analytics, Python for Web Scraping, Live Tweets Fetching from Twitter

## INTRODUCTION

As per the international statistical reports from Statista, there are around 1 million new user registrations on Whatsapp. Besides this, 700 million active users present on Whatsapp. Around 30 Billion messages are sent and 34 billion messages are received everyday.

If we analyze the statistics of Twitter, 350 Million Tweets daily and more than 500 Million Accounts. There is the huge and rapid growth in the unstructured data every moment. The production and generation of data is predicted to be 44 times in 2020 as compared to the data in 2009.

All these figures and statistical data are amazing and growing in exponential pattern. Such data is unstructured in nature which means the data of different and heterogeneous formats. This concept is classically known as Big Data. The deep investigation of intelligence and meaningful patterns from Big Data is known as Big Data Analytics. A number of researchers and scientists are working in this domain of Big Data using assorted technologies and tools.

There are number of approaches by which the live data can be obtained for research and development. One of these approaches is getting data from Open Data Portals. The open data portals provide authentic data sets for research and development in multiple domains. The data sets can be downloaded from these portals in multiple formats including XML, CSV, JSON and many others.

### LinkChecker

Using LinkChecker library in Python, the recursive and deep checking of the server pages can be done. Using LinkChecker, the site crawling is made easy with features of integrating the regular expressions and filtering. The output can be generated in multiple formats including HTML, XML, CSV, SQL or simply the sitemap graph.

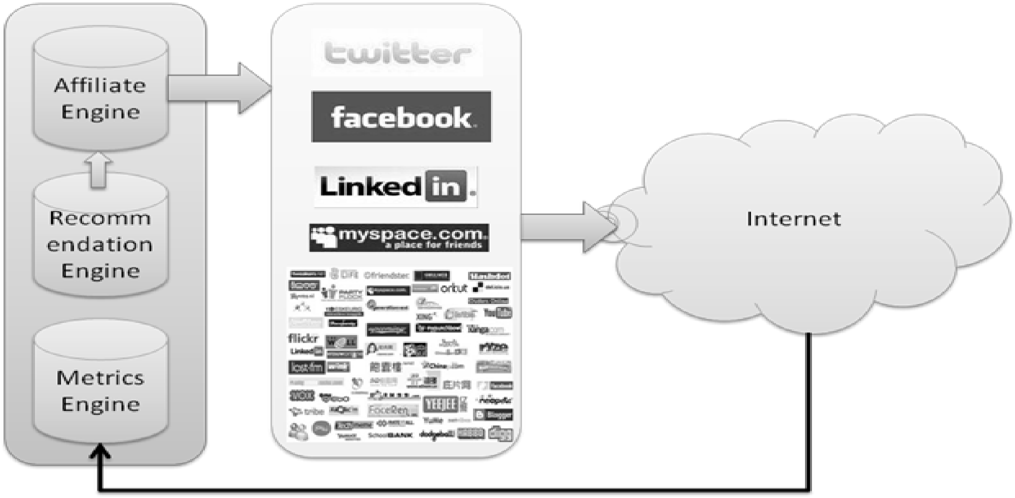


Figure 1 – Recommender System and Related Engines in Social Media

**WebScraping**

Python is used by the researchers and practitioners for collecting the live data for research and development. For example, we can fetch the live records of stock market, price of any product from E-Commerce websites. Such data collected is the foundation of BigData Analytics. If a researcher is doing research on big data analysis, the live data can be fetched using Python Script and then it can be processed based on the research objectives.

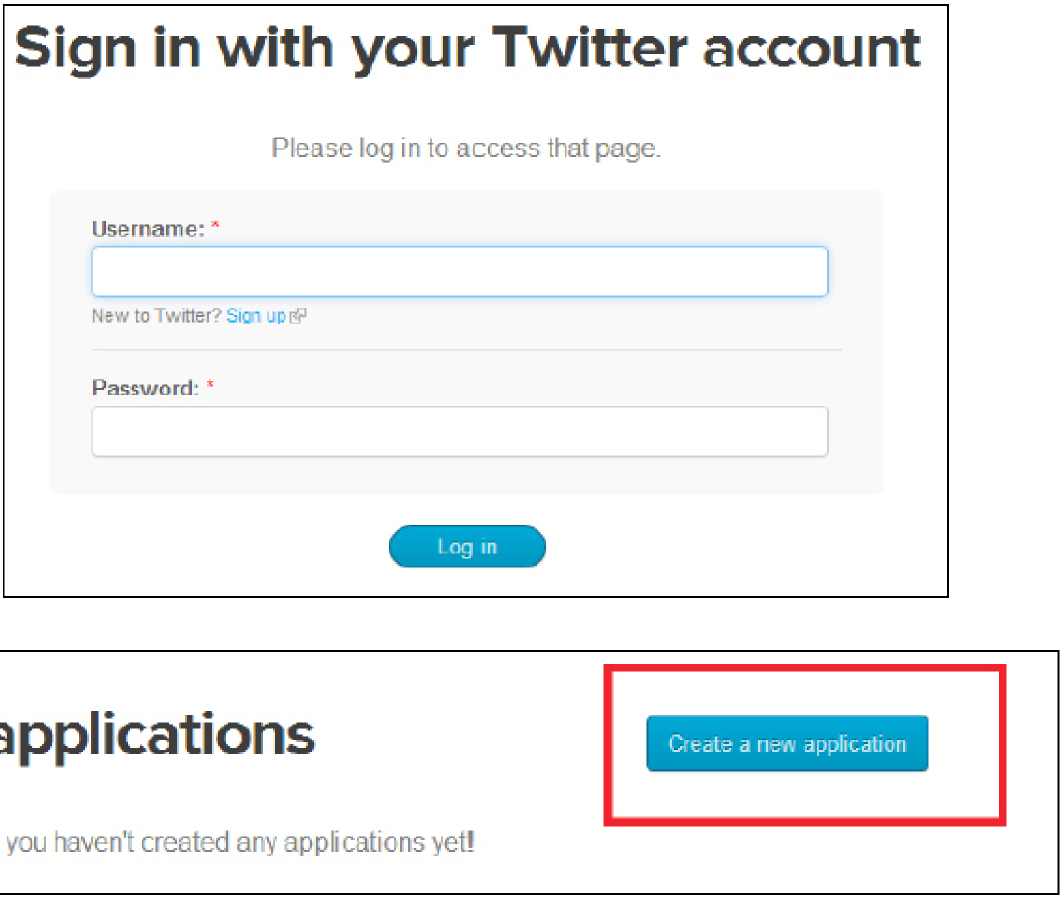
Here is the code snippet to fetch the live stock exchange data from the website timesofindia.com using Python

```
from bs4 import BeautifulSoup
import urllib.request
```

```
from time import sleep
from datetime import datetime
def getnews():
    url = "http://timesofindia.indiatimes.com/business"
    req = urllib.request.urlopen(url)
    page = req.read()
    scraping = BeautifulSoup(page)
    price = scraping.findAll("span",attrs={"class":"red14px"})[0].text
    return price
with open("bseindex.out","w") as f:
    for x in range(2,100):
        sNow = datetime.now().strftime("%I:%M:%S%p")
        f.write("{0}, {1} \n ".format(sNow, getnews()))
        sleep(1)
```

### Fetching Live Data from Social Media

In the same way, the twitter live feeds can be fetched using Python APIs. Using twitter developer account, the new app can be created and then the Python Script is mapped with the Twitter App



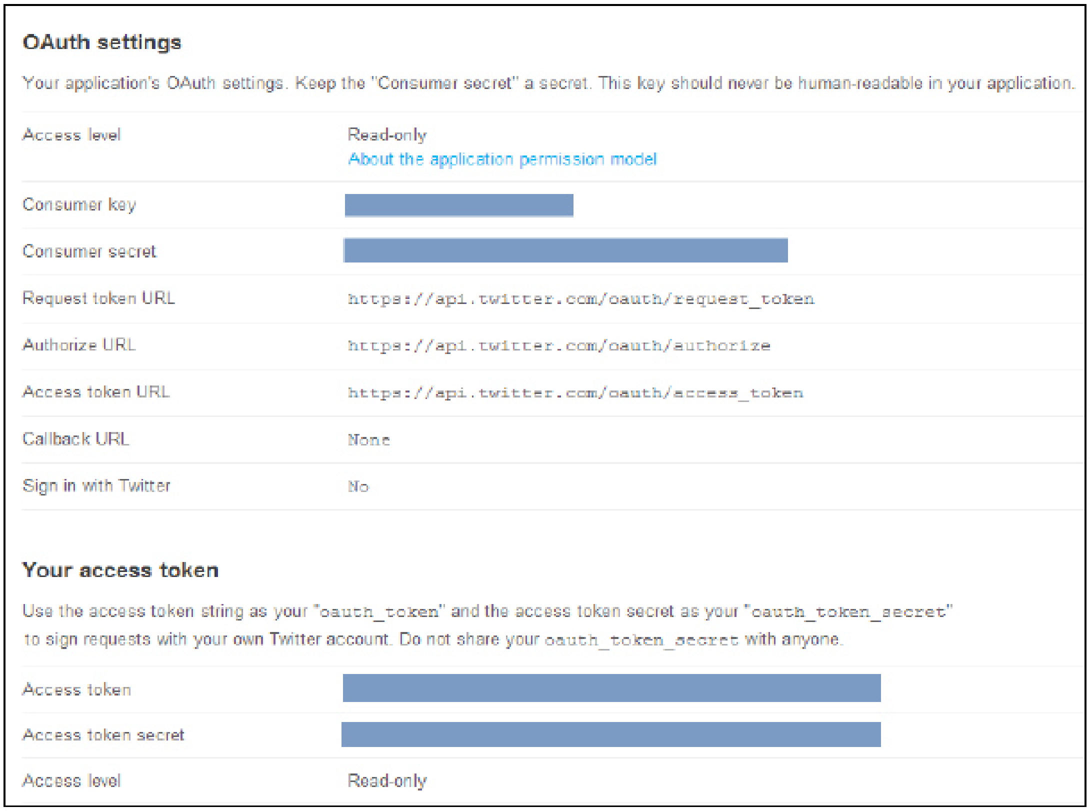
Sign in with your Twitter account

Please log in to access that page.

My applications

Looks like you haven't created any applications yet!

Figure 2 – Creating New App in Twitter



### OAuth settings

Your application's OAuth settings. Keep the "Consumer secret" a secret. This key should never be human-readable in your application.

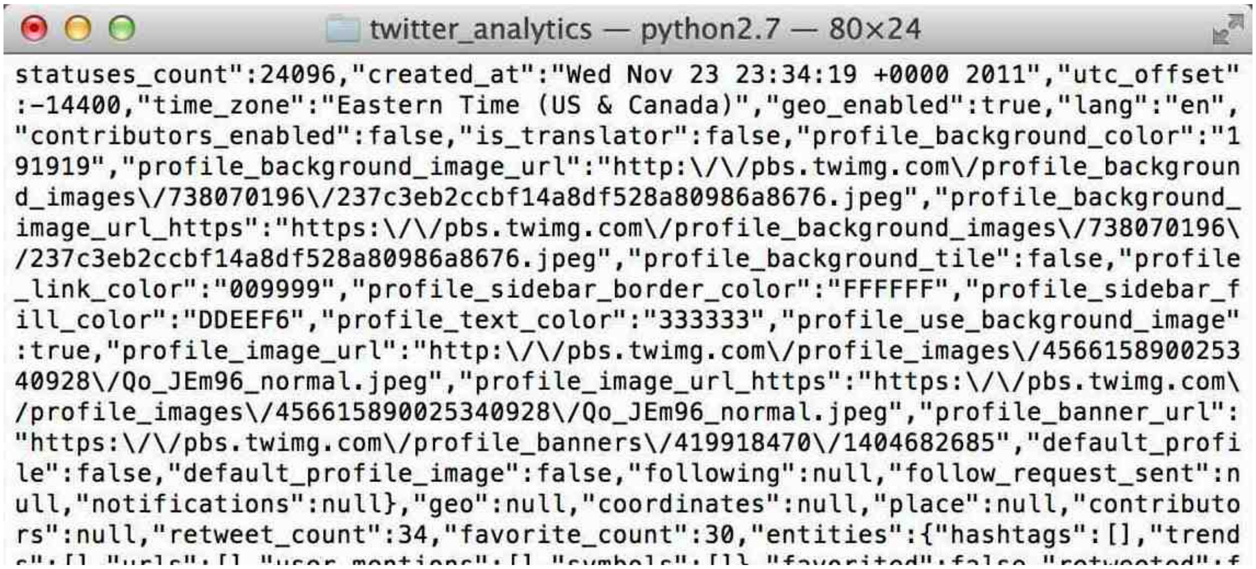
| | |
|---|---|
| Access level | Read-only<br>About the application permission model |
| Consumer key | [redacted] |
| Consumer secret | [redacted] |
| Request token URL | `https://api.twitter.com/oauth/request_token` |
| Authorize URL | `https://api.twitter.com/oauth/authorize` |
| Access token URL | `https://api.twitter.com/oauth/access_token` |
| Callback URL | None |
| Sign in with Twitter | No |

### Your access token

Use the access token string as your "`oauth_token`" and the access token secret as your "`oauth_token_secret`" to sign requests with your own Twitter account. Do not share your `oauth_token_secret` with anyone.

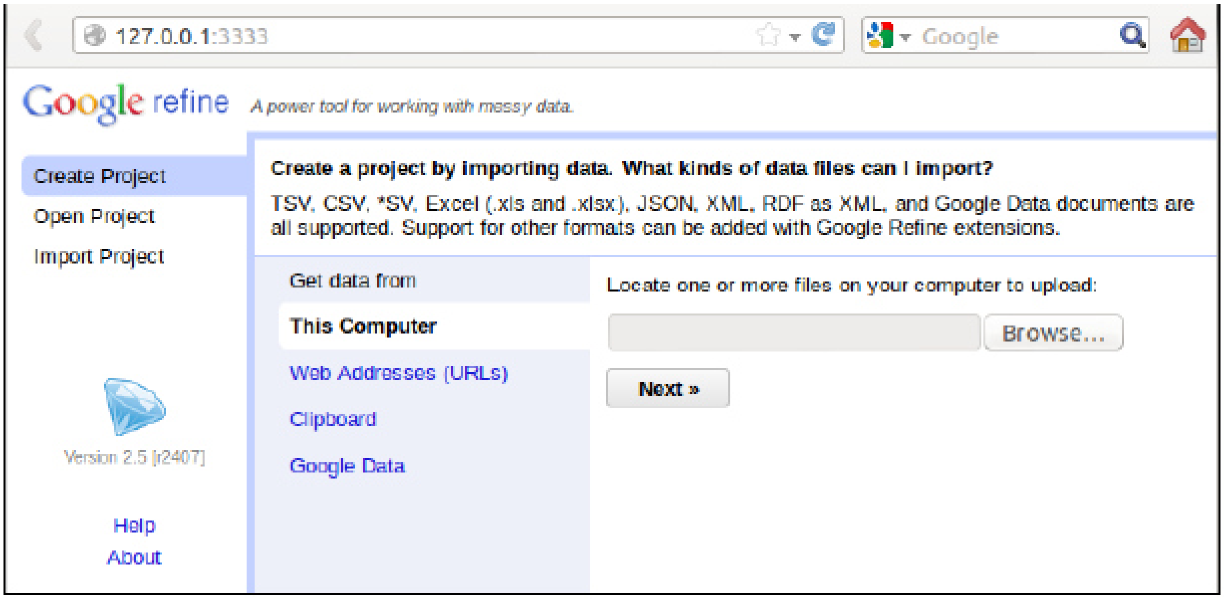
| | |
|---|---|
| Access token | [redacted] |
| Access token secret | [redacted] |
| Access level | Read-only |

Figure 3 – Generation of Authentication Tokens from Twitter



```
statuses_count":24096,"created_at":"Wed Nov 23 23:34:19 +0000 2011","utc_offset"
:-14400,"time_zone":"Eastern Time (US & Canada)","geo_enabled":true,"lang":"en",
"contributors_enabled":false,"is_translator":false,"profile_background_color":"1
91919","profile_background_image_url":"http:\/\/pbs.twimg.com\/profile_backgroun
d_images\/738070196\/237c3eb2ccbf14a8df528a80986a8676.jpeg","profile_background_
image_url_https":"https:\/\/pbs.twimg.com\/profile_background_images\/738070196\
/237c3eb2ccbf14a8df528a80986a8676.jpeg","profile_background_tile":false,"profile
_link_color":"009999","profile_sidebar_border_color":"FFFFFF","profile_sidebar_f
ill_color":"DDEEF6","profile_text_color":"333333","profile_use_background_image"
:true,"profile_image_url":"http:\/\/pbs.twimg.com\/profile_images\/4566158900253
40928\/Qo_JEm96_normal.jpeg","profile_image_url_https":"https:\/\/pbs.twimg.com\
/profile_images\/456615890025340928\/Qo_JEm96_normal.jpeg","profile_banner_url":
"https:\/\/pbs.twimg.com\/profile_banners\/419918470\/1404682685","default_profi
le":false,"default_profile_image":false,"following":null,"follow_request_sent":n
ull,"notifications":null},"geo":null,"coordinates":null,"place":null,"contributo
rs":null,"retweet_count":34,"favorite_count":30,"entities":{"hashtags":[],"trend
s":[],"urls":[],"user_mentions":[],"symbols":[]},"favorited":false,"retweeted":f
```

Figure 4 – Fetching Live Tweets from Twitter in JSON Format



**Create a project by importing data. What kinds of data files can I import?**

TSV, CSV, *SV, Excel (.xls and .xlsx), JSON, XML, RDF as XML, and Google Data documents are all supported. Support for other formats can be added with Google Refine extensions.

Locate one or more files on your computer to upload:

Figure 5 – Parsing of JSON using Google Refine

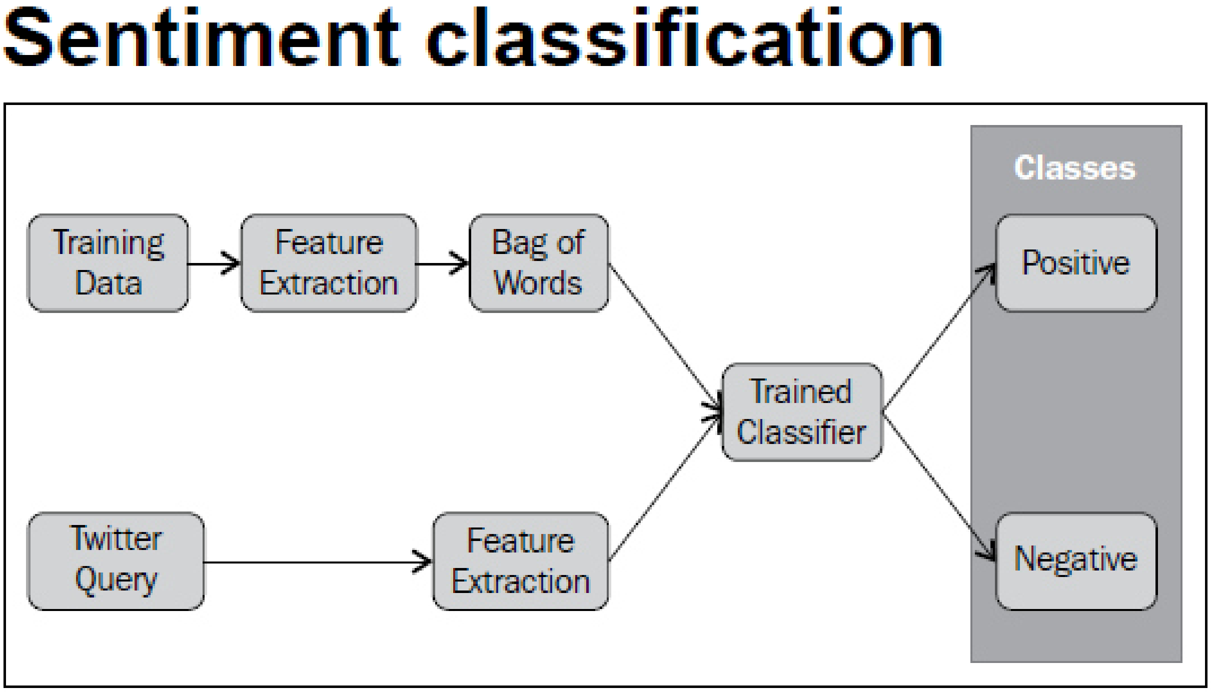


Figure 6 – Generation of Sentiments in Different Classes

```
from tweepy import Stream
from tweepy import OAuthHandler
from tweepy.streaming import StreamListener

#setting up the keys
consumer_key = 'XXXXXXXXXXXXXXXXXXXXX'
consumer_secret       = ' XXXXXXXXXXXXXXXXXXXXX '
access_token          = ' XXXXXXXXXXXXXXXXXXXXX '
access_secret         = ' XXXXXXXXXXXXXXXXXXXXX '
```

```
class TweetListener(StreamListener):
    # A listener handles tweets are the received from the stream.
    #This is a basic listener that just prints received tweets to standard output

    def on_data(self, data):
        print data
        return True

    def on_error(self, status):
        print status

#printing all the tweets to the standard output
auth = OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_secret)

stream = Stream(auth, TweetListener())
stream.filter(track=['research'])
```

Using this Python Code, the keyword ‘research’ is extracted from Twitter and the output is sent in JSON Format. JSON (JavaScript Object Notation) File Format is a special format that is used by many NoSQL and unstructured data handling engines. Once the JSON is obtained, after that using Refine Tool or any other tool the intelligence can be created and further predictions can be done.

## CONCLUSION

There is huge scope of research and development using Python scripts and specialized APIs for assorted applications including cyber security, data mining, Internet of Things, cloud simulation, grid implementation and many others. Python is one of the effective programming languages that can process and handle any type of data stream.

## REFERENCES

[1] Pang, B., & Lee, L. (2008). Opinion mining and sentiment analysis.*Foundations and trends in information retrieval*, *2*(1-2), 1-135.

[2] Pang, B., & Lee, L. (2004, July). A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd annual meeting on Association for Computational Linguistics* (p. 271). Association for Computational Linguistics.

[3] Wilson, T., Wiebe, J., & Hoffmann, P. (2005, October). Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the conference on human language technology and empirical methods in natural language processing* (pp. 347-354). Association for Computational Linguistics.

[4] Pak, A., & Paroubek, P. (2010, May). Twitter as a Corpus for Sentiment Analysis and Opinion Mining. In *LREC* (Vol. 10, pp. 1320-1326).

[5] Nasukawa, T., & Yi, J. (2003, October). Sentiment analysis: Capturing favorability using natural language processing. In *Proceedings of the 2nd international conference on Knowledge capture* (pp. 70-77). ACM.

[6] Liu, B. (2010). Sentiment analysis and subjectivity. *Handbook of natural language processing*, *2*, 627-666.

[7] Mullen, T., & Collier, N. (2004, July). Sentiment Analysis using Support Vector Machines with Diverse Information Sources. In *EMNLP* (Vol. 4, pp. 412-418).

[8] Agarwal, A., Xie, B., Vovsha, I., Rambow, O., & Passonneau, R. (2011, June). Sentiment analysis of twitter data. In *Proceedings of the Workshop on Languages in Social Media* (pp. 30-38). Association for Computational Linguistics.