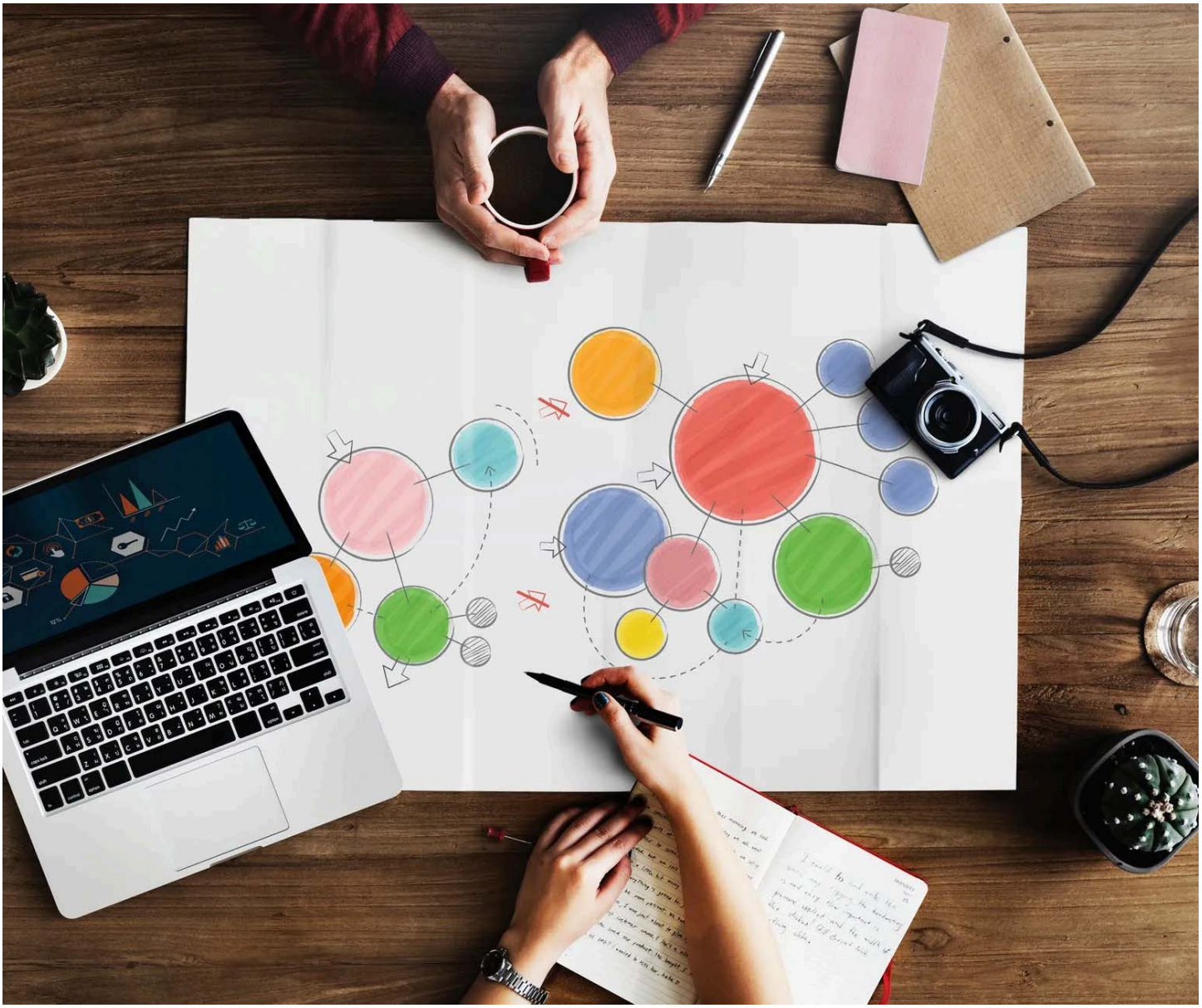


Fun with analyzing @BillGates tweets Twitter API's-Step by Step analysis from Extraction, Data Visualization, and Sentiment Analysis

Senthil E · Following
Published in Towards Data Science
11 min read · May 10, 2019

Listen Share More

This is the 2nd post of the web scraping and API's series. The first post is here. **Please check it out**. In this post, we can see how to extract the twitter data using Twitter API's and then do some basic visualization using word cloud, pie charts and then sentiment analysis using Textblob and Vader.

If you don't have Jupyter then please go ahead and **install anaconda**.

Let's do the following

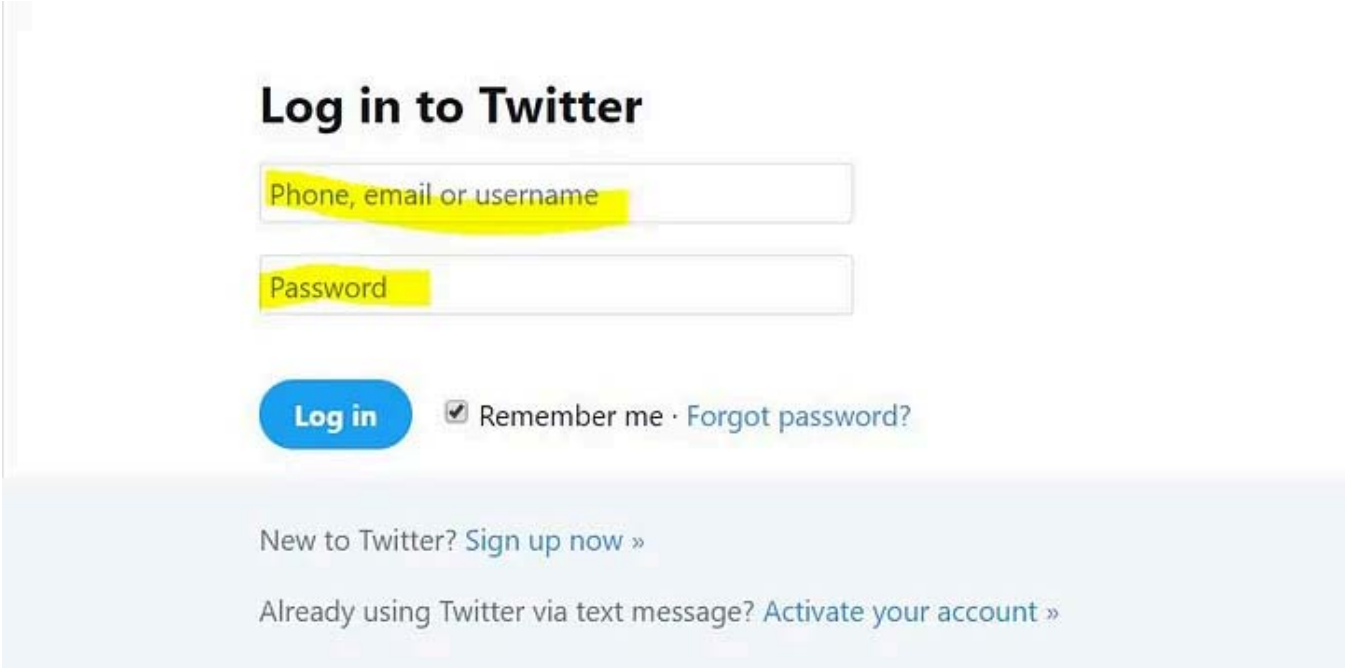
1. Create a twitter account or use your existing Twitter account.
2. Request for twitter developer access key
3. Extract real-time tweets using Twitter Streaming API
4. Extract history tweets using Twitter Search/Rest API using Tweepy
5. Load the data to a pandas data frame
6. Wordcloud
7. Scatter Text

8. Do some statistical analysis and visualization.
9. Sentiment Analysis using **Textblob**.
10. Sentiment Analysis using **VADER**
11. Sentiment Analysis using **Syuzhet(R)**

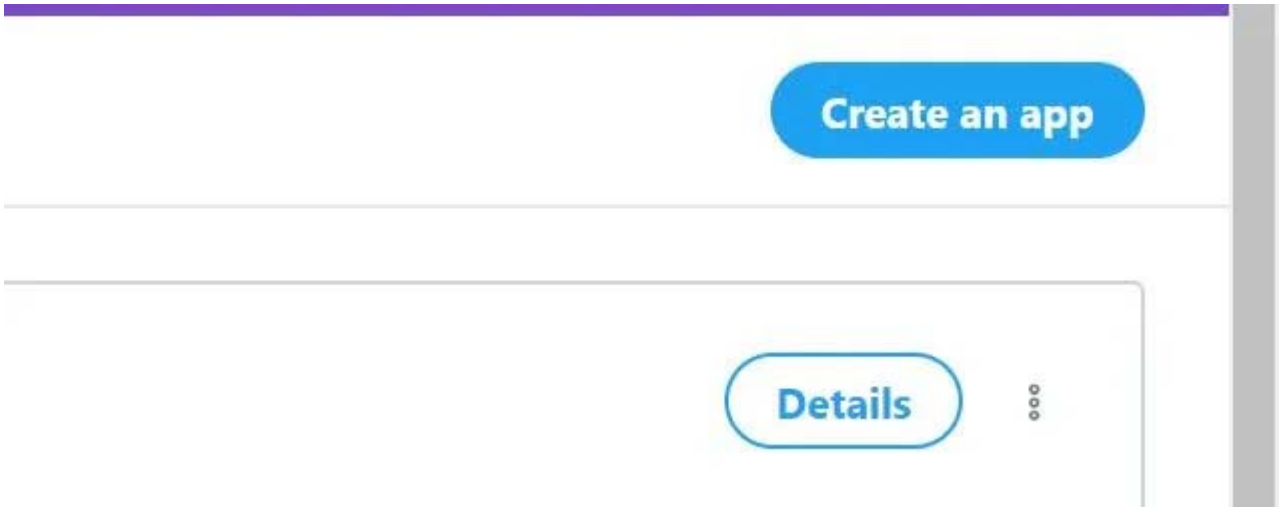
Before using the Twitter API we need a twitter account and then request for a developer key. First, create a twitter account if you don't have one. The steps to get the necessary developer keys are below

Watch this youtube video for creating a twitter account

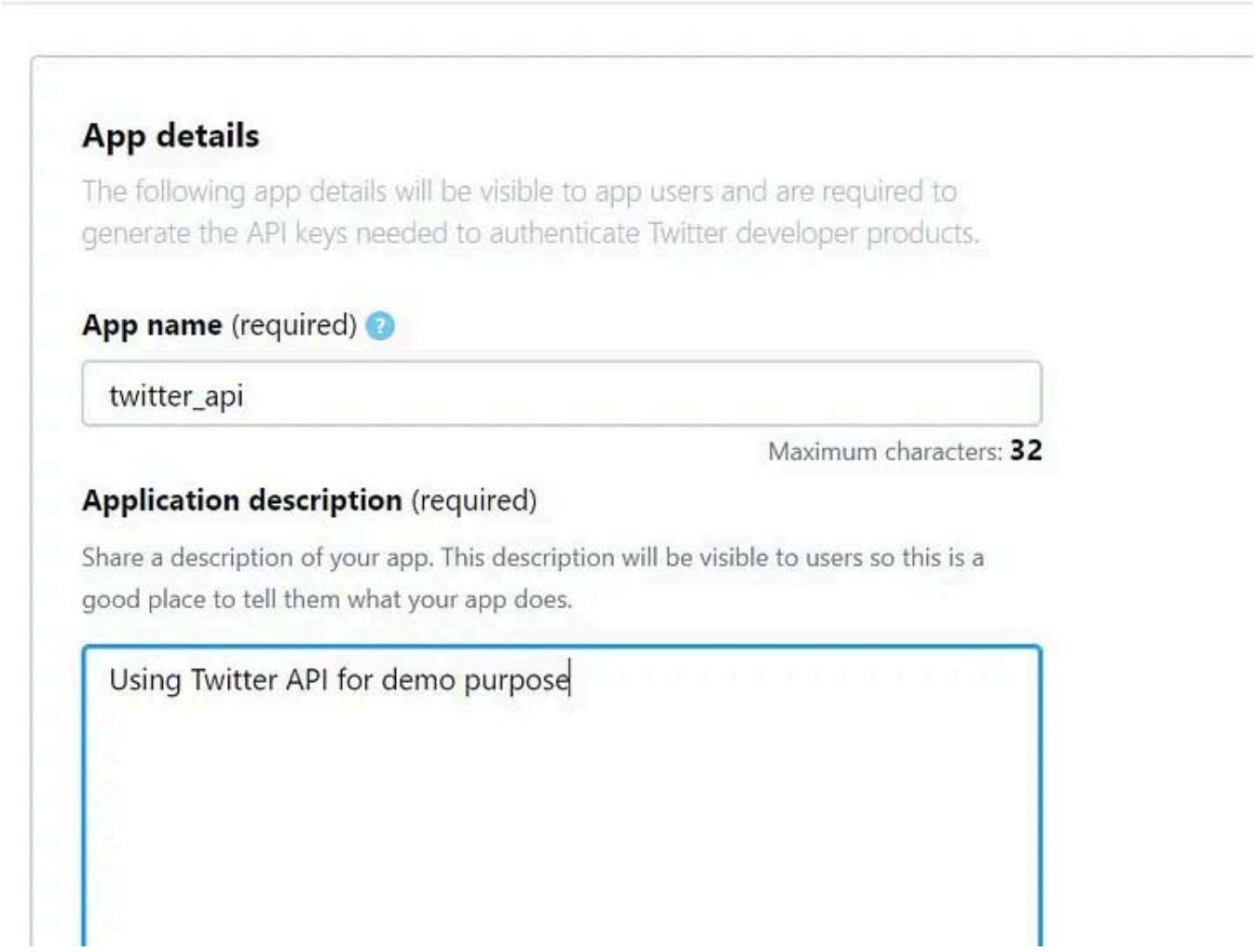
1. Goto to **https://developer.twitter.com/en/apps**.
2. Log in using your twitter account. If you don't have one then create one.



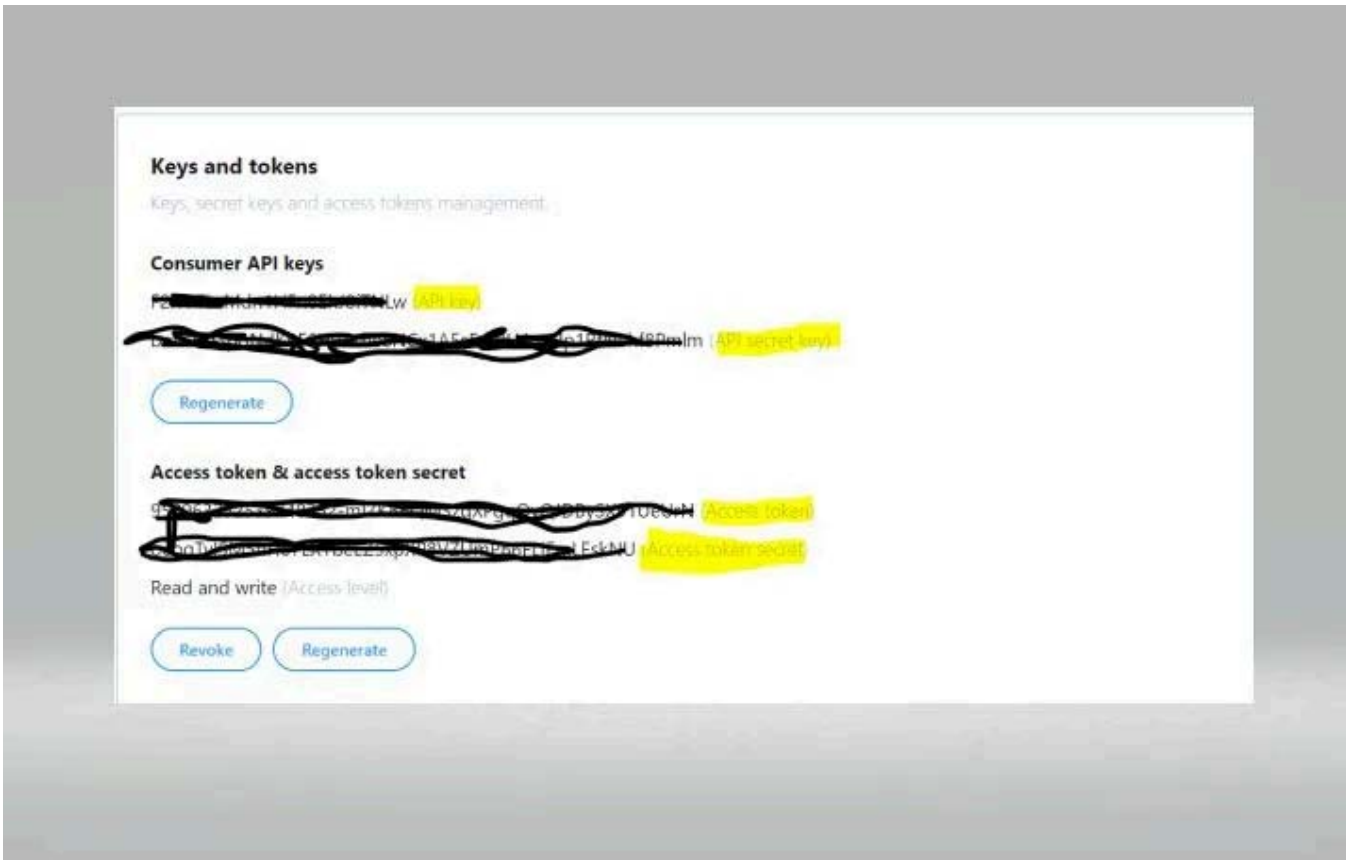
3.After logged in then create an app.



4. Fill out all the required fields.



5. You will get the following credentials



1. API Key
2. API secret key
3. Access token
4. Access token secret

Check out this youtube video on creating a twitter

Check out his documentation to get the Twitter Dev Keys

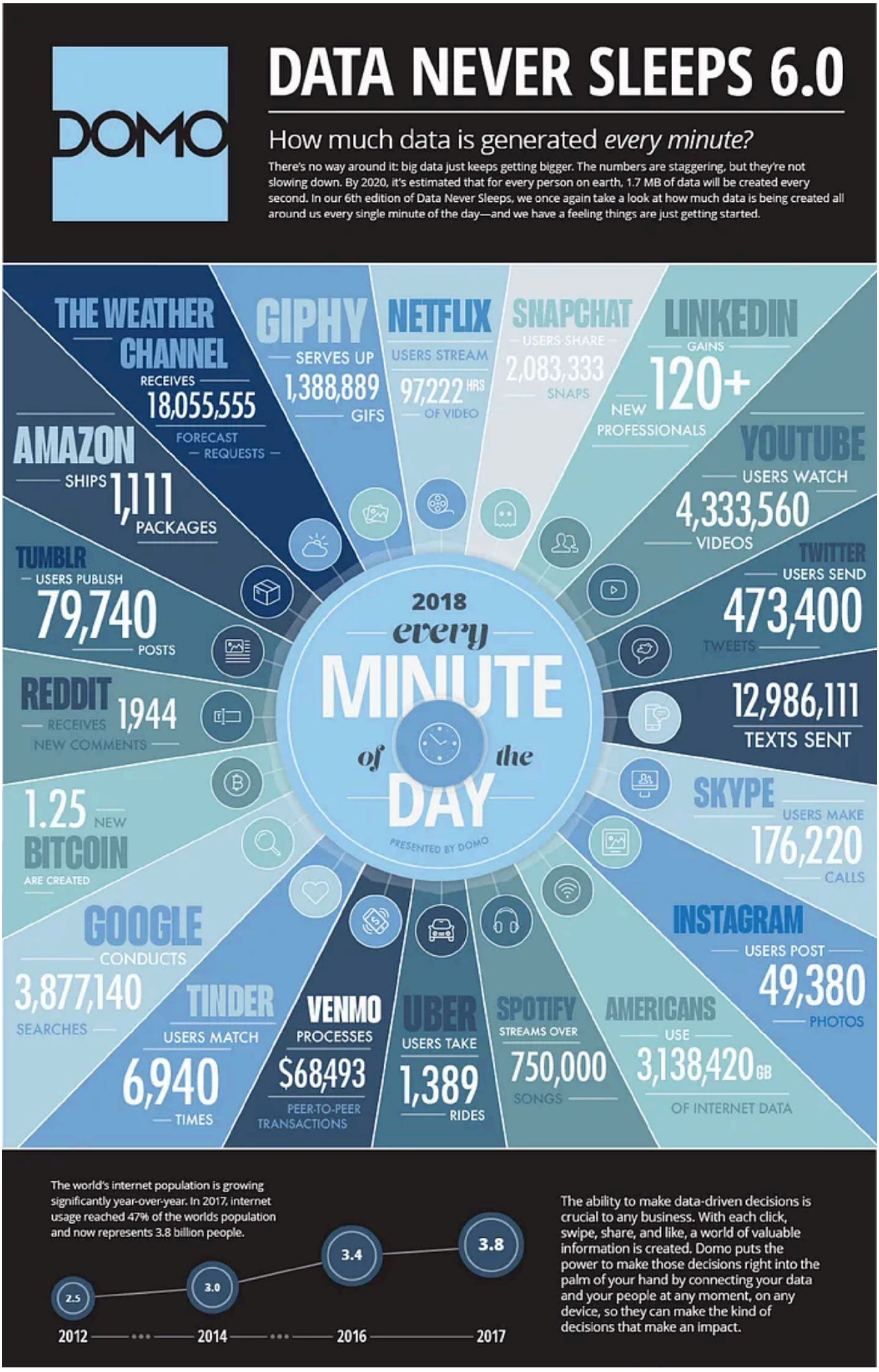
Twitter provides 2 API's

- *Streaming API*
- *Search API or Rest API*

From the twitter developer website, the available python related wrappers are

Python

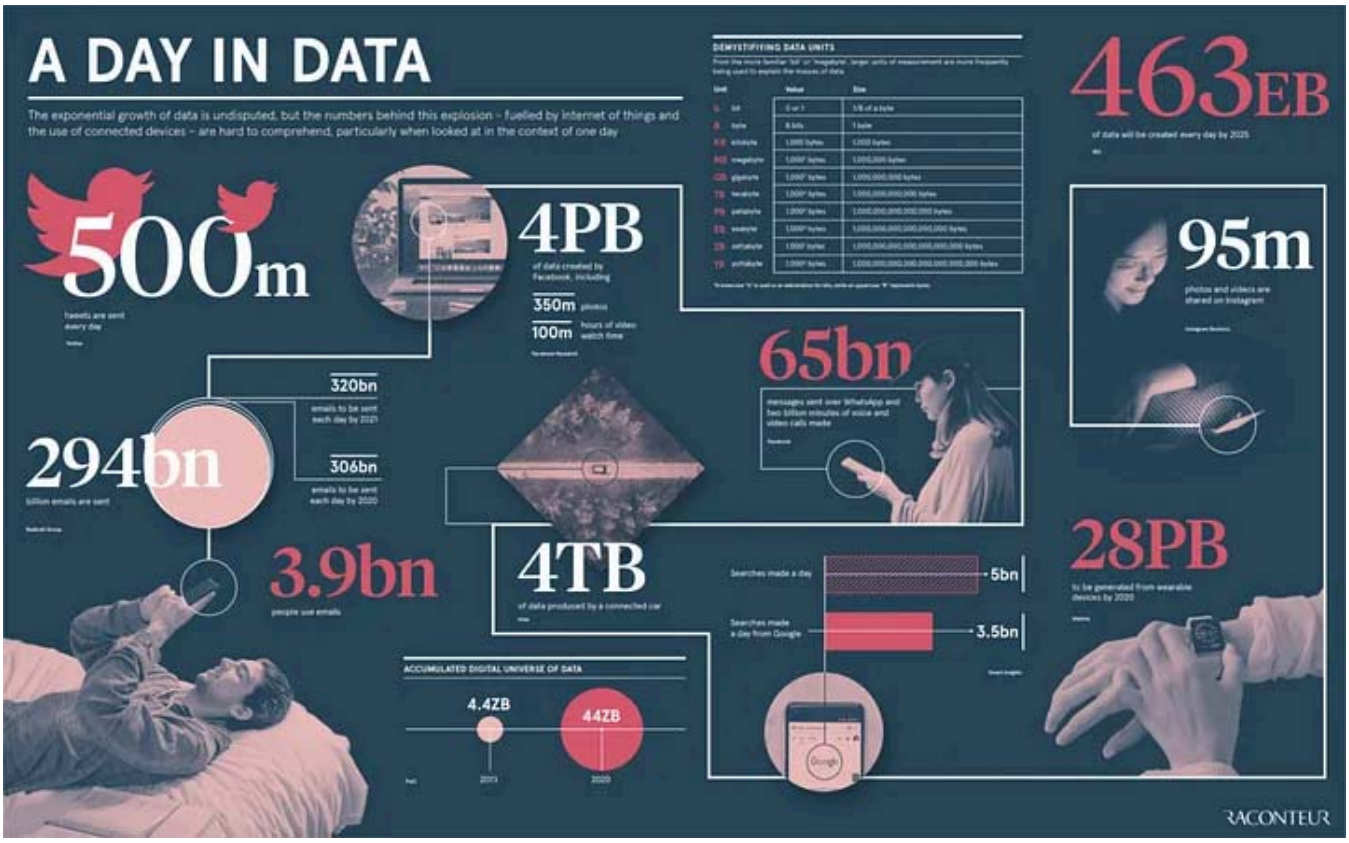
- python-twitter *maintained by @bear* — this library provides a pure Python interface for the Twitter API (documentation)
- tweepy *maintained by @applepie & more* — a Python wrapper for the Twitter API (documentation) (examples)
- TweetPony *by @Mezgrman* — A Python library aimed at simplicity and flexibility.
- Python Twitter Tools *by @sixohsix* — An extensive Python library for interfacing to the Twitter REST and streaming APIs (v1.0 and v1.1). Also features a command line Twitter client. Supports Python 2.6, 2.7, and 3.3+. (documentation)
- twitter-gobject *by @tchx84* — Allows you to access Twitter's 1.1 REST API via a set of GObject based objects for easy integration with your GLib2 based code. (examples)
- TwitterSearch *by @crw_koepp* — Python-based interface to the 1.1 Search API.
- twython *by @ryanmcgrath* — Actively maintained, pure Python wrapper for the Twitter API. Supports both normal and streaming Twitter APIs. Supports all v1.1 endpoints, including dynamic functions so users can make use of endpoints not yet in the library. (docs)
- TwitterAPI *by @boxnumber03* — A REST and Streaming API wrapper that supports python 2.x and python 3.x, TwitterAPI also includes iterators for both API's that are useful for processing streaming results as well as paged results.
- Birdy *by @sect2k* — "a super awesome Twitter API client for Python"


DOMO
DATA NEVER SLEEPS 6.0
How much data is generated every minute?
There's no way around it: big data just keeps getting bigger. The numbers are staggering, but they're not slowing down. By 2020, it's estimated that for every person on earth, 1.7 MB of data will be created every second. In our 6th edition of Data Never Sleeps, we once again take a look at how much data is being created all around us every single minute of the day—and we have a feeling things are just getting started.
THE WEATHER CHANNEL
RECEIVES 18,055,555 FORECAST REQUESTS
GIPHY
SERVES UP 1,388,889 GIFS
NETFLIX
USERS STREAM 97,222 HRS OF VIDEO
SNAPCHAT
USERS SHARE 2,083,333 SNAPS
LINKEDIN
GAINS 120+ NEW PROFESSIONALS
AMAZON
SHIPS 1,111 PACKAGES
YOUTUBE
USERS WATCH 4,333,560 VIDEOS
TUMBLR
USERS PUBLISH 79,740 POSTS
TWITTER
USERS SEND 473,400 TWEETS
2018
every
MINUTE
of the
DAY
PRESENTED BY DOMO
REDDIT
RECEIVES 1,944 NEW COMMENTS
12,986,111 TEXTS SENT
1.25 NEW BITCOIN ARE CREATED
SKYPE
USERS MAKE 176,220 CALLS
GOOGLE
CONDUCTS 3,877,140 SEARCHES
INSTAGRAM
USERS POST 49,380 PHOTOS
TINDER
USERS MATCH 6,940 TIMES
VENMO
PROCESSES $68,493 PEER-TO-PEER TRANSACTIONS
UBER
USERS TAKE 1,389 RIDES
SPOTIFY
STREAMS OVER 750,000 SONGS
AMERICANS
USE 3,138,420 GB OF INTERNET DATA
The world's internet population is growing significantly year-over-year. In 2017, internet usage reached 47% of the worlds population and now represents 3.8 billion people.
2.5
3.0
3.4
3.8
2012
2014
2016
2017
The ability to make data-driven decisions is crucial to any business. With each click, swipe, share, and like, a world of valuable information is created. Domo puts the power to make those decisions right into the palm of your hand by connecting your data and your people at any moment, on any device, so they can make the kind of decisions that make an impact.

- **6000** tweets every second
- **473,500** tweets per minute.
- **500 million** tweets are sent every day
- **200 billion** tweets per year.
- **326 million** people use Twitter every month
- **2.5 quintillion bytes** of data generated every day.
- **50%** of the earth's population is on social media, with a total of **2,789 billion** people.
- By **2025**, it's estimated that **463 exabytes** of data will be created each day globally — that's the equivalent of **212,765,957 DVDs** per day!
- **Social media accounts for 33%** of the total time spent online.

Source: Domo.com


2018 This Is What Happens In An Internet Minute
facebook
973,000 Logins
18 Million Text Messages
YouTube
4.3 Million Videos Viewed
375,000 Apps Downloaded
174,000 Scrolling Instagram
481,000 Tweets Sent
1.1 Million Swipes
tinder
187 Million Emails Sent
936,073 Views
twitch
67 Voice-First Devices Shipped
amazon echo
38 Million Messages
25,000 GIFs Sent via Messenger
2.4 Million Snaps Created
$862,823 Spent Online
NETFLIX
266,000 Hours Watched
Google
3.7 Million Search Queries
60 SECONDS
Created By:
@LoriLewis
@OfficiallyChadd



Source:visualcapitalist

Abbreviation	Unit	Value	Size (in bytes)
b	bit	0 or 1	1/8 of a byte
B	bytes	8 bits	1 byte
KB	kilobytes	1,000 bytes	1,000 bytes
MB	megabyte	$1{,}000^2$ bytes	1,000,000 bytes
GB	gigabyte	$1{,}000^3$ bytes	1,000,000,000 bytes
TB	terabyte	$1{,}000^4$ bytes	1,000,000,000,000 bytes
PB	petabyte	$1{,}000^5$ bytes	1,000,000,000,000,000 bytes
EB	exabyte	$1{,}000^6$ bytes	1,000,000,000,000,000,000 bytes
ZB	zettabyte	$1{,}000^7$ bytes	1,000,000,000,000,000,000,000 bytes
YB	yottabyte	$1{,}000^8$ bytes	1,000,000,000,000,000,000,000,000 bytes

Source:visualcapitalist

1.Streaming API

The **Twitter Search API** and **Twitter Streaming API** work well for individuals that just want to access Twitter data for **light analytics or statistical analysis.**

Twitter's Streaming API is a push of data as tweets happens in near real-time. The major drawback of the Streaming API is that Twitter's Streaming API provides only a

sample of tweets that are occurring. The actual percentage of total tweets users receive with Twitter's Streaming API varies heavily based on the criteria users request and the current traffic. Studies have estimated that using Twitter's Streaming API users can expect to **receive 1% of the tweets in near real-time. (reference website)**

Before starting install all the required libraries — command prompt using pip

Please check out more on PIP

```
pip install vaderSentiment
pip install nltk
pip install Textblob
pip install numby
pip install pandas
```

After that import all the required libraries

```
#Declare the libraries needed
import tweepy
import pandas as pd
import sys
import csv
from wordcloud import WordCloud, STOPWORDS
import matplotlib.pyplot as plt
import re
from PIL import Image
import pandas_profiling
```

tw19 hosted with ❤ by GitHub view raw

```
# The keys have been displayed here becuase they are not to be shared.
# The value inside quotes must be replaced by your keys if you are using this.
import twitter
#api = twitter.Api(consumer_key='consumer_key',consumer_secret='consumer_secret',
#access_token_key='access_token',access_token_secret='access_token_secret')
api = twitter.Api(consumer_key='enter your consumer key',
                      consumer_secret='enter your consumer secret',
                      access_token_key='enter your access token',
                      access_token_secret='enter your scret token')
```

tw1 hosted with ❤ by GitHub view raw

First, enter all your credentials. Login into twitter account and go to the app and then fill all the above information.

```
#Print your credentials and check
print (api.VerifyCredentials())
```

tw2 hosted with ❤ by GitHub view raw

Just do print and check your credentials

You can filter by

1. User ID
2. User Name
3. Query by keywords
4. Geo Location

Filter by location and select the tweets I am filtering it by location = San Francisco

```
for tweet in api.GetStreamFilter(locations = ['-122.75,36.8,-121.75,37.8']):
    print (tweet)
    break
```

tw3 hosted with ❤ by GitHub view raw

locations

A comma-separated list of longitude,latitude pairs specifying a set of bounding boxes to filter Tweets by. Only geolocated Tweets falling within the requested bounding boxes will be included—unlike the Search API, the user's location field is not used to filter Tweets.

Each bounding box should be specified as a pair of longitude and latitude pairs, with the southwest corner of the bounding box coming first. For example:

Parameter value	Tracks Tweets from...
-122.75,36.8,-121.75,37.8	San Francisco
-74,40,-73,41	New York City
-122.75,36.8,-121.75,37.8,-74,40,-73,41	San FranciscoOR New York City

Just one tweet contains so much information

{'created_at': 'Sat May 04 21:14:50 +0000 2019', 'id': 1124784475001384960, 'id_str': '1124784475001384960', 'text': "@AlaskaAi
r it doesn't seem Alaska airlines is up to meeting the norm for int'l flights-just nuts and chips on your… https://t.co/WWoEDdo
LZU", 'source': 'Twitter for iPhone', 'truncated': True, 'in_re
ply_to_status_id': None, 'in_reply_to_status_id_str': None, 'in_reply_to_user_id': 13192972, 'in_reply_to_user_id_str': '131929
72', 'in_reply_to_screen_name': 'AlaskaAir', 'user': {'id': 770271189264543744, 'id_str': '770271189264543744', 'name': 'Vietna
m-era Vet-PDX', 'screen_name': 'PdxVeteran', 'location': 'Portland, OR', 'url': None, 'description': 'Retired Air Force dude. B
ike & walk a ton every day. My church is the outdoors, and I spend lots of time in church 😂 I follow the news & vote 4 progre
ssives !', 'translator_type': 'none', 'protected': False, 'verified': False, 'followers_count': 600, 'friends_count': 1748, 'li
sted_count': 0, 'favourites_count': 5209, 'statuses_count': 4257, 'created_at': 'Mon Aug 29 14:45:37 +0000 2016', 'utc_offset':
None, 'time_zone': None, 'geo_enabled': True, 'lang': 'en', 'contributors_enabled': False, 'is_translator': False, 'profile_bac
kground_color': '000000', 'profile_background_image_url': 'http://abs.twimg.com/images/themes/theme1/bg.png', 'profile_backgrou
nd_image_url_https': 'https://abs.twimg.com/images/themes/theme1/bg.png', 'profile_background_tile': False, 'profile_link_colo
r': '981CEB', 'profile_sidebar_border_color': '000000', 'profile_sidebar_fill_color': '000000', 'profile_text_color': '000000',
'profile_use_background_image': False, 'profile_image_url': 'http://pbs.twimg.com/profile_images/770276640706301952/QG8qVKem_no
rmal.jpg', 'profile_image_url_https': 'https://pbs.twimg.com/profile_images/770276640706301952/QG8qVKem_normal.jpg', 'default_p
rofile': False, 'default_profile_image': False, 'following': None, 'follow_request_sent': None, 'notifications': None}, 'geo':
 None, 'coordinates': None, 'place': {'id': 'fbd6d2f5a4e4a15e', 'url': 'https://api.twitter.com/1.1/geo/id/fbd6d2f5a4e4a15e.jso
n', 'place_type': 'admin', 'name': 'California', 'full_name': 'California, USA', 'country_code': 'US', 'country': 'United State
s', 'bounding_box': {'type': 'Polygon', 'coordinates': [[[-124.482003, 32.528832], [-124.482003, 42.009519], [-114.131212, 42.0
09519], [-114.131212, 32.528832]]]}, 'attributes': {}}, 'contributors': None, 'is_quote_status': False, 'extended_tweet': {'ful
l_text': "@AlaskaAir it doesn't seem Alaska airlines is up to meeting the norm for int'l flights-just nuts and chips on your fl
ights from Costa Rica! That's indefensible!!!", 'display_text_range': [0, 163], 'entities': {'hashtags': [], 'urls': [], 'user
_mentions': [{'screen_name': 'AlaskaAir', 'name': 'Alaska Airlines', 'id': 13192972, 'id_str': '13192972', 'indices': [0, 1
0]}], 'symbols': []}}, 'quote_count': 0, 'reply_count': 0, 'retweet_count': 0, 'favorite_count': 0, 'entities': {'hashtags':
 [], 'urls': [{'url': 'https://t.co/WWoEDdoLZU', 'expanded_url': 'https://twitter.com/i/web/status/1124784475001384960', 'displ
ay_url': 'twitter.com/i/web/status/1…', 'indices': [116, 139]}], 'user_mentions': [{'screen_name': 'AlaskaAir', 'name': 'Alaska
Airlines', 'id': 13192972, 'id_str': '13192972', 'indices': [0, 10]}], 'symbols': []}, 'favorited': False, 'retweeted': False,
 'filter_level': 'low', 'lang': 'en', 'timestamp_ms': '1557004490982'}

Check out all the tweet objects available

The JSON object looks like

```
{
    "created_at": "Thu Apr 06 15:24:15 +0000 2017",
    "id_str": "850006245121695744",
    "text": "1\/ Today we\u2019re sharing our vision for the future of the Twitter API platform!\
    "user": {
      "id": 2244994945,
      "name": "Twitter Dev",
      "screen_name": "TwitterDev",
      "location": "Internet",
      "url": "https:\/\/dev.twitter.com\/",
      "description": "Your official source for Twitter Platform news, updates & events. Need tech
    },
    "place": {
    },
    "entities": {
      "hashtags": [
      ],
      "urls": [
        {
          "url": "https:\/\/t.co\/XweGngmxlP",
          "unwound": {
            "url": "https:\/\/cards.twitter.com\/cards\/18ce53wgo4h\/3xo1c",
            "title": "Building the Future of the Twitter API Platform"
          }
        }
      ],
      "user_mentions": [
      ]
    }
}
```

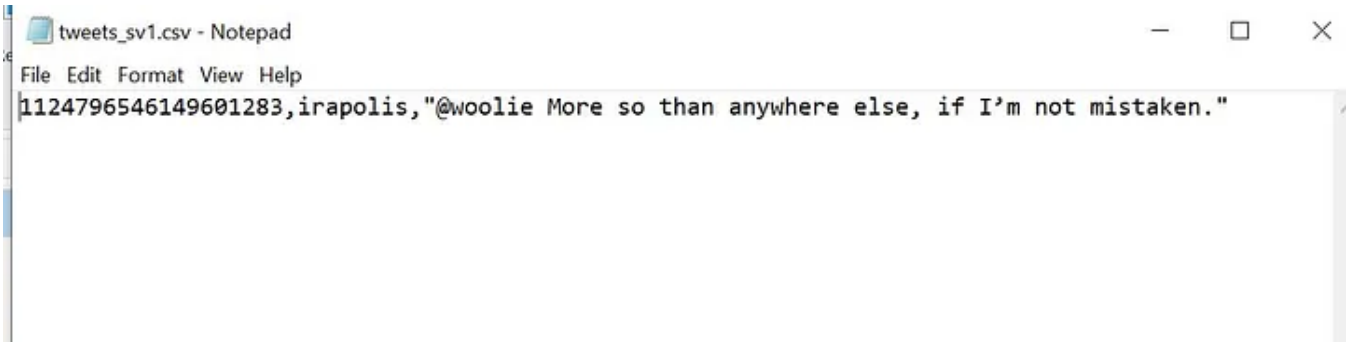

tw18 hosted with ❤ by GitHub view raw

You can also save it to the CSV file

```
import csv
geo_stream = api.GetStreamFilter(locations = ['-122.75,36.8,-121.75,37.8'])
#    print (tweet)
with open('tweets_sv1.csv', 'w+') as csv_file:
    csv_writer = csv.writer(csv_file)
    for line in geo_stream:
        #Signal that the line represents a tweet
        tweet = twitter.Status.NewFromJsonDict(line)
        row = [tweet.id, tweet.user.screen_name, tweet.text]
        csv_writer.writerow(row)
break
```

tw6 hosted with ❤ by GitHub view raw

The output of the 1 tweet with selected info



For more info on the streaming API

The output from the Twitter API is in JSON format. So what is JSON

Please check this youtube video.

Please check the data camp article for more info on JSON file and python

Now we can download the file in JSON format

```
# lets save the json data to a file: stream.json
# "\n" is new line. We save the data to a file,
# with each new line we save all the information related to one tweet
import json
f = open('./stream1.json', 'w')
for tweet in api.GetStreamFilter(locations = ['-122.75,36.8,-121.75,37.8']):
    f.write(json.dumps(tweet))
    f.write('\n')
    print (tweet)
```

tw7 hosted with ❤ by GitHub view raw

Upload the JSON file to a list.

```
#Upload the json file and append it to a list
import json
data=[]
with open('./streamingData1.json', 'r') as jsonFile:
    for line in jsonFile:
        data.append(json.loads(line))
print ("Total number of tweets loaded: ", len(data))
print(data)
```

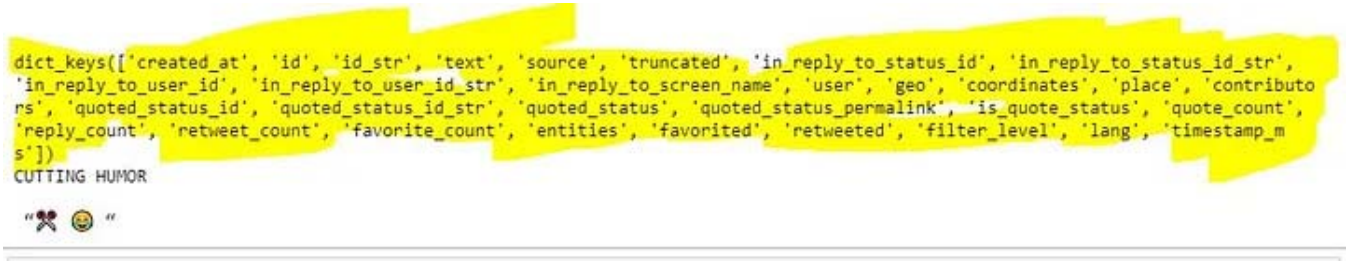
tw8 hosted with ❤ by GitHub view raw

Check the keys and the tweet itself

```
print (data[0].keys())   # properties
print (data[0]["text"])  # how to access the  tweet message itself
```

tw9 hosted with ❤ by GitHub view raw

The output looks like



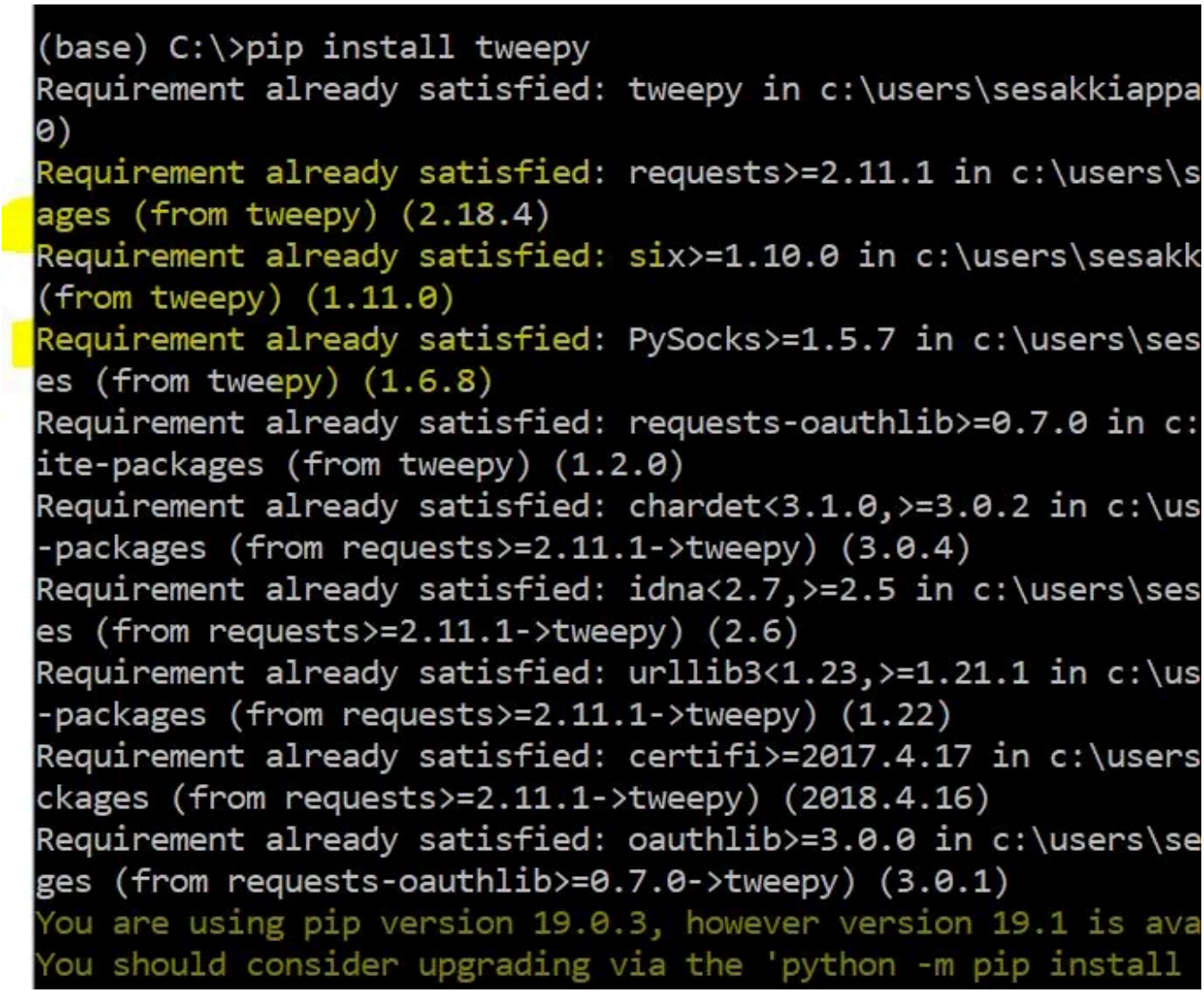
The dictionary consists of a lot of information like created, user ID, retweeted, timestamp, etc. Just read from the dictionary what information you need and then proceed with the text mining and visualization.

2. Search API or REST API

Twitter's Search API gives you access to a data set that already exists from tweets that have occurred. For an individual user, the maximum number of tweets you can receive is the last **3,200 tweets**, regardless of the query criteria. With a specific keyword, you can typically only poll the last 5,000 tweets per keyword. You are further limited by the number of requests you can make in a certain time period. The Twitter request limits have changed over the years but are currently limited to **180 requests in a 15 minute period.**

Let's use **tweepy** for search API.

1. pip install tweepy at the command prompt



Since I already installed it says that Requirement already satisfied.

2. Import the libraries needed.

```
#Declare the libraries needed
import tweepy
import pandas as pd
import sys
import csv
from wordcloud import WordCloud, STOPWORDS
import matplotlib.pyplot as plt
import re
from PIL import Image
import pandas_profiling
```

tw19 hosted with ❤ by GitHub view raw

3. Provide all the credentials you got before

```
# Fill the credentials
# following the above mentioned procedure.
api = twitter.Api(consumer_key='Enter your consumer key',
                      consumer_secret='Enter your consumer secret',
                      access_token_key='Enter your access token',
                      access_token_secret='enter your access token')

```

tw10 hosted with ❤ by GitHub view raw

4. Now we are going to extract the tweets The main function is below. I am extracting the tweets of Bill Gates. ***The maximum tweets allowed is 3200***. I can't extract the whole tweet history of the user if they have tweeted more than 3200 tweets.

```
import tweepy
import pandas as pd
import sys
import csv

# Fill the X's with the credentials obtained by
# following the above mentioned procedure.

api = twitter.Api(consumer_key='Please enter your credentials',
                      consumer_secret='Please enter your credentials',
                      access_token_key='Please enter your credentials',
                      access_token_secret='Please enter your credentials')

# Function to extract tweets
def get_tweets(username):

        # Authorization to consumer key and consumer secret
        auth = tweepy.OAuthHandler(consumer_key, consumer_secret)

        # Access to user's access key and access secret
        auth.set_access_token(access_token_key, access_token_secret)

        # Calling api
        api = tweepy.API(auth)
        #set count to however many tweets you want - max count is 3200 and this doesn't have an
#        number_of_tweets = 5000

        tfile = []
        for tweet in tweepy.Cursor(api.user_timeline, screen_name = username).items():

        #username, tweet id, date/time, text
            tfile.append([username, tweet.id_str,tweet.source, tweet.created_at,tweet.retweet_c

        #write to a new csv file from the array of tweets
        outfile = username + "_tweets_V1.csv"
        print ("writing to " + outfile)
        with open(outfile, 'w+') as file:
            writer = csv.writer(file, delimiter=',')
            writer.writerow(['User_Name', 'Tweet_ID', 'Source', 'Created_date','Retweet_count',
            writer.writerows(tfile)
# user name
get_tweets("@BillGates")
```

tw11 hosted with ❤ by GitHub view raw

20. Bill Gates
@BillGates
Bio: Sharing things I'm learning through my foundation work and other interests.
Location: Seattle, WA

followers	following	tweets
47,134,550	196	3,057

The downloaded file looks like below

```
User_Name,Tweet_ID,Source,created_date,retweet_count,favorite_count,tweet

@BillGates,1124761460964458497,Sprinklr,2019-05-04 19:43:24,492,4536,Melinda and I love meeting with people who are trying to change the world. Here are
four of the people who inspired… https://t.co/MrmYszRNMG

@BillGates,1124310352848674816,Sprinklr,2019-05-03 13:50:51,905,4574,What does a "dirt detective" do for a living? I'm eager to watch this talk from soil
scientist Asmeret Asefaw Berhe… https://t.co/Rw3GrZ1V3q

@BillGates,1124070620722094087,Twitter Web Client,2019-05-02 21:58:14,301,2241,"A few months ago, Fred contributed a piece to my blog which outlined why
educating the next generation of African l… https://t.co/jPqGy5OhR3"

@BillGates,1124070618553618432,Twitter Web Client,2019-05-02 21:58:14,711,3570,"By the end of the century, almost half of the young people in the world will
```

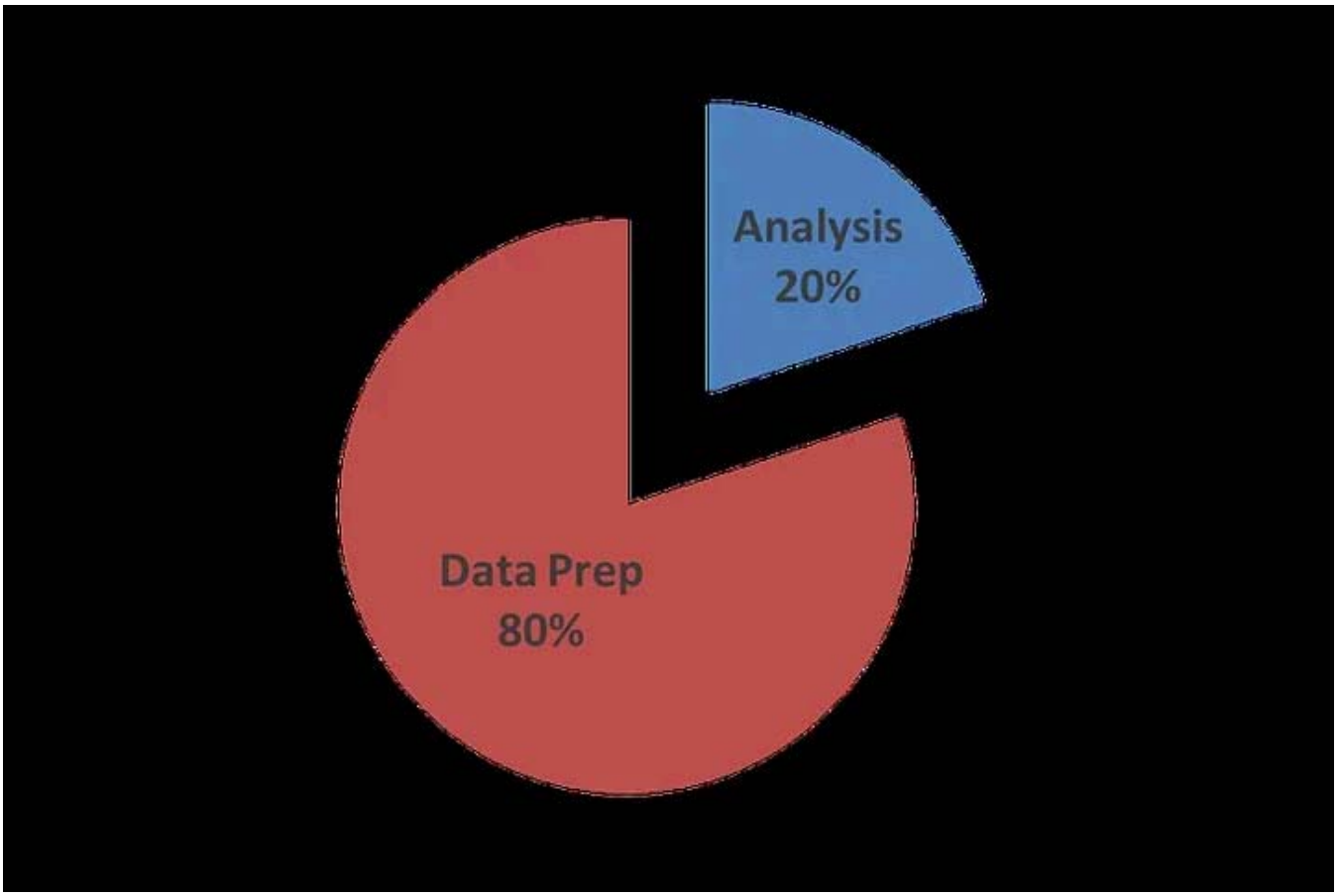
5. Now upload the data into a panda data frame for data visualization and sentiment analysis

```
1  #Read the file and also assign it to dataframe
2  import pandas as pd
3  bg= pd.read_csv("@BillGates_tweets_V4.csv",encoding='utf-8')
4  bg.head(10)
```

tw12 hosted with ❤ by GitHub view raw

The data looks like below in the dataframe

	User_Name	Tweet_ID	Source	created_date	retweet_count	favorite_count	tweet
0	@BillGates	1124761460964458497	Sprinklr	2019-05-04 19:43:24	492	4536	Melinda and I love meeting with people who are...
1	@BillGates	1124310352848674816	Sprinklr	2019-05-03 13:50:51	905	4574	What does a "dirt detective" do for a living? ...
2	@BillGates	1124070620722094087	Twitter Web Client	2019-05-02 21:58:14	301	2241	A few months ago, Fred contributed a piece to ...
3	@BillGates	1124070618553618432	Twitter Web Client	2019-05-02 21:58:14	711	3570	By the end of the century, almost half of the ...
4	@BillGates	1124070613344292871	Twitter Web Client	2019-05-02 21:58:13	576	3869	.@FredSwaniker has an ambitious plan to unlock...
5	@BillGates	1123977200708018176	Sprinklr	2019-05-02 15:47:01	521	4229	It's hard for me to overstate how brave people...
6	@BillGates	1123726002654515202	Twitter Web Client	2019-05-01 23:08:51	228	2815	Congratulations @SueDHellmann on your 5th anni...
7	@BillGates	1123625634943635457	Sprinklr	2019-05-01 16:30:01	608	2625	The world's historic progress to #endpolio wou...
8	@BillGates	1123263708258377730	Twitter Media Studio	2019-04-30 16:31:51	1122	5769	If you enjoyed the @Avengers this weekend, I t...
9	@BillGates	1123036364130406400	Sprinklr	2019-04-30 01:28:28	1592	7730	It's amazing to think about the creative ways ...



About 80% of the time is spent in preparing the data and only 20% in the analysis.

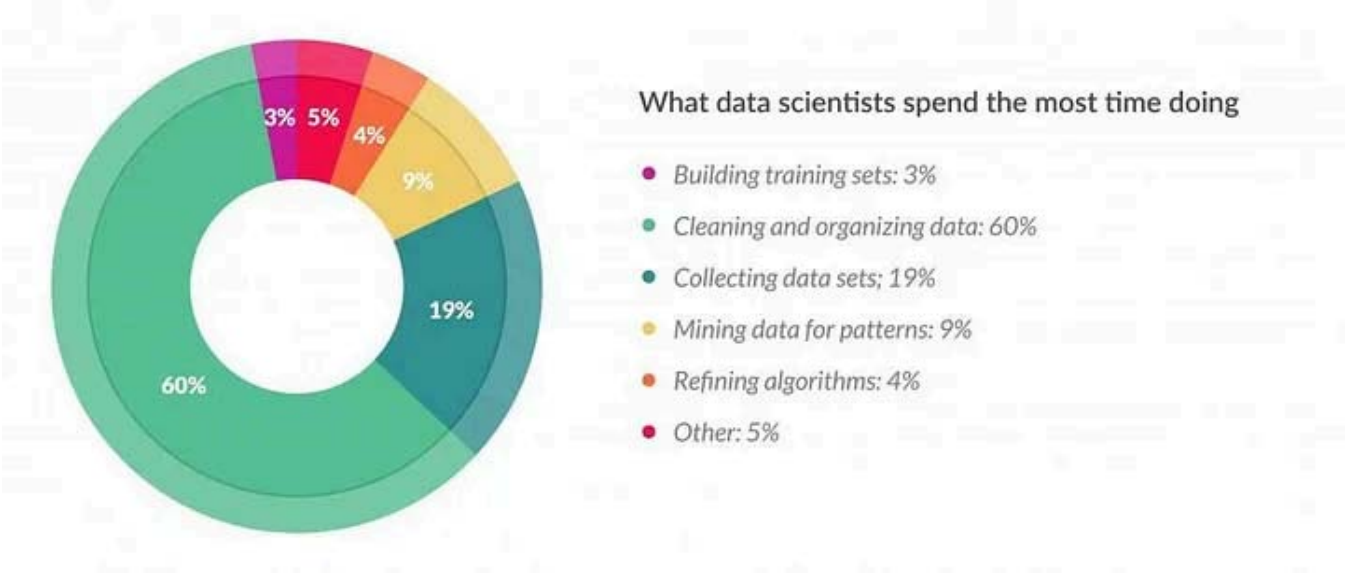
6. Do some basic cleaning of the tweets

```
bg2 = []
import re
pattern1 = re.compile(" ' # S % & ' ( ) * + , - . / : ; < = >  @ [ / ] ^ _ { | } ~")
pattern2 = re.compile("@[A-Za-z0-9]+")
pattern3 = re.compile("https?://[A-Za-z0-9./]+")

for item in bg1:
    tweet = re.sub(pattern1, "", item)   # version 1 of the tweet
    tweet = re.sub(pattern2, "", tweet)
    tweet = re.sub(pattern3, "", tweet)
    bg2.append(tweet)

bg3 = pd.DataFrame(bg2,columns = ['tweet'])
```

tw13 hosted with ❤ by GitHub view raw



Source: Forbes

7. Now we can do basic data analysis and visualization

Let's do a word cloud. **Check the documentation for more info.**

The only required parameter is the text and all others are optional.

```
import numpy as np
import pandas as pd
import matplotlib as mpl
import matplotlib.pyplot as plt
%matplotlib inline
from subprocess import check_output
from wordcloud import WordCloud, STOPWORDS

mpl.rcParams['figure.figsize']=(16.0,10.0)
mpl.rcParams['font.size']=12
mpl.rcParams['savefig.dpi']=1400
mpl.rcParams['figure.subplot.bottom']=.1

stopwords = set(STOPWORDS)
text = " ".join(tweet for tweet in bg3.tweet)
print ("There are {} words in the combination of all tweets.".format(len(text)))

wordcloud = WordCloud(
                          background_color='white',
                          stopwords=stopwords,
                          max_words=200,
                          max_font_size=40,
                          random_state=42
                         ).generate(str(text))

print(wordcloud)
fig = plt.figure(1)
plt.imshow(wordcloud)
plt.axis('off')
plt.show()
fig.savefig("word1.png", dpi=1400)
```

tw14 hosted with ❤ by GitHub view raw

The output is

You can also do a mask

The original picture is

and the wordcloud

and the code is

```
There are 292731 words in the combination of all tweets.
```

Scatter Text

A tool for finding distinguishing terms in small-to-medium-sized corpora, and presenting them in a sexy, interactive scatter plot with non-overlapping term labels. Exploratory data analysis just got more fun.

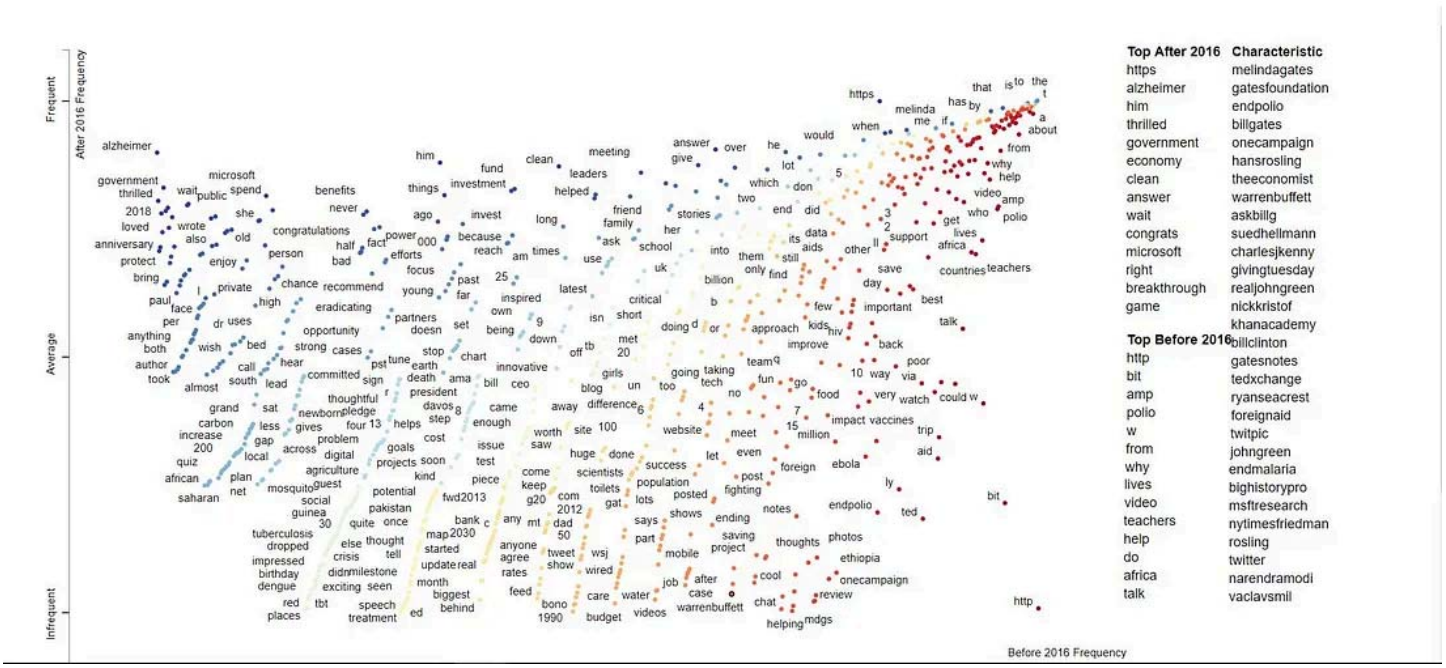
Check out the GitHub link

The visual is stunning. I tried to just use the tweets to do a scatter text. Please refer the GitHub for sample code and scenarios. I think definitely my scatter text can be

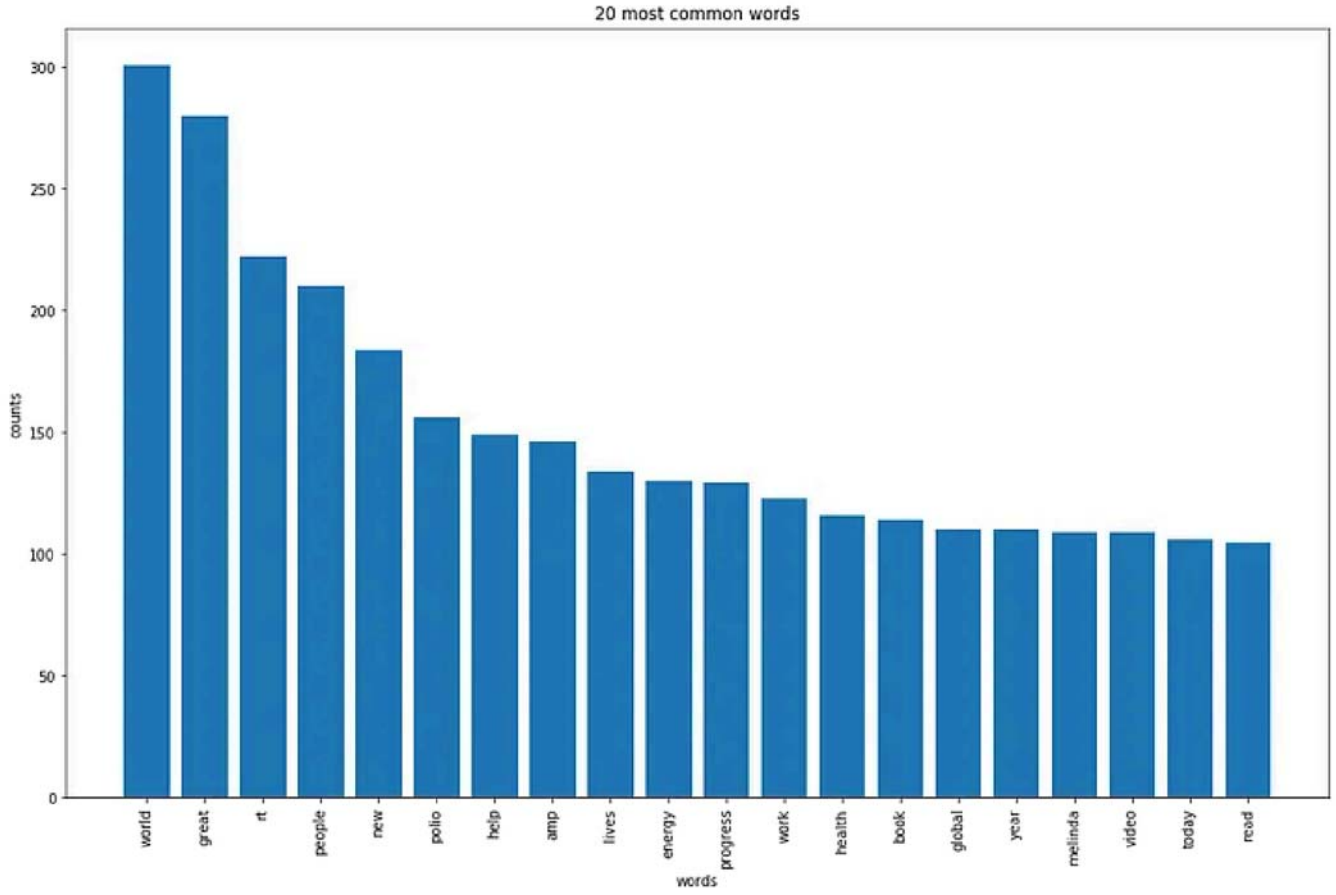
improved.

The code is

I love the graph in the HTML display. I have attached a screenshot.



Top 20 common words in the tweets



Now do some basic analysis. Find the most liked tweets and most retweeted tweets.

The code is

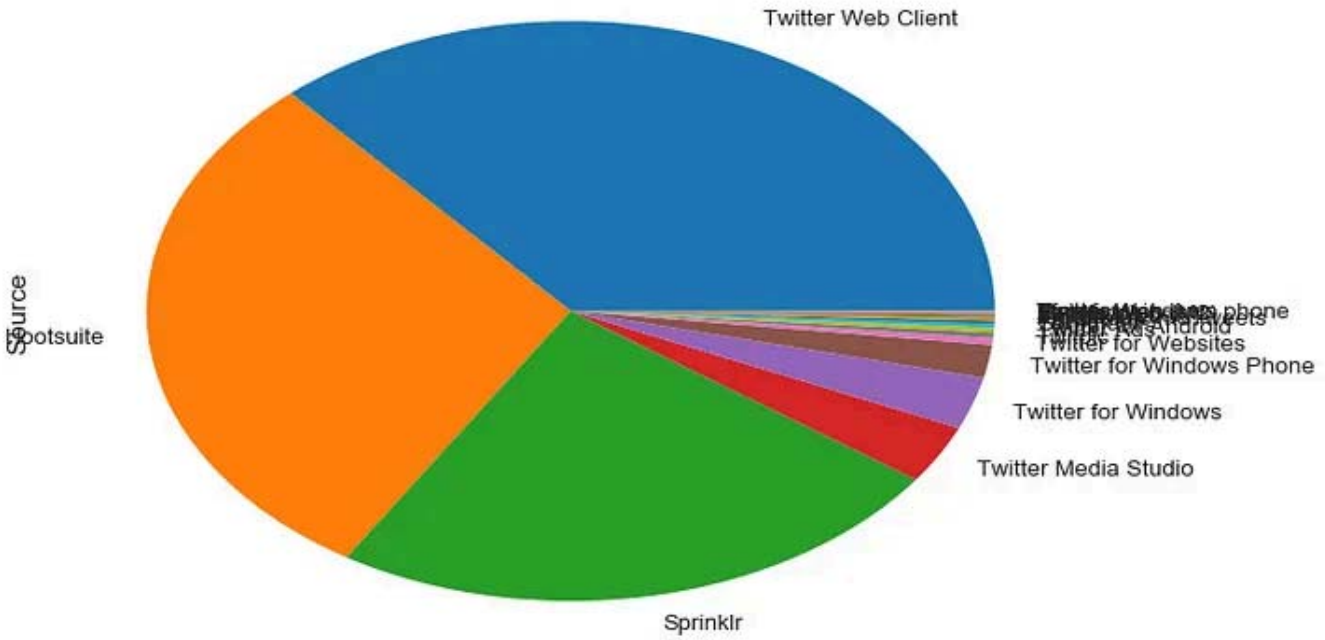
The output

```
The tweet with more likes is:
Congratulations to the Indian government on the first 100 days of
@AyushmanNHA. It's great to see how many people h…
https://t.co/mGXaz16H7W
Number of likes: 56774
The tweet with more retweets is:
RT @WarrenBuffett: Warren is in the house.
Number of retweets: 39904
```

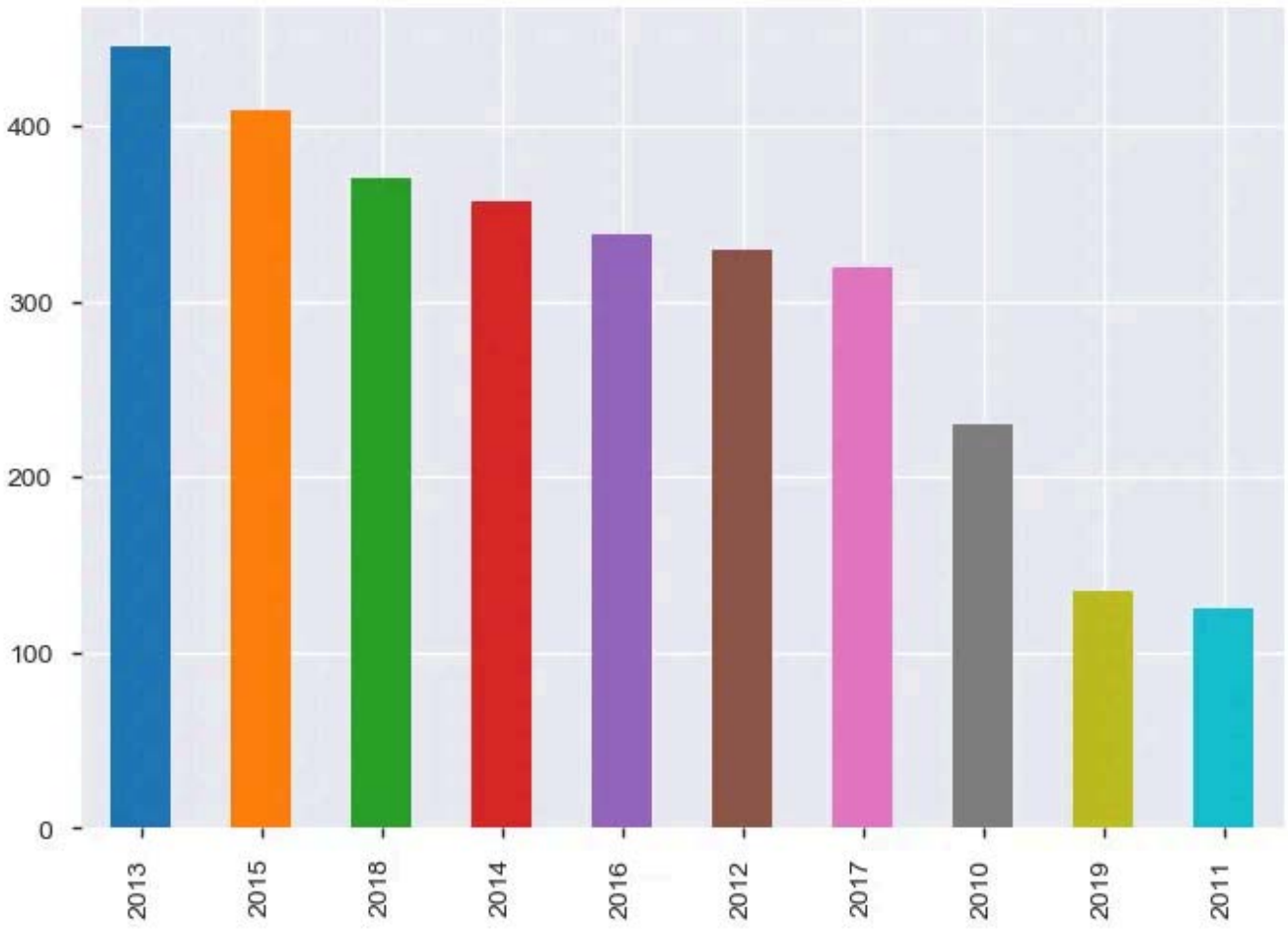
The most liked tweet is

Source of the tweet — Obviously no *iPhone and Android* 🤣

```
Twitter Web Client           1115
Hootsuite                     907
Sprinklr                      733
Twitter Media Studio          100
Twitter for Windows            89
Twitter for Windows Phone      56
Twitter for Websites           11
Twitpic                        10
Twitter Ads                     8
Spredfast                       6
Twitter for Android             6
Panoramic moTweets              5
Mobile Web                      3
Seesmic Look                    3
Yfrog                           1
Vine for windows phone          1
Facebook                        1
Twitter Web App                 1
Mobile Web (M2)                 1
Name: Source, dtype: int64
```



Tweets by Year

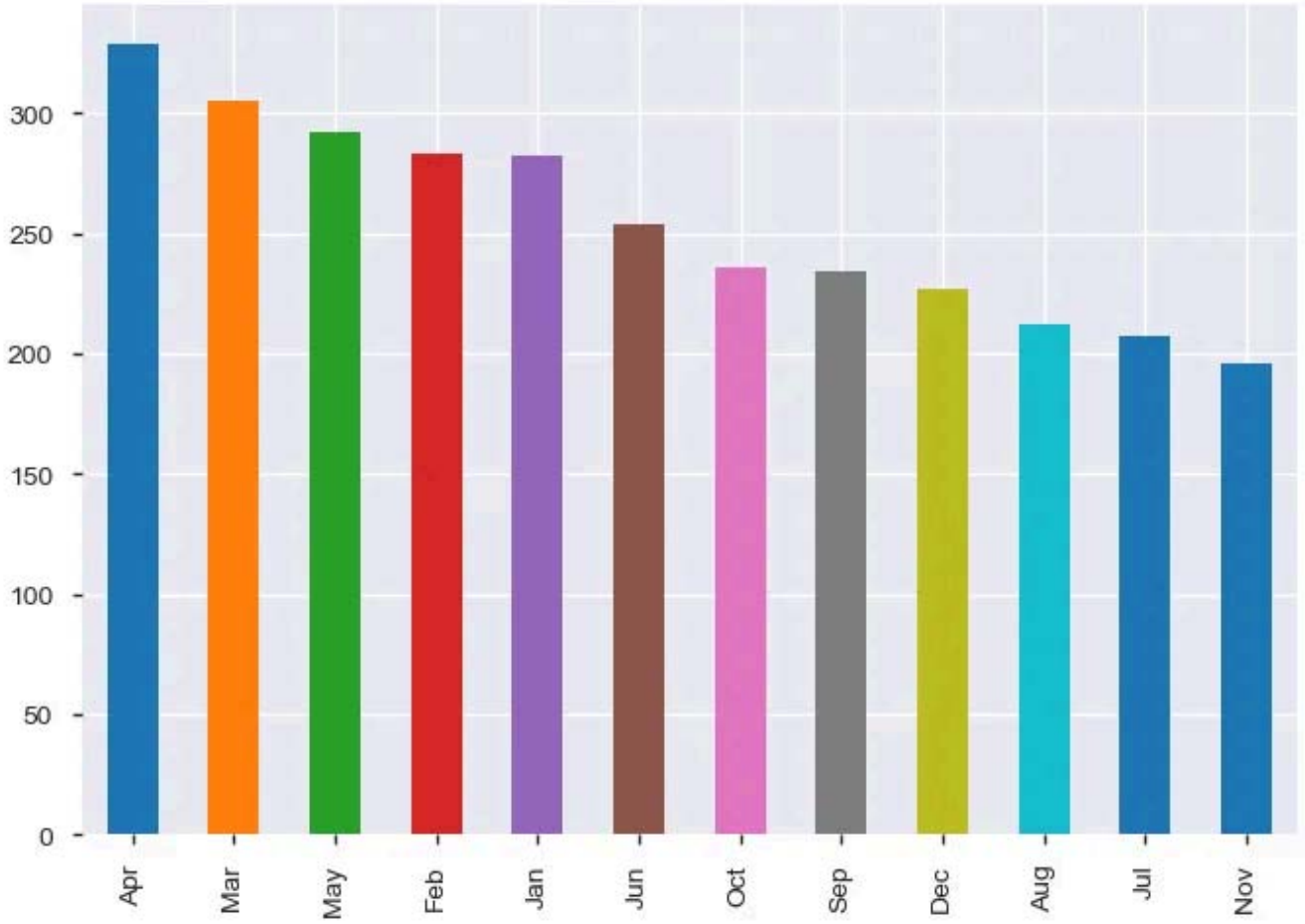


2013 is the year in which Gates tweeted 445 tweets and 409 in 2015. The lowest is 2011 with 125 tweets.

```
2013    445
2015    409
2018    370
2014    357
2016    338
2012    329
2017    319
2010    230
2019    135
2011    125
```

Tweets by Month

April is the month he tweeted the most with **329** and the lowest is November with 196

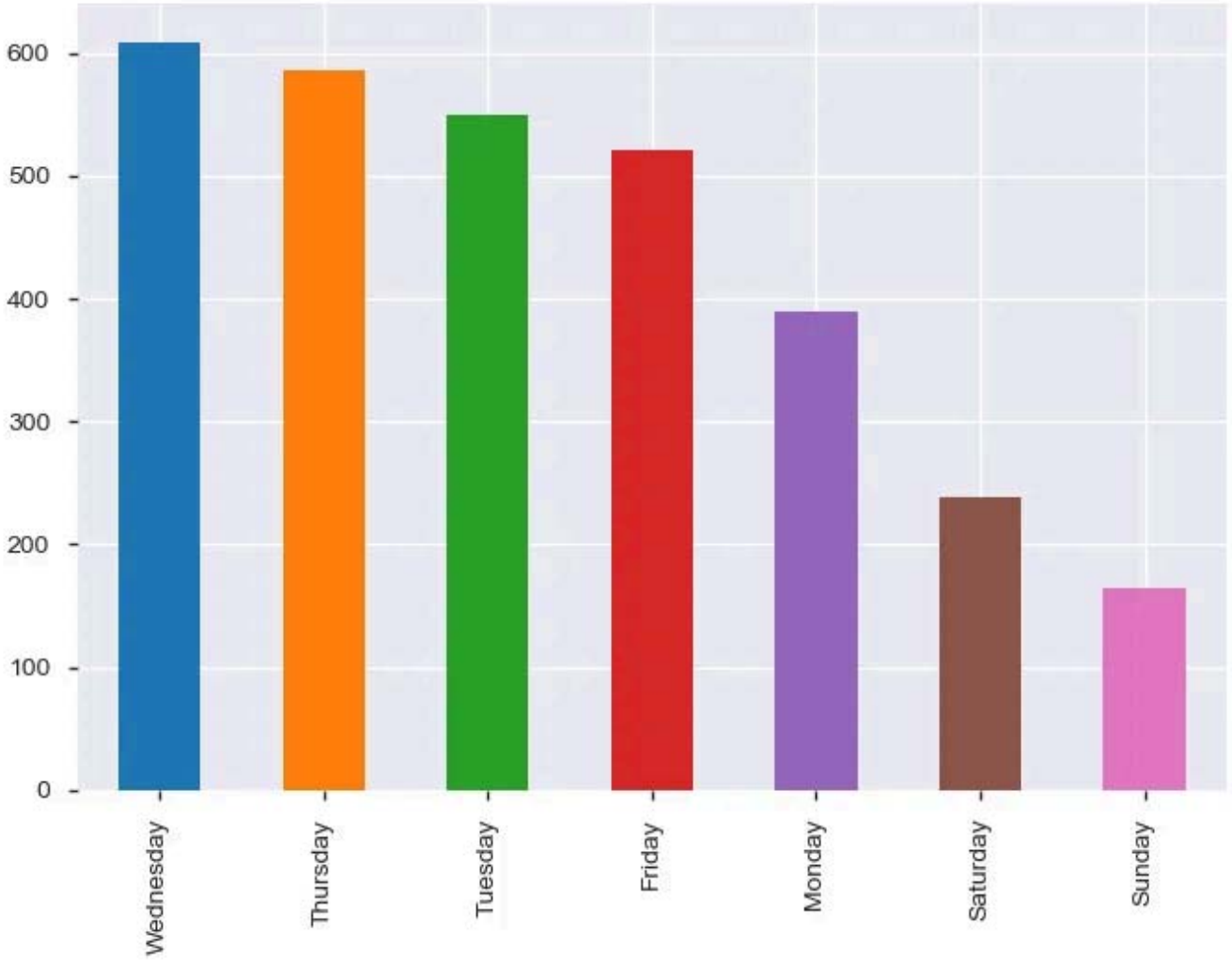



```
Apr    329
Mar    305
May    292
Feb    283
Jan    282
Jun    254
Oct    236
```

```
Sep    234
Dec    227
Aug    212
Jul    207
Nov    196
```

Tweets by day

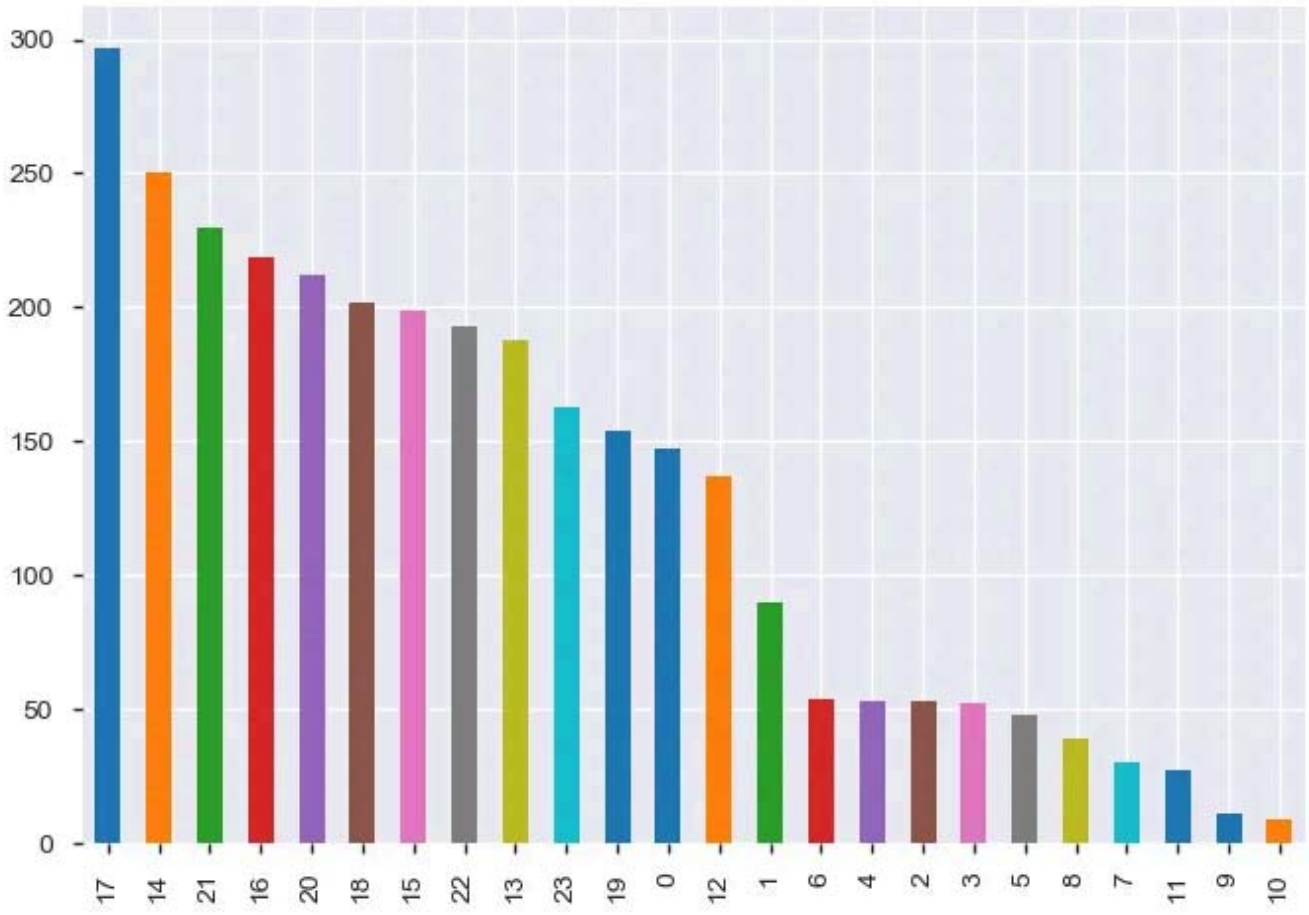
Wednesday is the day in which he tweeted almost **609** tweets and next is Thursday with 586. The weekends are the lowest 😄




```
Wednesday    609
Thursday     586
Tuesday      549
Friday       521
Monday       389
Saturday     239
Sunday       164
```

Tweets by the hour:

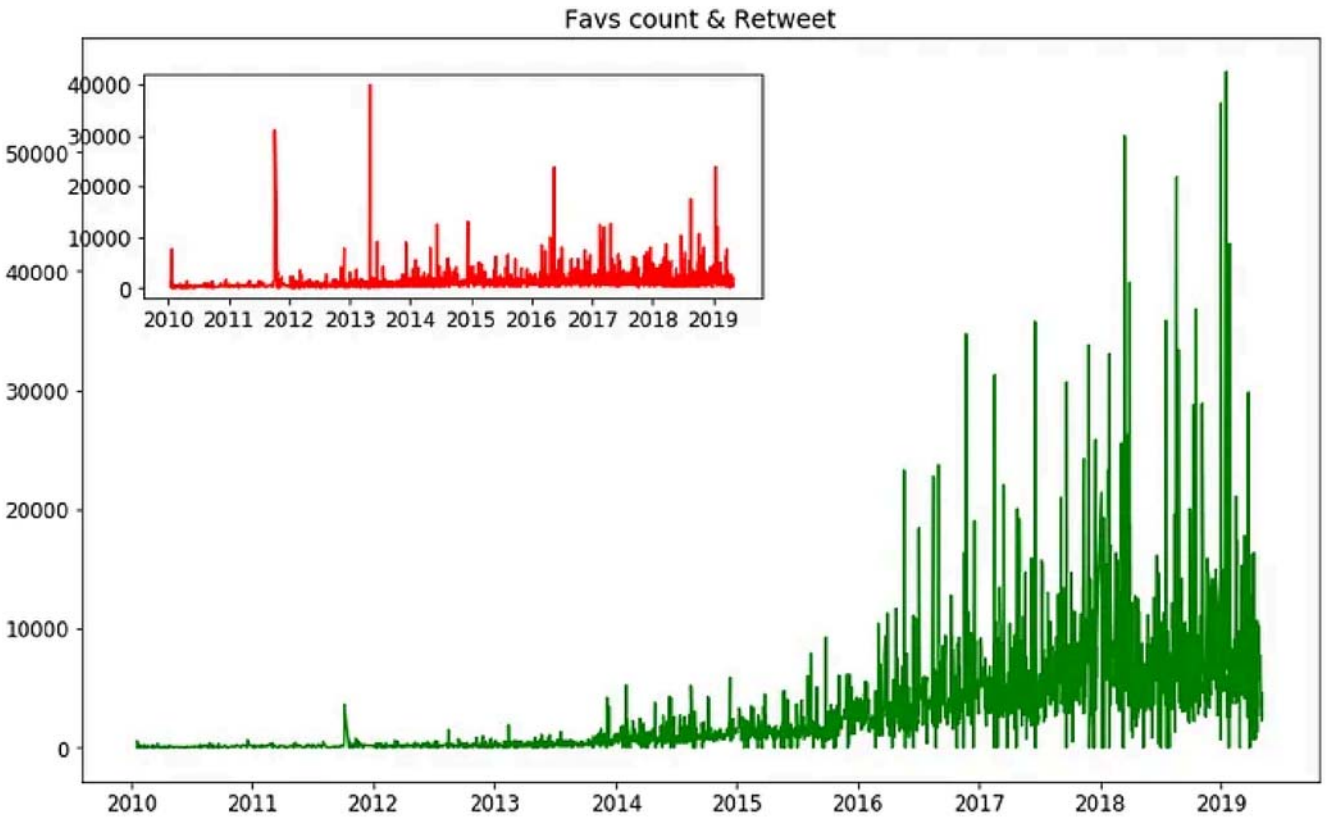
Most of the tweets are in the PM. The maximum is around **5.00 PM** with 297. Even he tweeted at midnight or 1.00 AM 😀.AM tweets are very less.



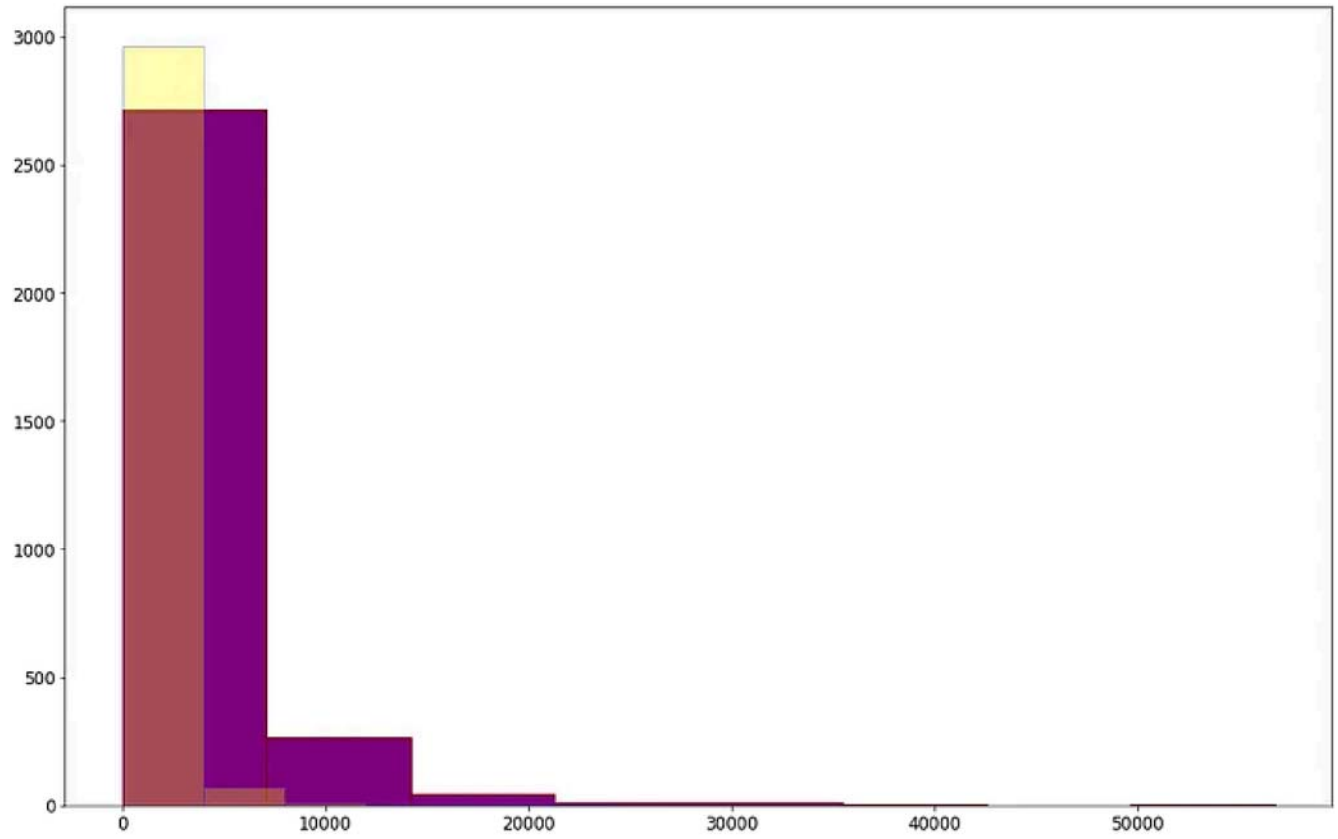

```
17    297
14    250
21    230
16    219
20    212
18    202
15    199
22    193
13    188
23    163
19    154
0     147
12    137
1      90
6      54
4      53
2      53
3      52
5      48
8      39
7      30
11     27
```

```
9       11
10       9
```

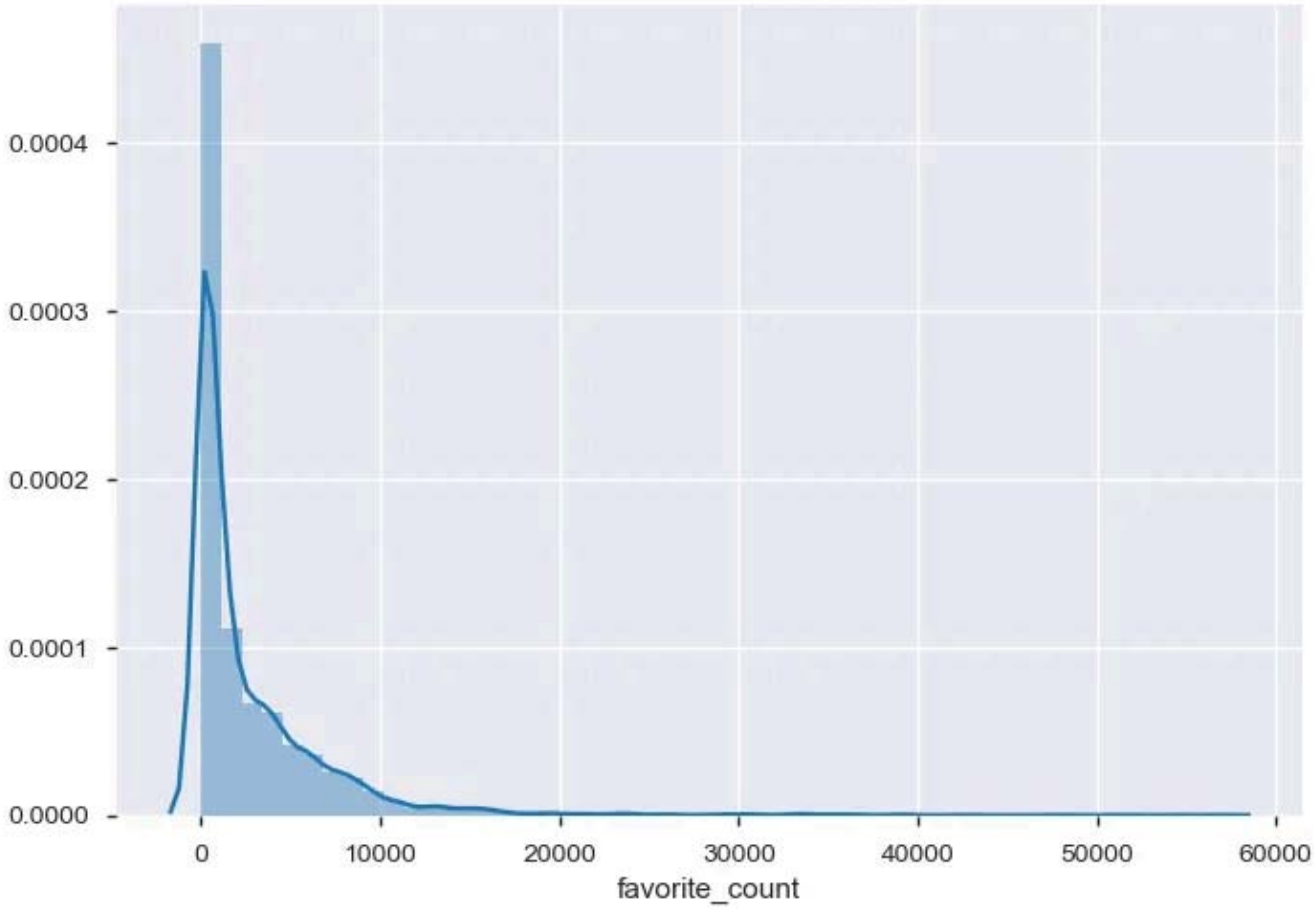
Fav count & Retweets over the years

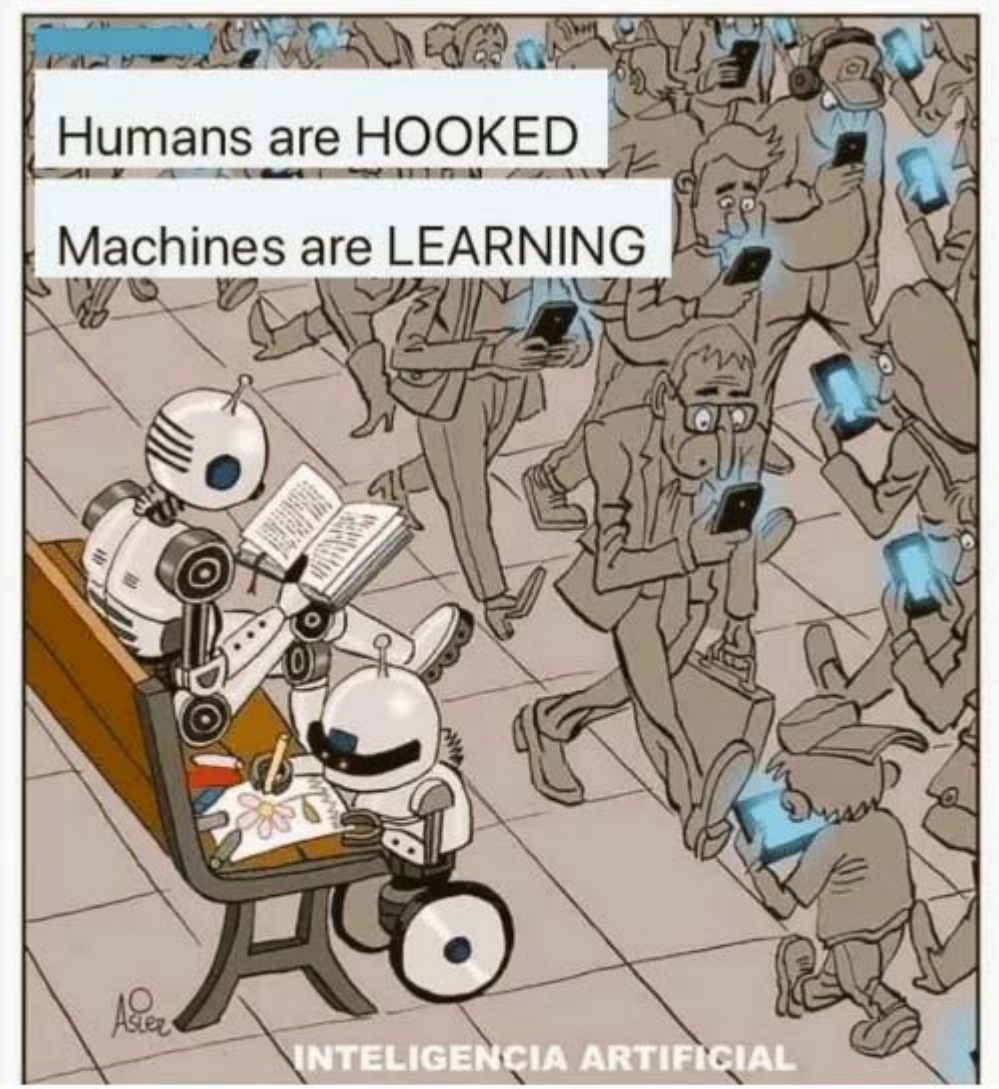


Fav and Retweet Histogram



Fav count — Seaborn





Machines are LEARNING, Humans are HOOKED. Received as a WhatsApp forward from a friend.

Sentiment Analysis

Image Source: Medium Article

What is the Sentiment Analysis?

Sentiment Analysis also is known as *Opinion Mining* is a field within Natural Language Processing (NLP) that builds systems that try to identify and extract opinions within the text. Usually, besides identifying the opinion, these systems extract attributes of the expression e.g.:

- *Polarity*: if the speaker expresses a *positive* or *negative* opinion,
- *Subject*: the thing that is being talked about,
- *Opinion holder*: the person, or entity that expresses the opinion.

With the help of sentiment analysis systems, this unstructured information could be automatically transformed into structured data of public opinions about products, services, brands, politics, or any topic that people can express opinions about. This data can be very useful for commercial applications like marketing analysis, public relations, product reviews, net promoter scoring, product feedback, and customer service.

Some of the Python Sentiment Analysis API's & Libraries

- Scikit-learn
- NLTK

- **SpaCy**
- **TensorFlow**
- **Keras**
- **PyTorch**
- **Gensim**
- **Polglot**
- **TextBlob**
- **Pattern**
- **Vocabulary**
- **Stanford CoreNLP Python**
- **Montylingua**

Sentiment Analysis by TextBlob:

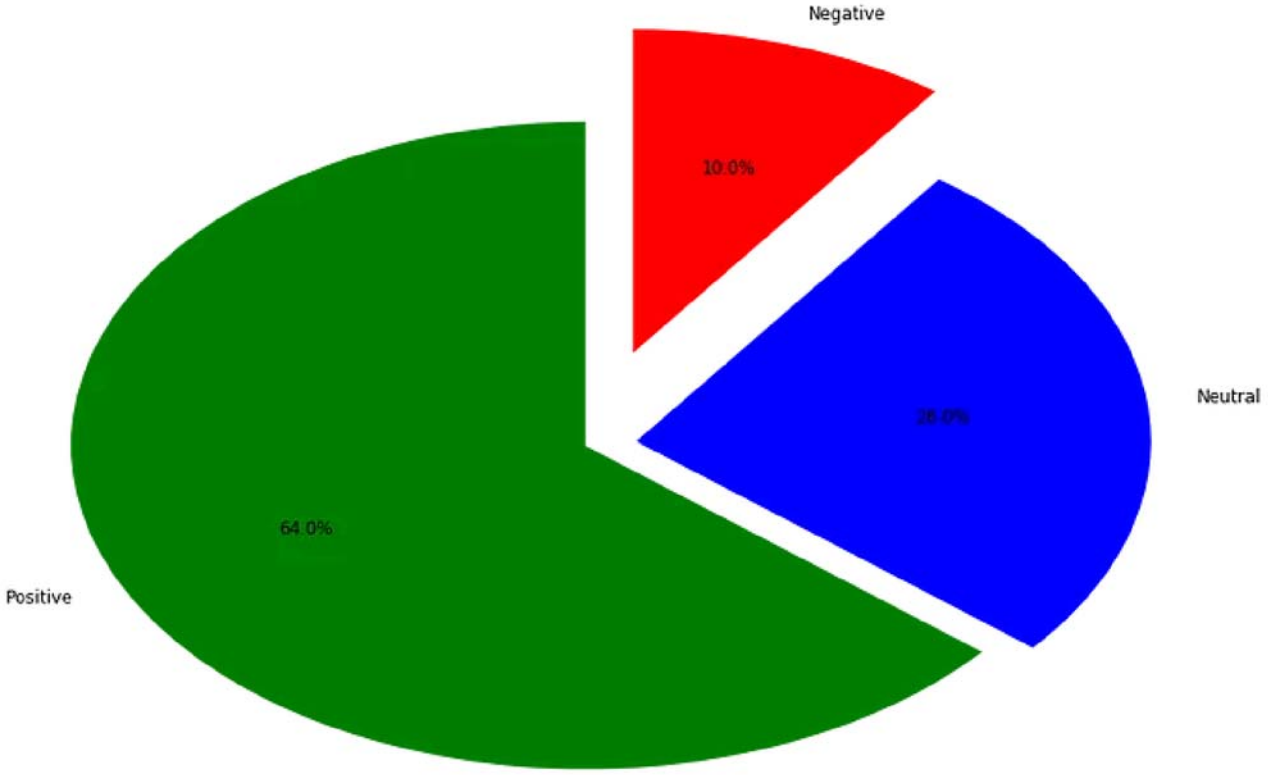
TextBlob is a Python (2 and 3) library for processing textual data. It provides a simple API for diving into common natural language processing (NLP) tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and more.

The code is

```
Percentage of positive tweets: 63.23192672554792%
Percentage of neutral tweets: 26.66012430487406%
Percentage of negative tweets: 10.107948969578018%
```

The output is above

The pie chart is



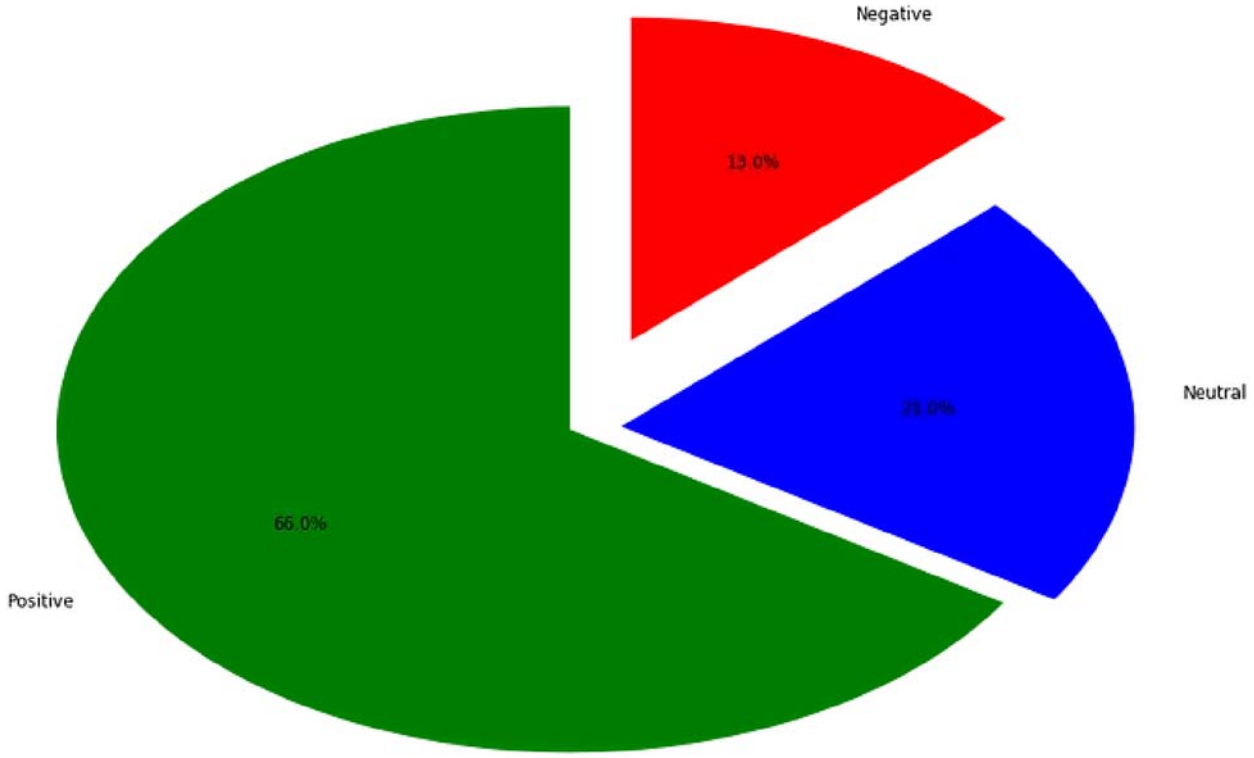
Sentiment Analysis by Vader:

VADER (Valence Aware Dictionary and sEntiment Reasoner) is a lexicon and rule-based sentiment analysis tool that is *specifically attuned to sentiments expressed in social media*. It is fully open-sourced under the [MIT License]

For more information please check this GitHub link

The code is similar to the above in calculating the sentiment score. Only the library is different. We are considering only the compound score.

```
Percentage of positive tweets: 66.47039581288846%
Percentage of neutral tweets: 20.870134118416747%
Percentage of negative tweets: 12.659470068694798%
```



	tweet	SA	VSA
0	Melinda and I love meeting with people who are trying to change the world. Here are four of the people who inspired...	1	1
1	What does a "dirt detective" do for a living? I'm eager to watch this talk from soil scientist Asmeret Asefaw Berhe...	0	1
2	A few months ago, Fred contributed a piece to my blog which outlined why educating the next generation of African l...	-1	0
3	By the end of the century, almost half of the young people in the world will be in sub-Saharan Africa.	-1	0
4	. has an ambitious plan to unlock the talents of young Africans. I was thrilled to see him featured in...	1	1
5	It's hard for me to overstate how brave people who fight outbreaks are. They face a near-impossible task with grace...	1	1
6	Congratulations on your 5th anniversary . Your leadership on some of the world's toug...	0	1
7	The world's historic progress to #endpolio would not be possible without the strong commitment of partners like...	1	-1
8	If you enjoyed the this weekend, I think you'll be inspired by these real-life heroes:...	1	1
9	It's amazing to think about the creative ways we've delivered vaccines over the years (even by dog sled, when neces...	1	1
10	The promise and perils of gene-editing might be the most important public debate we haven't been having widely enou...	1	1
11	One of my favorite authors, , co-wrote a piece about why nuclear power is so important in fighting climate...	1	1
12	Deaths from malaria have dropped 42% since 2000. I'm as optimistic as ever that we can get to zero deaths in our ge...	0	1
13	Stories like this one from Kenya remind me of why Melinda and I started our foundation. I'm optimistic we will cont...	0	1
14	Melinda's new book, The #MomentofLift, is out today: \r\n\r\nShe combines her mastery of data wit...	1	0
15	On #EarthDay, I'm inspired by the inventors who are tackling climate change and all the people who are supporting t...	1	1
16	Melinda's new book The #MomentofLift is an urgent call to action. I'll commit to the lift by working to ensure that...	1	1
17	Gene drive is a technique where you give a few mosquitoes an edited gene and it drives itself into all their offspr...	-1	0
18	Less than a century ago, families everywhere lived in fear of a mosquito bite. Now 75% of all malaria cases occur i...	-1	-1
19	3: Which of the following factors affect the spread of malaria? The answer is in my latest blog post:	1	0
20	2: What kind of pathogen causes malaria? Find the answer here:	1	0

SA — Sentiment Analysis Score from Textblob

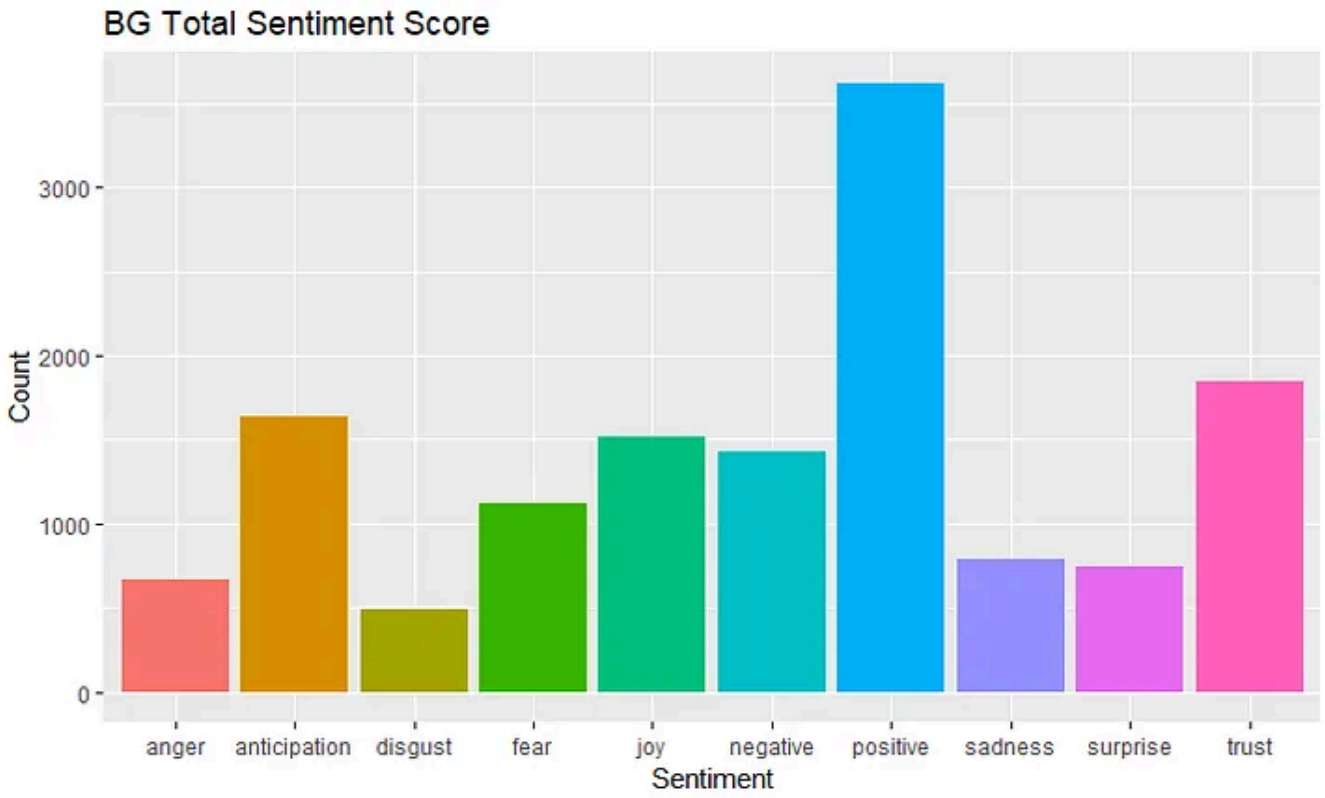
VSA — Sentiment Analysis Score from Vader

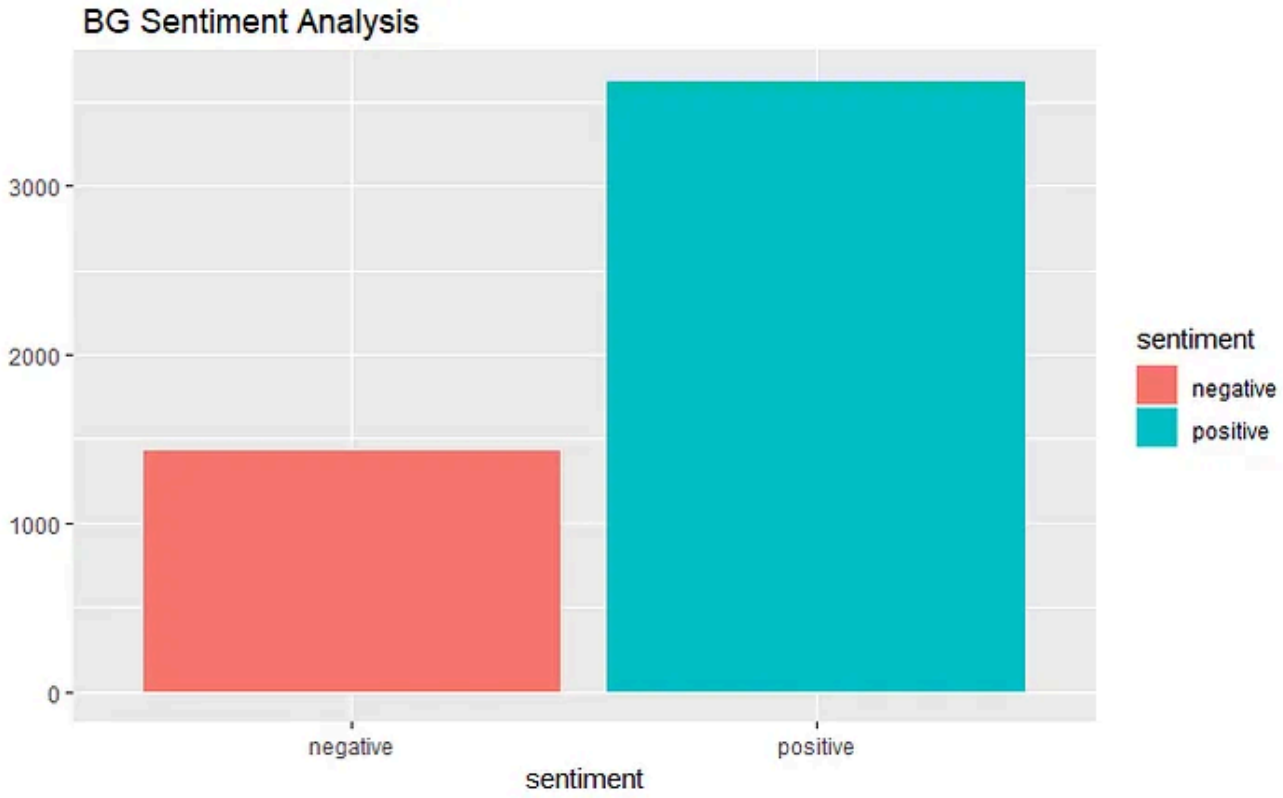
Sentiment Analysis by Syuzhet

This package comes with four sentiment dictionaries and provides a method for accessing the robust, but computationally expensive, sentiment extraction tool developed in the NLP group at Stanford.

For more information please check the link.

This package is available in R and the code is below. I referred to the sample code and did it. If you want more details then please check the link I provided and also check Rcran projects.


BG Total Sentiment Score
Count
3000
2000
1000
0
anger
anticipation
disgust
fear
joy
negative
positive
sadness
surprise
trust
Sentiment

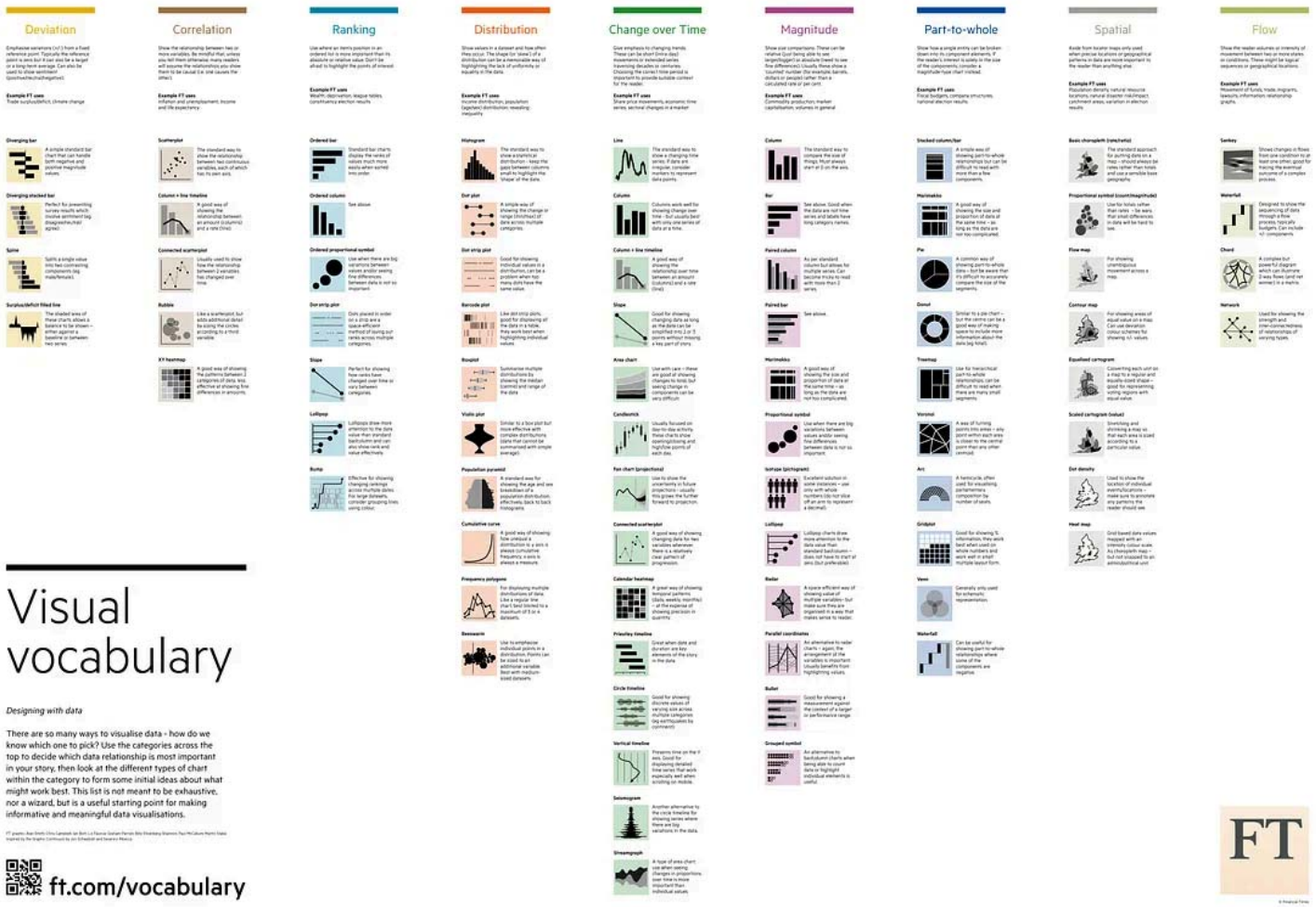

BG Sentiment Analysis
3000
2000
1000
0
negative
positive
sentiment
sentiment
negative
positive

	anger	anticipation	disgust	fear	joy	sadness	surprise	trust	negative	positive
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	1	2	0	1	1	0	2
3	0	3	2	1	1	0	1	2	2	3
4	0	0	0	0	0	0	0	0	0	0
5	0	1	0	0	1	0	1	0	0	1
6	0	2	0	0	1	0	1	0	0	1

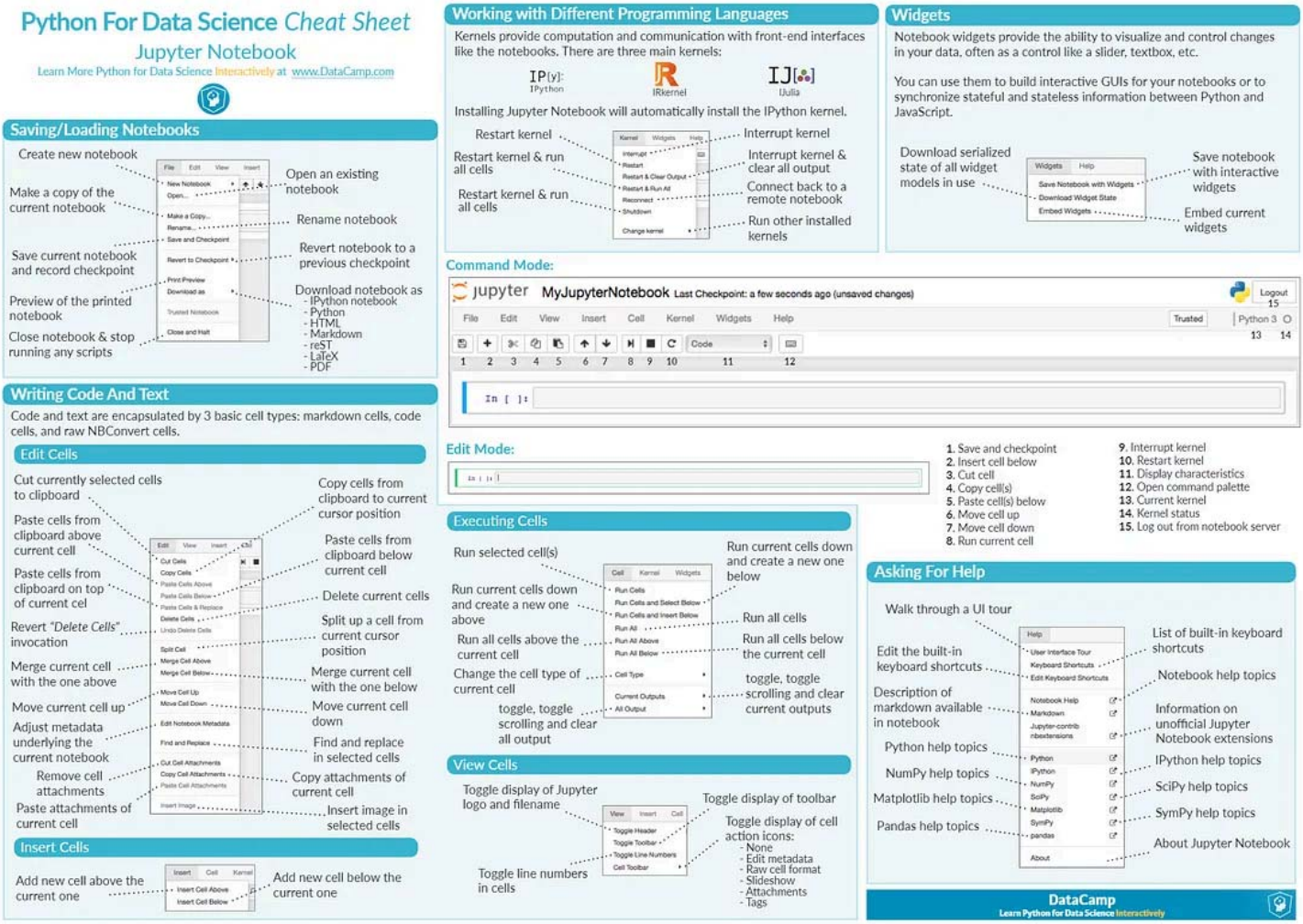
Conclusion

Hope you enjoyed reading this article. Now you know how easy to extract data by using Twitter API's -Streaming and Search. With only a few lines of code, you can get the data from Twitter and use the pandas for all the analyses. In the next post, I will write about scrapy and selenium. I will also update my GitHub sometime soon and provide the link. Meanwhile, if you want to provide feedback or suggestions or have any questions then please go ahead and send an email to epmanalytics100@gmail.com. Thanks again for reading my post 👍

I found this chart very helpful. (Source: FT)



Jupyter Cheat Sheet (Source: Datacamp)



Pandas Cheat Sheet 1(Source: Datacamp)

Python For Data Science *Cheat Sheet*

Pandas Basics

Learn Python for Data Science Interactively at www.DataCamp.com

Pandas

The **Pandas** library is built on NumPy and provides easy-to-use **data structures** and **data analysis** tools for the Python programming language.

Use the following import convention:

```
>>> import pandas as pd
```

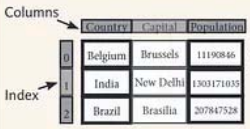
Pandas Data Structures

Series

A one-dimensional labeled array capable of holding any data type


```
>>> s = pd.Series([3, -5, 7, 4], index=['a', 'b', 'c', 'd'])
```

DataFrame



A two-dimensional labeled data structure with columns of potentially different types

```
>>> data = {'Country': ['Belgium', 'India', 'Brazil'],
            'Capital': ['Brussels', 'New Delhi', 'Brasilia'],
            'Population': [11190846, 1303171035, 207847528]}

>>> df = pd.DataFrame(data,
                      columns=['Country', 'Capital', 'Population'])
```

Asking For Help

```
>>> help(pd.Series.loc)
```

Selection

Also see NumPy Arrays

Getting

```
>>> s['b']
-5
```
Get one element

```
>>> df[1:]
   Country    Capital  Population
1   India  New Delhi  1303171035
2  Brazil   Brasilia   207847528
```
Get subset of a DataFrame

Selecting, Boolean Indexing & Setting

By Position

```
>>> df.iloc[[0],[0]]
'Belgium'
>>> df.iat([0],[0])
'Belgium'
```
Select single value by row & column

By Label

```
>>> df.loc[[0], ['Country']]
'Belgium'
>>> df.at([0], ['Country'])
'Belgium'
```
Select single value by row & column labels

By Label/Position

```
>>> df.ix[2]
Country      Brazil
Capital    Brasilia
Population  207847528
```
Select single row of subset of rows

```
>>> df.ix[:,'Capital']
0     Brussels
1    New Delhi
2     Brasilia
```
Select a single column of subset of columns

```
>>> df.ix[1,'Capital']
'New Delhi'
```
Select rows and columns

Boolean Indexing

```
>>> s[~(s > 1)]
>>> s[(s < -1) | (s > 2)]
>>> df[df['Population']>1200000000]
```
Series `s` where value is not >1
`s` where value is <-1 or >2
Use filter to adjust DataFrame

Setting

```
>>> s['a'] = 6
```
Set index `a` of Series `s` to 6

Dropping

```
>>> s.drop(['a', 'c'])
>>> df.drop('Country', axis=1)
```
Drop values from rows (axis=0)
Drop values from columns(axis=1)

Sort & Rank

```
>>> df.sort_index()
>>> df.sort_values(by='Country')
>>> df.rank()
```
Sort by labels along an axis
Sort by the values along an axis
Assign ranks to entries

Retrieving Series/DataFrame Information

Basic Information

```
>>> df.shape
>>> df.index
>>> df.columns
>>> df.info()
>>> df.count()
```
(rows,columns)
Describe index
Describe DataFrame columns
Info on DataFrame
Number of non-NA values

Summary

```
>>> df.sum()
>>> df.cumsum()
>>> df.min()/df.max()
>>> df.idxmin()/df.idxmax()
>>> df.describe()
>>> df.mean()
>>> df.median()
```
Sum of values
Cummulative sum of values
Minimum/maximum values
Minimum/Maximum index value
Summary statistics
Mean of values
Median of values

Applying Functions

```
>>> f = lambda x: x*2
>>> df.apply(f)
>>> df.applymap(f)
```
Apply function
Apply function element-wise

Data Alignment

Internal Data Alignment

NA values are introduced in the indices that don't overlap:

```
>>> s3 = pd.Series([7, -2, 3], index=['a', 'c', 'd'])
>>> s + s3
a    10.0
b     NaN
c     5.0
d     7.0
```

Arithmetic Operations with Fill Methods

You can also do the internal data alignment yourself with the help of the fill methods:

```
>>> s.add(s3, fill_value=0)
a    10.0
b    -5.0
c     5.0
d     7.0
>>> s.sub(s3, fill_value=2)
>>> s.div(s3, fill_value=4)
>>> s.mul(s3, fill_value=3)
```

I/O

Read and Write to CSV

```
>>> pd.read_csv('file.csv', header=None, nrows=5)
>>> df.to_csv('myDataFrame.csv')
```

Read and Write to Excel

```
>>> pd.read_excel('file.xlsx')
>>> pd.to_excel('dir/myDataFrame.xlsx', sheet_name='Sheet1')
```

Read multiple sheets from the same file

```
>>> xlsx = pd.ExcelFile('file.xls')
>>> df = pd.read_excel(xlsx, 'Sheet1')
```

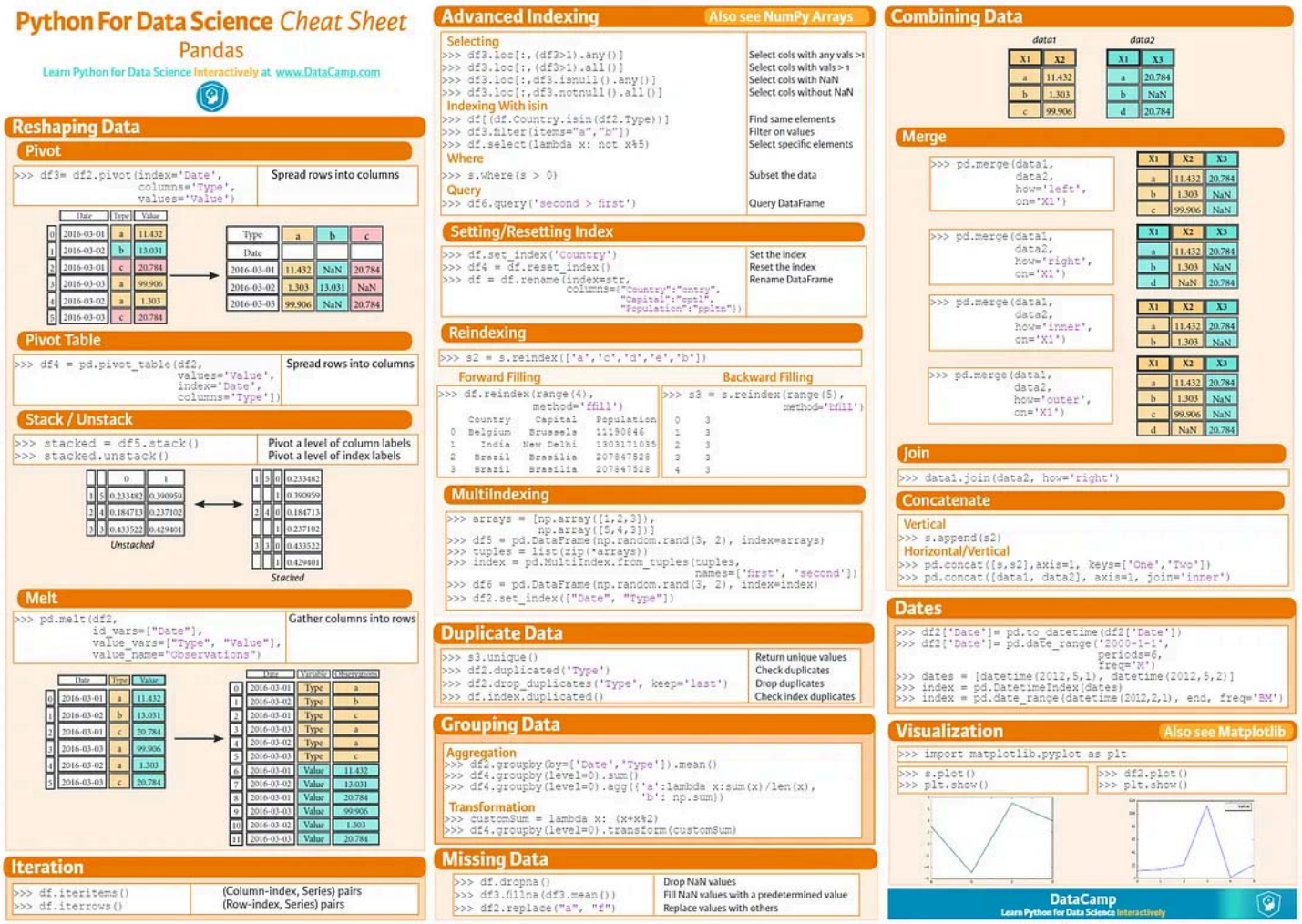
Read and Write to SQL Query or Database Table

```
>>> from sqlalchemy import create_engine
>>> engine = create_engine('sqlite:///:memory:')
>>> pd.read_sql("SELECT * FROM my_table;", engine)
>>> pd.read_sql_table('my_table', engine)
>>> pd.read_sql_query("SELECT * FROM my_table;", engine)
```

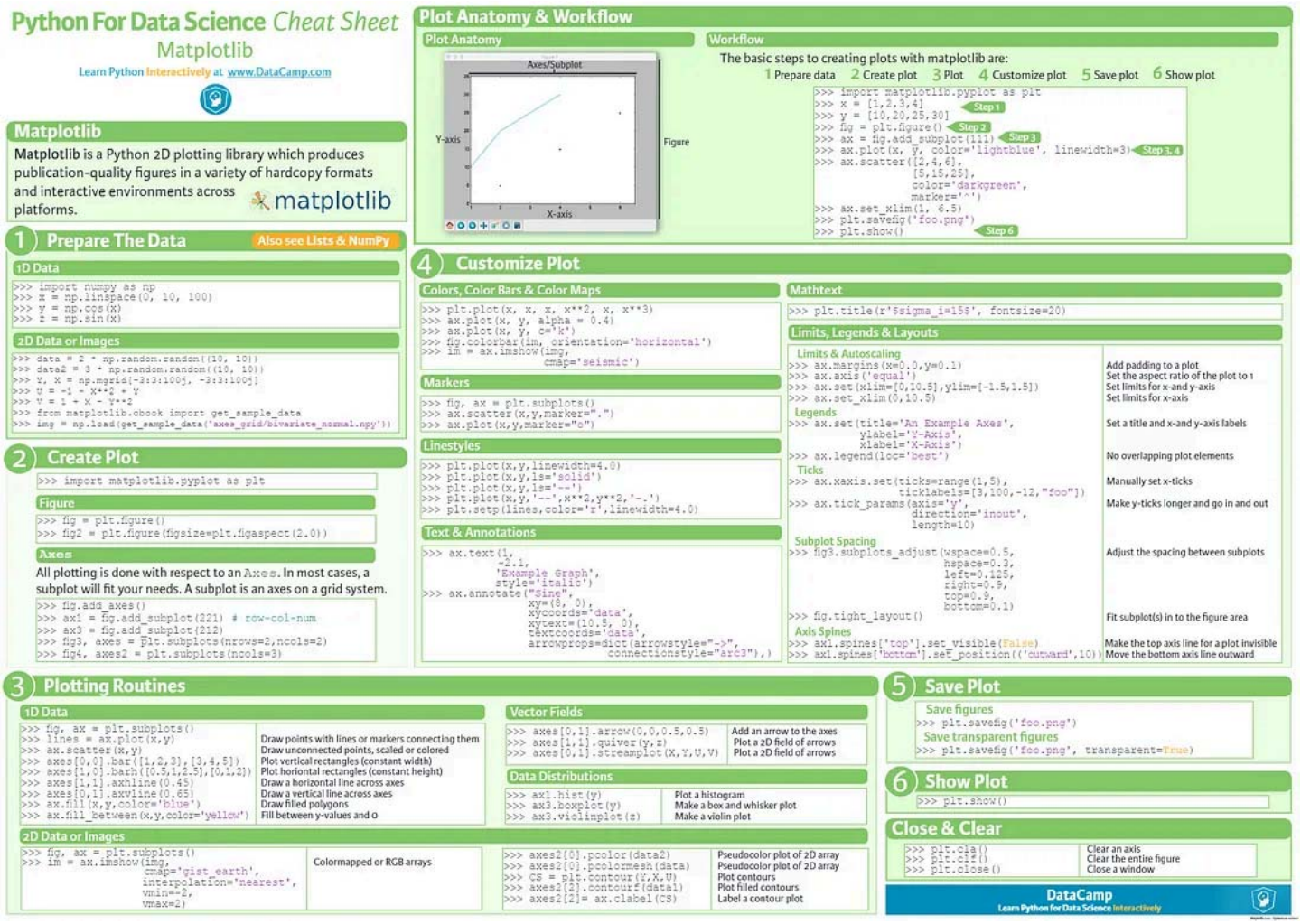
`read_sql()` is a convenience wrapper around `read_sql_table()` and `read_sql_query()`

```
>>> pd.to_sql('myDf', engine)
```

DataCamp
Learn Python for Data Science Interactively



Matplotlib Cheat Sheet(Source: Datacamp)



Thanks again for reading my post 😀

NLP Python Data Visualization Data Science Twitter

Following

Written by Senthil E

2.7K Followers · Writer for Towards Data Science

ML/DS - Certified GCP Professional Machine Learning Engineer, Certified AWS Professional Machine learning Speciality,Certified GCP Professional Data Engineer .

More from Senthil E and Towards Data Science

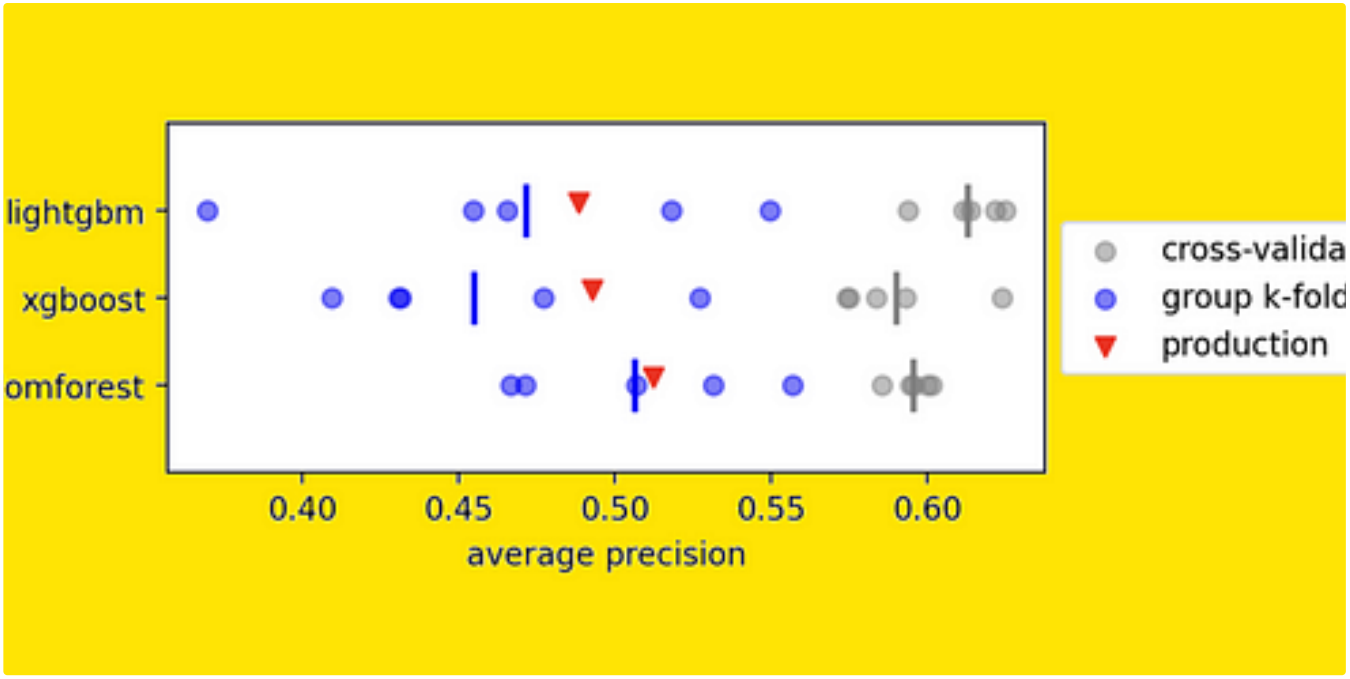
Senthil E in Level Up Coding

Navigating the World of LLMs: A Beginner's Guide to Prompt Engineering-Part 1

From Basics To Advanced Techniques

26 min read · Mar 17, 2024

213 1



Samuele Mazzanti in Towards Data Science

Why You Should Never Use Cross-Validation

In real-world applications, using randomized cross-validation is always a bad choice. Here is why.

· 12 min read · Mar 27, 2024

1.1K 29

Dario Radečić in Towards Data Science

Pandas vs. Polars—Time to Switch?

Looking to speed up your data processing pipelines up to 10 times? Maybe it's time to say goodbye to Pandas.

✦ · 7 min read · Apr 7, 2024

1K 17

Senthil E in Level Up Coding

Navigating the World of LLMs: A Beginner's Guide to Prompt Engineering-Part 2

From Basics To Advanced Techniques

32 min read · Mar 17, 2024

302

See all from Senthil E

See all from Towards Data Science

Recommended from Medium

Bale Chen in Towards Data Science

Emojis Aid Social Media Sentiment Analysis: Stop Cleaning Them Out!

Leverage emojis in social media sentiment analysis to improve accuracy.

14 min read · Jan 31, 2023

332 2

Eulene

Sentiment Analysis of Amazon Reviews Using Natural Language Processing

11 min read · Feb 8, 2024

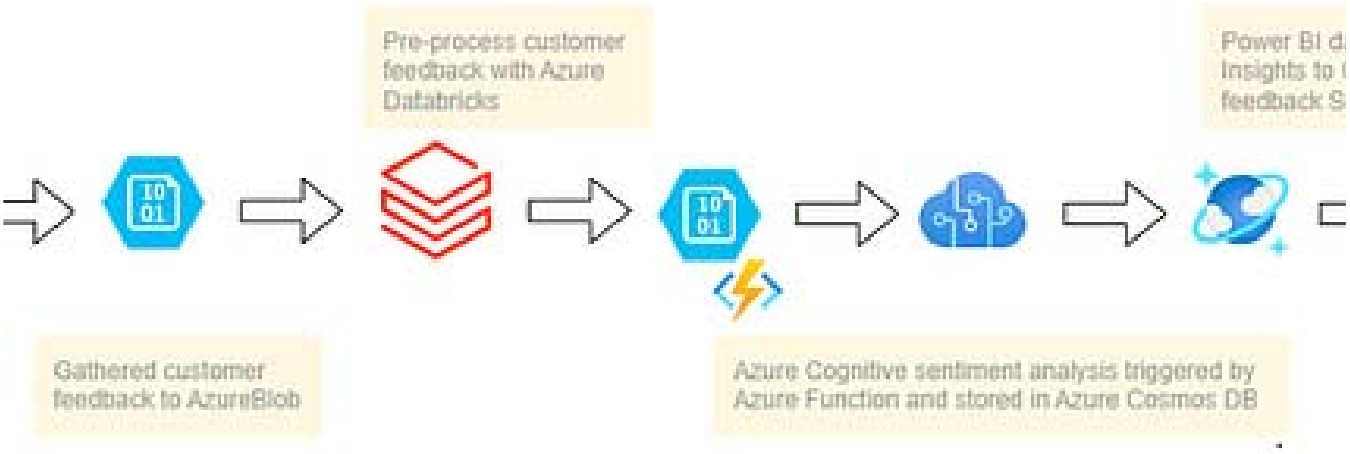
Nuno Bispo in Django Unleashed

How to Build an AI Chatbot for Q&A on Any Website

This article delves into the development of a chatbot, designed to read a website's content with web scraping and conversational AI...

✦ · 11 min read · Apr 5, 2024

11



Merwan Chinta in CodeNx

🎯 Automated Customer Review Sentiment Analysis for Businesses

🌟Objective

12 min read · Nov 13, 2023

100

Ubaid Shah

Twitter Sentiment Analysis — A Step by Guide

In today's digital age, social media platforms like Twitter have become a goldmine for valuable data. Businesses, researchers, and…

6 min read · Nov 8, 2023

30

Onat Kaya

How to Scrape Tweets From Twitter For Free (2023)

A Free and Easy Way To Scrape Tweets

7 min read · Nov 27, 2023

39