

# Measuring User's Influence in Twitter

V Sai Pratyusha

Akula Bhavani

V Jagath Chandra

M Madhan

**Abstract**— Directed links in social media may represent something from intimate friendships to common interests, or even a passion for breaking news or celebrity gossip. Such directed links confirm the flow of data and therefore indicate a user's influence on others — an inspiration that is crucial in social science and infective agent selling. Throughout this paper, using a good deal of data collected from Twitter, we tend to gift an in-depth comparison of 3 measures of influence: indegree, retweets, and mentions. Supported these measures, we tend to investigate the dynamics of user influence across topics and time. We tend to create many fascinating observations. First, in style users World Health Organization have high indegree are not essentially prestigious in terms of spawning retweets or mentions. Second, most prestigious users will hold vital influence over a spread of topics. We tend to believe that these findings give new insights for infective agent selling and counsel that topological measures like indegree alone reveals very little or no concerning the influence of a user.

These measures are terribly numerous. Some are supported easy metrics provided by the Twitter API, whereas others are supported complicated mathematical models.

**Keywords**— Twitter, User's influence, Social media, Indegree, Retweets, Mentions, Twitter API.

## 1. Introduction

Since the origins of social network analysis, there has been interest in distinguishing the foremost relevant actors of a social network. With the increase of the net and technology, on-line social networks (OSNs) turned into difficult cases of study, on that massive information content and sophisticated social ties among actors converge. Knowing the influence of users and having the ability to predict it are often helpful for several applications, like infective agent promoting, data propagation, looking out, expertise recommendation, social client relationship management, percolation theory, etc. A social network will be depicted by a graph, as in Figure one, whose nodes will represent the users, and the edges the social ties among them. The position of a node refers to its relative importance within the network to that it belongs. there's a huge literature regarding position measures to spot the most necessary actors on a social network. Every live relies on completely different relevancy criteria. for instance, two of the foremost ancient measures square measure degree and closeness. The primary one considers for every node the amount of its adjacent edges, whereas the other considers the minimum sum of the shortest ways from a node to any or all the opposite nodes inside the network.

	user	tweet
user	follows / is followed by mention, replies to, retweets to	Posts, retweets, likes, replies
tweet	posted by, retweeted by, liked by, replied by	replies / is replied from retweets / is retweeted from

Figure 1: Twitter relationships between users and tweets.

## 2. Software Requirements & Specifications

Google colab (Online IDE)

### Libraries:

#### BeautifulSoup



Beautiful Soup is a Python library for pulling data out of HTML and XML files. It works with your favorite parser to provide idiomatic ways of navigating, searching, and modifying the parse tree. It commonly saves programmers hours or days of work.

These instructions illustrate all major features of BeautifulSoup 4, with examples. I show you what the library is good for, how it works, how to use it, how to make it do what you want, and what to do when it violates your expectations.

This document covers BeautifulSoup version 4.9.3. The examples in this documentation should work the same way in Python 2.7 and Python 3.8.

You might be looking for the documentation for BeautifulSoup 3. If so, you should know that BeautifulSoup 3 is no longer being developed and that support for it will be dropped on or after December 31, 2020. If you want to learn about the differences between BeautifulSoup 3 and BeautifulSoup 4, see [Porting code to BS4](#).

```
from bs4 import BeautifulSoup
```

#### Pandas



Pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series. It is free software released under the three-clause BSD license. The name is derived from the term "panel data", an econometrics term for data sets that include observations over multiple time periods for the same individuals. Its name is a play on the phrase "Python data analysis" itself. Wes McKinney started building what would become pandas at AQR Capital while he was a researcher there from 2007 to 2010.

## RegEx



This module provides regular expression matching operations similar to those found in Perl.

Both patterns and strings to be searched can be Unicode strings (str) as well as 8-bit strings (bytes). However, Unicode strings and 8-bit strings cannot be mixed: that is, you cannot match a Unicode string with a byte pattern or vice-versa; similarly, when asking for a substitution, the replacement string must be of the same type as both the pattern and the search string.

## Tweepy



An easy-to-use Python library for accessing the Twitter API.

## OAuthHandler

Tweepy supports both OAuth 1a (application-user) and OAuth 2 (application-only) authentication. Authentication is handled by the `tweepy.AuthHandler` class.

Tweepy tries to make OAuth 1a as painless as possible for you. To begin the process we need to register our client application with Twitter. Create a new application and once you are done you should have your consumer key and secret. Keep these two handy, you'll need them.

The next step is creating an `OAuthHandler` instance. Into this we pass our consumer key and secret which was given to us in the previous paragraph:

```
auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
```

## Preprocessor

Preprocessor is a preprocessing library for tweet data written in Python. It was written as part of my bachelor thesis in sentiment analysis. Later I extracted it to a library for broader usage.

When building Machine Learning systems based on tweet data, a preprocessing is required. This library makes it easy to clean, parse or tokenize the tweets.

## Seaborn



Seaborn is an open-source Python library built on top of matplotlib. It is used for data visualization and exploratory data analysis. Seaborn works easily with dataframes and the Pandas library. The graphs created can also be customized easily. Below are a few benefits of Data Visualization.

Graphs can help us find data trends that are useful in any machine learning or forecasting project.

## Fire

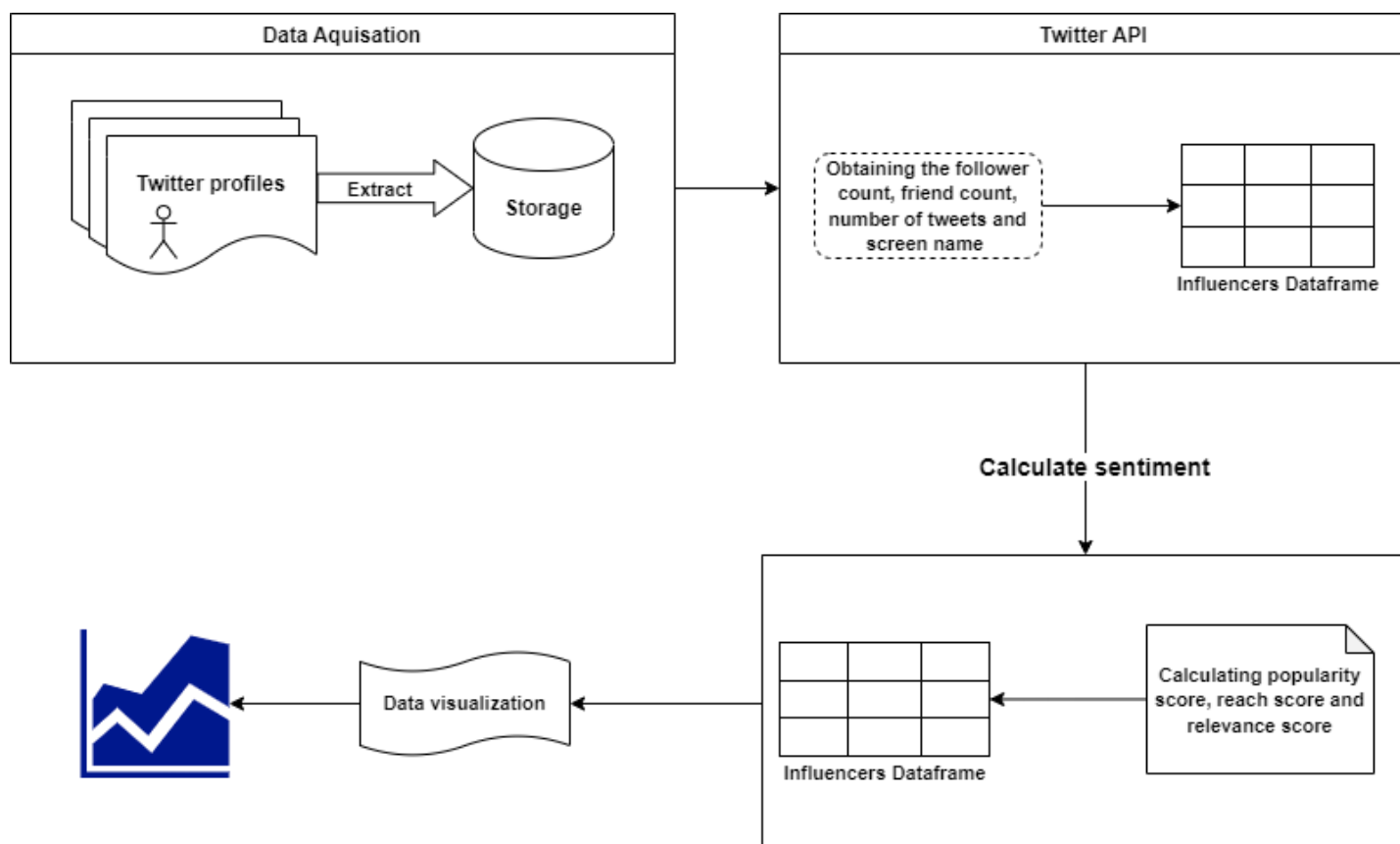
Python Fire is a library for automatically generating command line interfaces (CLIs) from absolutely any Python object.

## NLTK

NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries.

Natural Language Processing with Python provides a practical introduction to programming for language processing. Written by the creators of NLTK, it guides the reader through the fundamentals of writing Python programs, working with corpora, categorizing text, analyzing linguistic structure, and more. The online version of the book has been updated for Python 3 and NLTK 3.

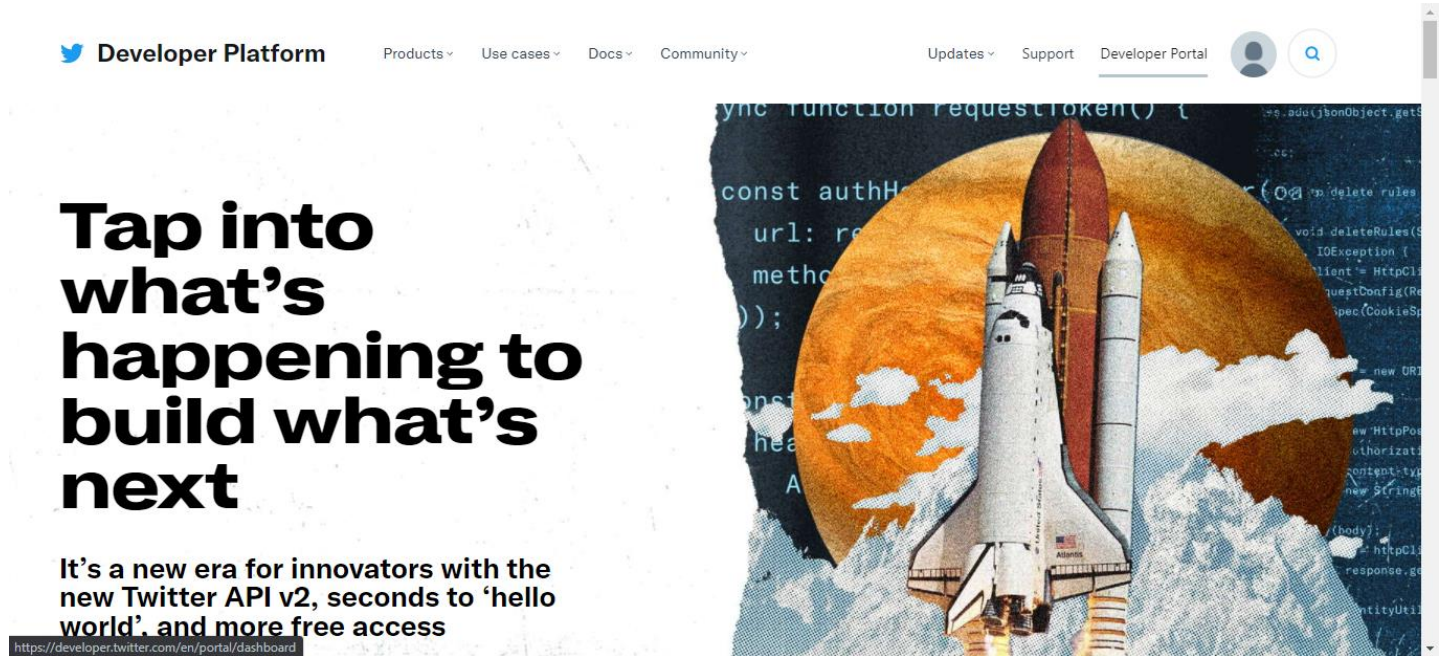
## 3. System Design



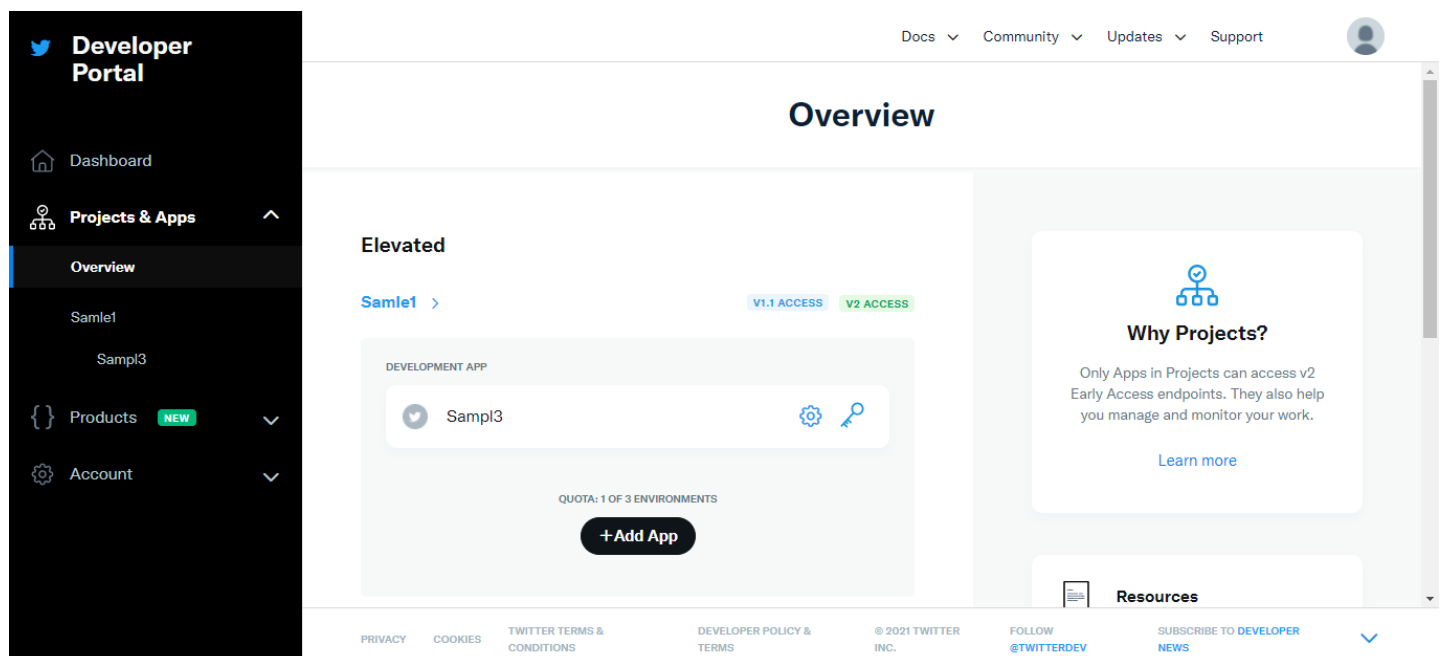
## 4. Implementation

### Obtaining API key

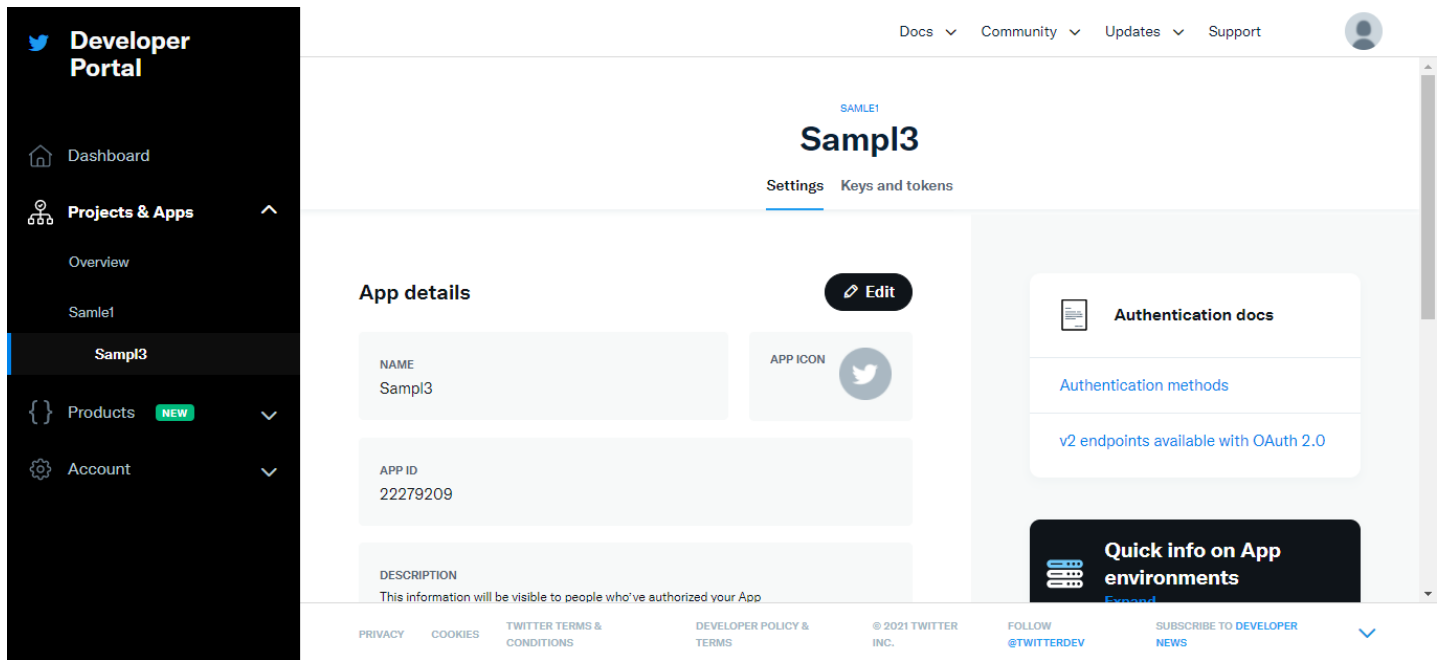
- Type <https://developer.twitter.com/> in browser.
- Navigate to **Developer Portal**.



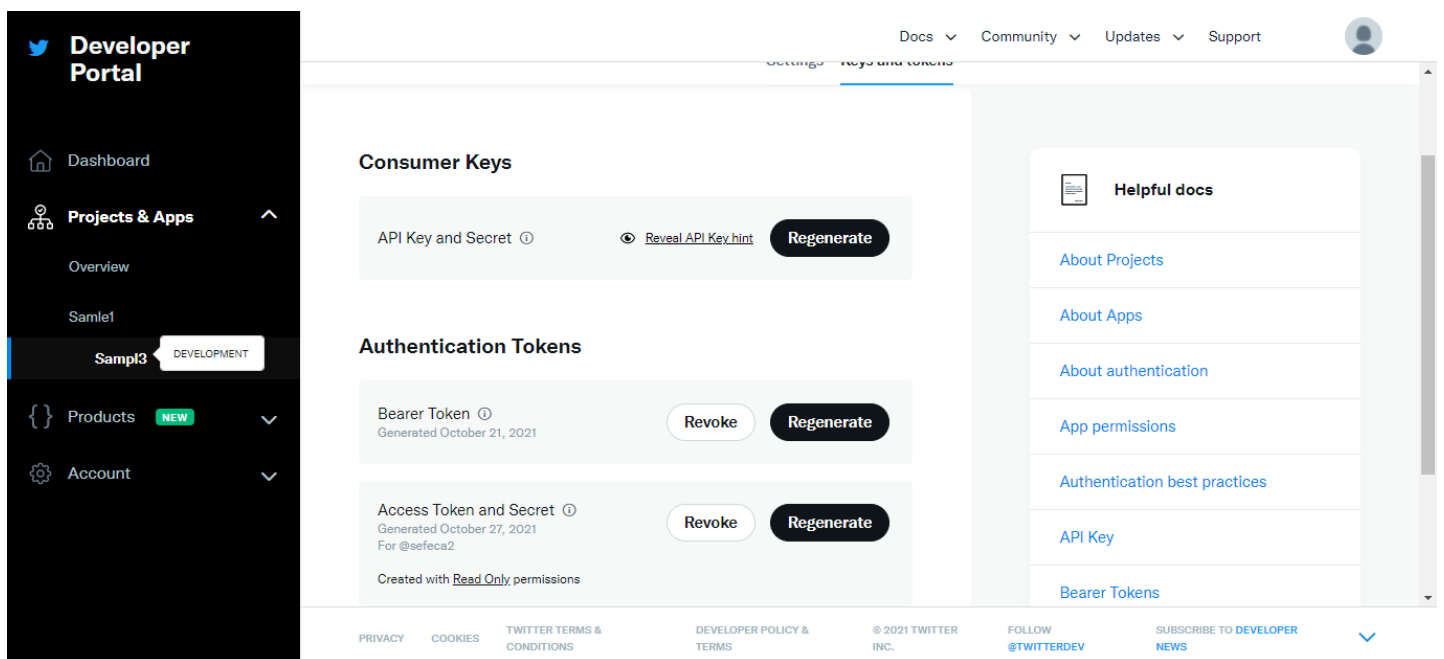
- Click on **Projects & Apps > Overview > Add App**



- After creating the app & filling up details the page looks like this.



- Click on **Keys and Tokens**
- Click on **Generate** for **API Key and Secret**, **Bearer Token**, **Access Token and Secret**.



- Now we can implement these keys in the code to access **tweets**.

## 5. Testing

### ▼ SMA Project

### ▼ Web Scrapping

✓ [1] 1 pip install preprocessing

```
Collecting preprocessing
  Downloading preprocessing-1.1.3.tar.gz (4.2 kB)
Building wheels for collected packages: preprocessing
  Building wheel for preprocessing (setup.py) ... done
  Created wheel for preprocessing: filename=preprocessing-1.1.3-py3-none-any.whl size=4477 sha256=d0e5543d6b65baa94b507403e74acd99d7144b125d2b728b047d02de7e0ab131
  Stored in directory: /root/.cache/pip/wheels/0e/b7/36/aa37256db62b4bfd35a6f1b5536e9ba843f257b79dcfb3d5f1
Successfully built preprocessing
Installing collected packages: preprocessing
Successfully installed preprocessing-1.1.3
```

✓ [2] 1 pip install fire

```
Collecting fire
  Downloading fire-0.4.0.tar.gz (87 kB)
    |████████████████████████████████████████| 87 kB 3.3 MB/s
Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages (from fire) (1.15.0)
Requirement already satisfied: termcolor in /usr/local/lib/python3.7/dist-packages (from fire) (1.1.0)
Building wheels for collected packages: fire
  Building wheel for fire (setup.py) ... done
  Created wheel for fire: filename=fire-0.4.0-py3-none-any.whl size=115943 sha256=e57953a5b35d021cc33ed815a6b4ad3150e26027cafabe21e6ed6dc25a8809
  Stored in directory: /root/.cache/pip/wheels/8a/67/fb/2e8a12fa16661b9d5af1f654bd199366799740a85c64981226
Successfully built fire
Installing collected packages: fire
Successfully installed fire-0.4.0
```

✓ [3] 1 pip install tweet-preprocessor

```
Collecting tweet-preprocessor
  Downloading tweet_preprocessor-0.6.0-py3-none-any.whl (27 kB)
Installing collected packages: tweet-preprocessor
Successfully installed tweet-preprocessor-0.6.0
```

✓ [4] 1 import nltk

✓ [ ] 1 nltk.download('stopwords')

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data] Unzipping corpora/stopwords.zip.
True
```

```
[nltk_data] Downloading package gazetteers to /root/nltk_data...
[nltk_data] Unzipping corpora/gazetteers.zip.
[nltk_data] Downloading package genesis to /root/nltk_data...
[nltk_data] Unzipping corpora/genesis.zip.
[nltk_data] Downloading package gutenberg to /root/nltk_data...
[nltk_data] Unzipping corpora/gutenberg.zip.
[nltk_data] Downloading package inaugural to /root/nltk_data...
[nltk_data] Unzipping corpora/inaugural.zip.
[nltk_data] Downloading package movie_reviews to
[nltk_data] /root/nltk_data...
[nltk_data] Unzipping corpora/movie_reviews.zip.
[nltk_data] Downloading package names to /root/nltk_data...
[nltk_data] Unzipping corpora/names.zip.
[nltk_data] Downloading package shakespeare to /root/nltk_data...
[nltk_data] Unzipping corpora/shakespeare.zip.
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data] Unzipping corpora/stopwords.zip.
[nltk_data] Downloading package treebank to /root/nltk_data...
[nltk_data] Unzipping corpora/treebank.zip.
[nltk_data] Downloading package twitter_samples to
[nltk_data] /root/nltk_data...
[nltk_data] Unzipping corpora/twitter_samples.zip.
```



✓  
0s

```
[6] 1 #importing the necessary libraries needed for web scraping
    2 from requests import get
    3 from requests.exceptions import RequestException
    4 from contextlib import closing
    5 from bs4 import BeautifulSoup
    6 import pandas as pd
    7 import os, sys
    8 import re
    9 import tweepy
   10 from tweepy import OAuthHandler
   11 import preprocessor as p
   12 import seaborn as sns
   13 import fire
```

✓  
0s

```
[7] 1 def simple_get(url):
    2     """
    3     Attempts to get the content at `url` by making an HTTP GET request.
    4     If the content-type of response is some kind of HTML/XML, return the
    5     text content, otherwise return None.
    6     """
    7     try:
    8         with closing(get(url, stream=True)) as resp:
    9             if is_good_response(resp):
   10                 return resp.content #.encode(BeautifulSoup.original_encoding)
   11             else:
   12                 return None
   13
   14     except RequestException as e:
   15         log_error('Error during requests to {0} : {1}'.format(url, str(e)))
   16         return None
   17
   18
   19 def is_good_response(resp):
   20     """
   21     Returns True if the response seems to be HTML, False otherwise.
   22     """
   23     content_type = resp.headers['Content-Type'].lower()
   24     return (resp.status_code == 200
   25             and content_type is not None
   26             and content_type.find('html') > -1)
   27
```



```

1 #python code to obtain 100 African Twitter Influencers
2 res = get_elements('https://www.businessinsider.co.za/the-biggest-twitter-profiles-south-africa-trevor-noah-elon-musk-bonang-2019-5', tag='h2')
3 res.reverse()
4
5 #converting the list into string
6 list_inf = str(res)
7
8 #deriving the top 100 african twitter influencers using regex
9 listinf = re.findall(r"@[\w]*", list_inf)
10 count = 1
11 print("\nTop African Twitter Influencers:
12 | | | | \n-----\n")
13 for aff in listinf:
14     print (f"{count}. {aff}")
15     count = count+1
16
17 #saving data as csv file
18 dat = pd.DataFrame(listinf, columns=['Influencers'])
19 dat.to_csv('top_african_influencers_.csv')

```

Top African Twitter Influencers:

-----

1. @ABdeVilliers17
2. @KP24
3. @bonang\_m
4. @MinnieDlamini
5. @News24
6. @DJZinhle
7. @Julius\_S\_Malema
8. @SuperSportTV
9. @casspernyovest
10. @PearlThusi

## ▼ Twitter API

```

[9] 1 #Variables that contains the user credentials to access Twitter API
2 consumer_key = 'qprNvAYFLJ0PI3TDgLN5jOXSh'
3 consumer_secret = 'oPNiksEiB1ljDospQlaYPZIdrV5EkvHYx6HFkHTRKGcRFogCuG'
4 access_token = '1449350837407465473-EPnn7XP2nOUNFstG2poP2tmA4SRatw'
5 access_token_secret = 'afj6I7No4UpiBjhg2blWou62BCTtJXNubQniZWgdLn70y'
6
7 #This handles Twitter authentication and the connection to Twitter Streaming API
8 auth = OAuthHandler(consumer_key, consumer_secret)
9 auth.set_access_token(access_token, access_token_secret)
10
11 api = tweepy.API(auth)
12 api = tweepy.API(auth, wait_on_rate_limit=True)

```

## Obtaining the follower count, friend count, number of tweets and screen name for african influencers from the Twitter API

```

✓ [10] 1 #getting the followers count, following, number of tweets, and screen name of the african influencers
1s 2 followers_count = {}
3 following_count = {}
4 num_of_tweets = {}
5 screen_name = {}
6
7 for count in range(len(listinf)):
8     try:
9         user = api.get_user(listinf[count])
10        followers_count[listinf[count]] = user.followers_count
11        following_count[listinf[count]] = user.friends_count
12        num_of_tweets[listinf[count]] = user.statuses_count
13        screen_name[listinf[count]] = user.screen_name
14    except Exception as e:
15        pass

✓ [11] 1 #adding the follower_count, following_count, number of tweets, and screen name as a new column in the influencers dataframe
0s 2 dat['Follower_count'] = dat['Influencers'].map(followers_count)
3 dat['Following_count'] = dat['Influencers'].map(following_count)
4 dat['Number_of_Tweets'] = dat['Influencers'].map(num_of_tweets)
5 dat['Screen_name'] = dat['Influencers'].map(screen_name)
6
7 #removing rows with no values
8 dat.dropna(inplace=True)
9
10 #condition to only choose influencers with more than 1000 followers
11 dat = dat[dat['Follower_count'] > 1000]
12
13 #removing rows with duplicate values and keeping the first occurrence
14 dat.drop_duplicates(subset='Screen_name', keep="first", inplace=True)
15 dat.set_index('Screen_name')

```

Influencers Follower\_count Following\_count Number\_of\_Tweets

Screen\_name

Influencers	Follower_count	Following_count	Number_of_Tweets
ABdeVilliers17	@ABdeVilliers17	8233087.0	181.0

✓ 0s completed at 1:06 PM

Influencers Follower\_count Following\_count Number\_of\_Tweets

Screen\_name

Influencers	Follower_count	Following_count	Number_of_Tweets
ABdeVilliers17	@ABdeVilliers17	8233087.0	181.0
KP24	@KP24	3832355.0	425.0
MinnieDlamini	@MinnieDlamini	4222114.0	917.0
News24	@News24	4347894.0	719.0
DJZinhle	@DJZinhle	3315449.0	9310.0
Julius_S_Malema	@Julius_S_Malema	3636469.0	656.0
SuperSportTV	@SuperSportTV	2985485.0	592.0
casspernyovest	@casspernyovest	3479441.0	1351.0
PearlThusi	@PearlThusi	2914713.0	6232.0

## Obtaining the follower count, friend count, number of tweets from the Twitter API

python code to search and download tweets of influencers.

```
✓ [12] 1 #python code to search and download tweets of influencers.
0s      2 #importing all the necessary libraries needed
        3 import sys
        4 import os
        5 import json
        6 import pandas as pd
        7 import matplotlib.pyplot as plt
        8 import re
        9 import string
        10
        11 # to view all columns
        12 pd.set_option("display.max.columns", None)
        13
        14 #Import the necessary methods from tweepy library
        15 import tweepy
        16 from tweepy.streaming import StreamListener
        17 from tweepy import OAuthHandler
        18 from tweepy import Stream
        19
        20 #sentiment analysis package
        21 from textblob import TextBlob
        22
        23 #general text pre-processor
        24 import nltk
```

```
✓ [13] 1 class tweetsearch():
0s      2     '''
        3     This is a basic class to search and download twitter data.
        4     You can build up on it to extend the functionalities for more
        5     sophisticated analysis
        6     '''
        7     def __init__(self, cols=None, auth=None):
        8         #
        9         if not cols is None:
        10             self.cols = cols
        11         else:
        12             self.cols = ['id', 'created_at', 'source', 'original_text', 'clean_text',
        13                          'sentiment', 'polarity', 'subjectivity', 'lang',
        14                          'favorite_count', 'retweet_count', 'original_author',
        15                          'possibly_sensitive', 'hashtags',
        16                          'user_mentions', 'place', 'place_coord_boundaries']
        17
        18         if auth is None:
        19
        20             #Variables that contains the user credentials to access Twitter API
        21             consumer_key = 'qprNvAYFLJ0PI3TDgLN5jOXSh'
        22             consumer_secret = 'oPNiksEiB1ljDospQlaYPZIdrV5EkvHYx6HfKkHTRKGcRFogCuG'
        23             access_token = '1449350837407465473-EPnn7XP2nOUNFstG2poP2tmA4SRatw'
        24             access_token_secret = 'afj6I7No4UpiBjhg2blWou62BCTtJXNubQniZwgdlN70y'
        25
        26
        27
        28             #This handles Twitter authentication and the connection to Twitter Streaming API
        29             auth = OAuthHandler(consumer_key, consumer_secret)
```

✓  
46s

```
[14] 1 """
      2 #creating a csv file for the influencers twitter data and
      3 calling the tweetsearch function
      4 """
      5 tweets_file2 = 'Influencers.csv'
      6 ts = tweetsearch()
      7
      8 #getting the data for the top African Influencers
      9 for handle in dat['Influencers']:
10     ##get data on username
11     df = ts.get_tweets(handle, csvfile=tweets_file2)
12     print("Printing for user:",handle)
```

```
Printing for user: @ABdeVilliers17
Printing for user: @KP24
Printing for user: @MinnieDlamini
Printing for user: @News24
Printing for user: @DJZinhle
Printing for user: @Julius_S_Malema
Printing for user: @SuperSportTV
Printing for user: @casspernyovest
Printing for user: @PearlThusi
```

```
✓ [15] 1 #importing the twitter data for top african influencers
      2 data_inf = pd.read_csv(tweets_file2)
      3
      4 #creating a dataframe from the data imported
      5 inf_df = pd.DataFrame(data_inf)
      6
      7 #python code to calculate the total retweets and likes
      8 new2 = inf_df.groupby('original_author', sort=False)['favorite_count', 'retweet_count'].agg('sum')
      9 new2.columns = ['Favorite_count', 'Retweet_count']
10
11 #merging the previous dataframe with the favorite count and retweet count
12 new_inf_df = pd.merge(dat, new2, left_on='Screen_name', right_on='original_author', how='outer')
13 new_inf_df.dropna(inplace=True)
14
15 #calculating the popularity score
16 new_inf_df['Popularity_score'] = new_inf_df['Retweet_count'] + new_inf_df['Favorite_count']
17
18 #calculating the reach score
19 new_inf_df['Reach_score'] = new_inf_df['Follower_count'] - new_inf_df['Following_count']
20 new_inf_df
```

/usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:8: FutureWarning: Indexing with multiple keys (implicitly converted to a tuple of keys) will be deprecated

Influencers	Follower_count	Following_count	Number_of_Tweets	Screen_name	Favorite_count	Retweet_count	Popularity_score	Reach_score
-------------	----------------	-----------------	------------------	-------------	----------------	---------------	------------------	-------------

```

12 new_inf_df = pd.merge(data, new_inf_df, left_on='screen_name', right_on='original_author', how='outer')
13 new_inf_df.dropna(inplace=True)
14
15 #calculating the popularity score
16 new_inf_df['Popularity_score'] = new_inf_df['Retweet_count'] + new_inf_df['Favorite_count']
17
18 #calculating the reach score
19 new_inf_df['Reach_score'] = new_inf_df['Follower_count'] - new_inf_df['Following_count']
20 new_inf_df

```

/usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:8: FutureWarning: Indexing with multiple keys (implicitly converted to a tuple of keys) will be deprecated, use a list instead

	Influencers	Follower_count	Following_count	Number_of_Tweets	Screen_name	Favorite_count	Retweet_count	Popularity_score	Reach_score
0	@ABdeVilliers17	8233087.0	181.0	5669.0	ABdeVilliers17	4461916	352979	4814895	8232906.0
1	@KP24	3832355.0	425.0	25509.0	KP24	806005	49797	855802	3831930.0
2	@MinnieDlamini	4222114.0	917.0	25170.0	MinnieDlamini	559230	29127	588357	4221197.0
3	@News24	4347894.0	719.0	383855.0	News24	30925	7801	38726	4347175.0
4	@DJZinhle	3315449.0	9310.0	71627.0	DJZinhle	76068	6071	82139	3306139.0
5	@Julius_S_Malema	3636469.0	656.0	42730.0	Julius_S_Malema	93748	16241	109989	3635813.0
6	@SuperSportTV	2985485.0	592.0	331826.0	SuperSportTV	98942	13658	112600	2984893.0
7	@casspernyovest	3479441.0	1351.0	195574.0	casspernyovest	179290	14342	193632	3478090.0
8	@PearlThusi	2914713.0	6232.0	24975.0	PearlThusi	281579	22793	304372	2908481.0

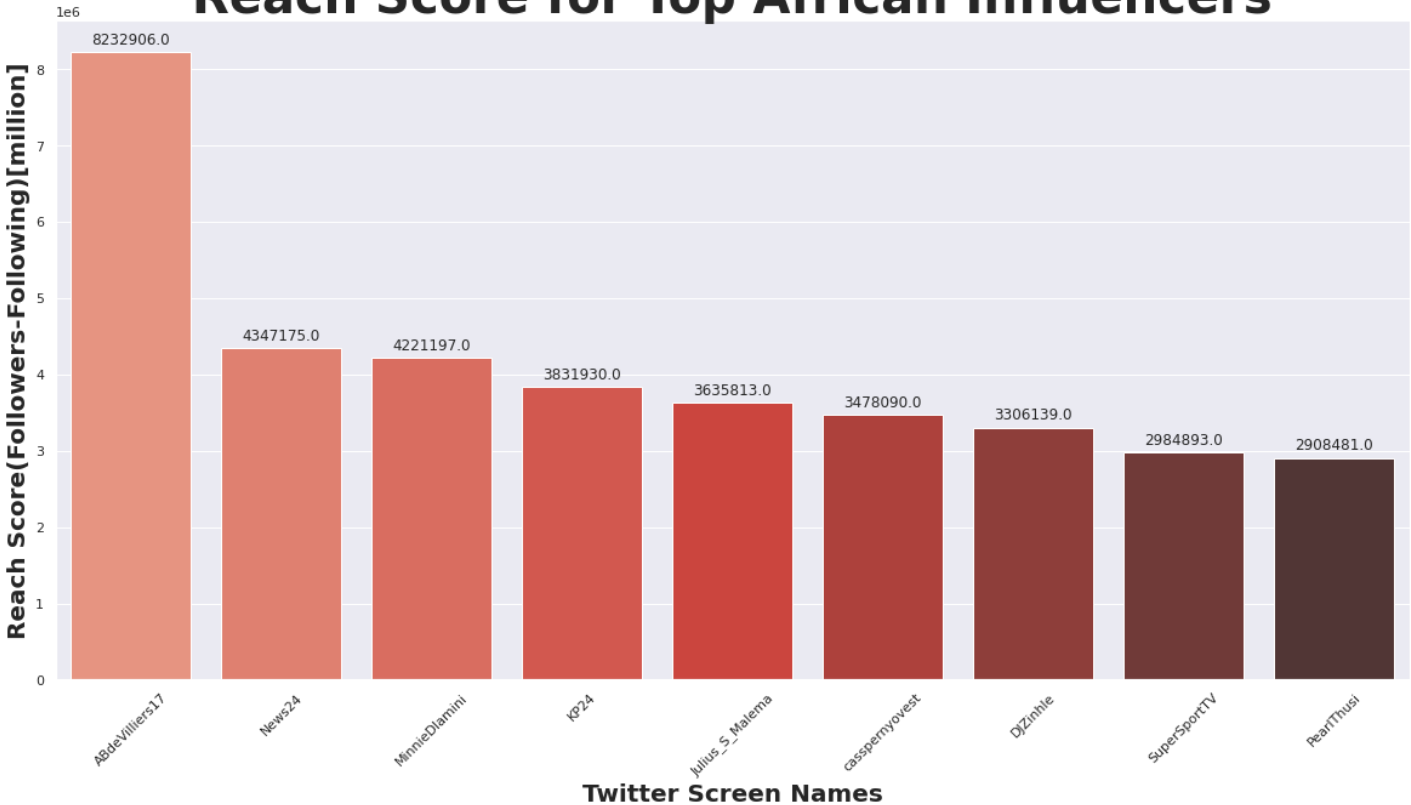
## Results

```

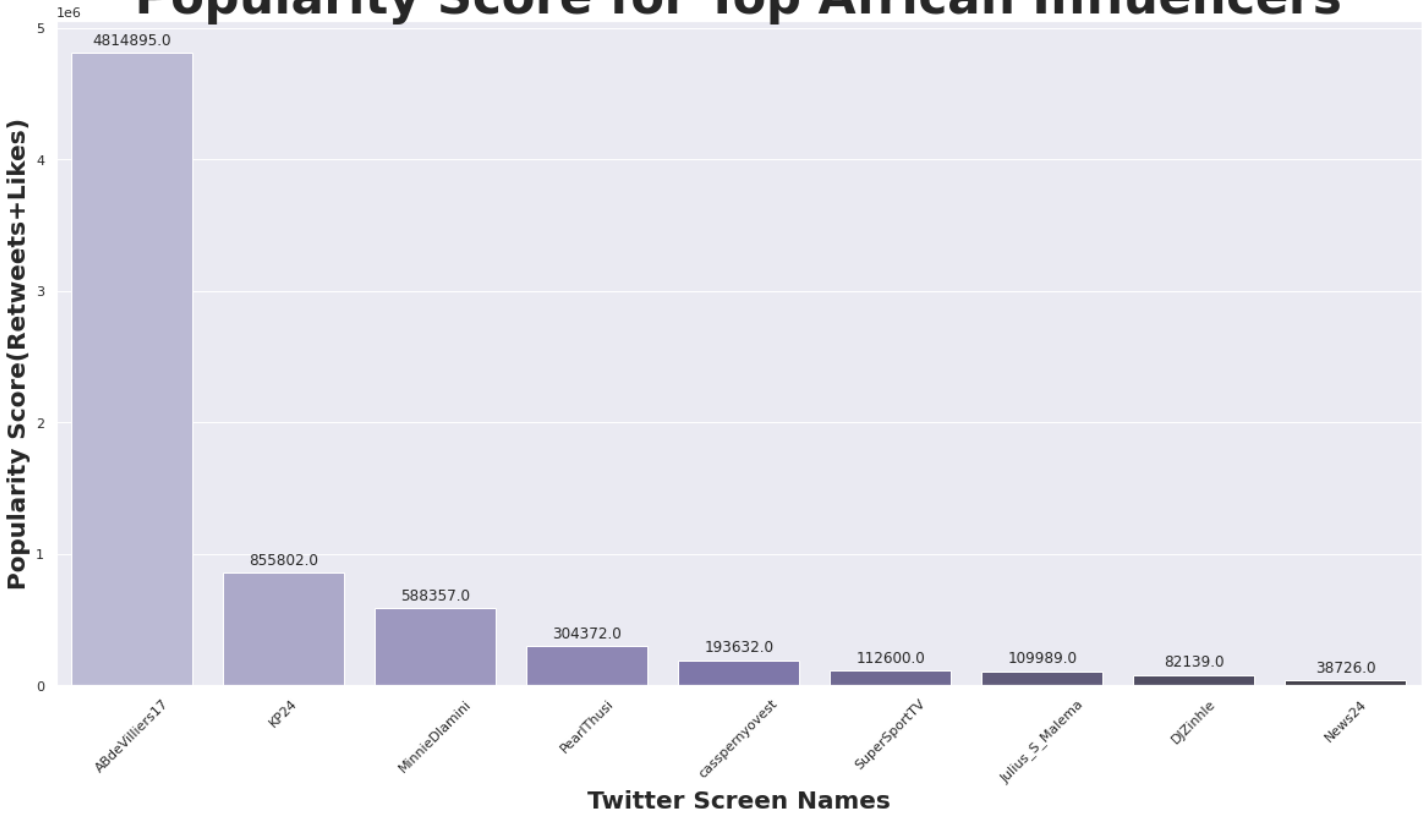
[16] 1 ##data visualization - influencers
      2
      3 #bar plot - reach score
      4 plt.figure(figsize=(20,10)) #customizing the size of the plot
      5 sns.set(style="darkgrid") #customizing the style of the plot
      6
      7 #visualizing the data using bar plot
      8 ax = sns.barplot(x='Screen_name', y='Reach_score', palette="Reds_d",
      9                 data=new_inf_df.sort_values(by='Reach_score', ascending=False)[0:10]
     10                 )
     11
     12 #getting the values of the data
     13 for p in ax.patches:
     14     ax.annotate(format(p.get_height()), (p.get_x() + p.get_width() / 2.,
     15     p.get_height()), ha = 'center', va = 'center',
     16     xytext = (0, 10), textcoords = 'offset points')
     17
     18 #setting the parameters for the title, x and y labels of the plot
     19 ax.set_title("Reach Score for Top African Influencers", size=40, weight='bold')
     20 ax.set_xlabel("Twitter Screen Names", size=20, weight='bold')
     21 ax.set_ylabel("Reach Score(Followers-Following)[million]", size=20, weight='bold')
     22
     23 #changing the rotation of the x axis tick labels
     24 for item in ax.get_xticklabels():

```

# Reach Score for Top African Influencers



# Popularity Score for Top African Influencers



## **6. Conclusion**

In this work, we highlight the importance of planning a new digital campaign in a particular region through obtaining the twitter screen names from a particular website, we understood who the social media influencers are in particular region and in which area fall into (e.g., politics, fashion, art, etc.). This is key to companies as they don't want to be associated with the wrong influencers (ethnic and religious polarization figures for example). This solution gives the company understand greatly about that region where they want to amplify their planned digital campaign and growing their business through influencers. Overall, twitter is a great avenue to try to better understand the general public's views, sentiments and understanding of certain issues, in byte size pieces.

## **7. Future Enhancements**

In the future, we would like to continue this line of analysis and extend into all areas of social media (Facebook, YouTube, Instagram, etc.) to grasp the impact it's on sport consumers. Strengthening this understanding could lead the way to simpler more effective sport marketing strategies designed to attach with fans and enhance the social affiliation and relationship.