

1 Open Source Tutorial 1

In the following you will need to register for a Twitter account API key. If you have a Twitter account, go to the following page and create your App:

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

Once you have this, take a note of your Consumer Key, Consumer Secret, Access Key and Access Secret.

1. We will first run some Python code:

```
1 https://repl.it/@billbuchanan/twitter02
```

```
1 import tweepy
2
3 import sys
4
5 consumer_key = ''
6 consumer_secret = ''
7 access_key= ''
8 access_secret = ''
9
10 auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
11 auth.set_access_token(access_key, access_secret)
12 api_obj = tweepy.API(auth)
13 def get_user_tweets(username, count):
14     tweets = api_obj.user_timeline(screen_name=username, count=count)
15     res=[]
16     tweets_for_csv = [tweet.text for tweet in tweets]
17     for j in tweets_for_csv:
18         res.append(j)
19     return(res)
20
21 user="billatnapier"
22 if (len(sys.argv)>1):
23     user=str(sys.argv[1])
24
25 rtn=get_user_tweets(user,5)
26 for s in rtn:
27     print (s.encode('utf-8'))
28     print ()
```

Modify the code so that it finds the last five tweets from the following users:

- "bbc"

- "cisco"
- "microsoft"
- "facebook"
- "cmn"

2. We will now uses sentimental analysis:

```

1 import tweepy
2 import tweepy as tw
3 from textblob import TextBlob
4 import sys
5
6 consumer_key = ''
7 consumer_secret = ''
8 access_key= ''
9 access_secret = ''
10
11 auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
12 auth.set_access_token(access_key, access_secret)
13 api_obj = tweepy.API(auth)
14
15
16 search_term = "cyber+security"
17
18 if (len(sys.argv)>1):
19     search_term=(sys.argv[1])
20
21 print ("Search term: ",search_term)
22
23 search_results = api_obj.search(q=search_term, count=10)
24
25 res = []
26
27 for tw in search_results:
28     res.append(tw.text.encode('ascii', 'ignore'))
29
30 sentiment_objects = [TextBlob(tweet.decode()) for tweet in res]
31
32 sentiment_values = [[tweet.sentiment.polarity, str(tweet)] for tweet in
    sentiment_objects]
33
34 score=0
35 for s in sentiment_values:
36     score = score+float (s[0])
37
38 print ("Overall score from 10 tweets: ", score)
39 print

```

```

40
41 i=0
42 for s in sentiment_values:
43     i=i+1
44     print (i,end=' ')
45     if float (s[0]) > 0:
46         print ('Positive ', end=' ')
47     elif float(s[0]) == 0:
48         print ('Neutral ',end=' ')
49     else:
50         print ('Negative ',end=' ')
51
52     print ("Score: ",s[0], "Text: ",s[1])
53     print

```

Perform searches for the following words:

- "cybersecurity"
- "bbc"
- "edinburgh"
- "glasgow"
- "scotland"
- "cryptography"

3. if you have a Reddit account, create an App from:

```
1 https://www.reddit.com/prefs/apps
```

Next use the following code:

```
1 https://repl.it/@billbuchanan/twitter02
```

```

1 import praw
2 import pandas as pd
3 import datetime as dt
4 import sys
5
6
7 reddit = praw.Reddit(client_id='xxxxx', \
8                       client_secret='xxxxx', \
9                       user_agent='xxxx', \
10                      username='xxxxx', \
11                      password='xxxx')

```

```

12 |
13 | search_term='cybersecurity'
14 | option="1"
15 | if (len(sys.argv)>1):
16 |     search_term=(sys.argv[1])
17 |
18 | if (len(sys.argv)>2):
19 |     option=(sys.argv[2])
20 |
21 | print ("Search term: ",search_term)
22 | print ("Option: ",option)
23 |
24 | subreddit = reddit.subreddit(search_term)
25 | resp = subreddit.top(limit=10)
26 |
27 | if (option=="2"): resp = subreddit.hot(limit=10)
28 | if (option=="3"): resp = subreddit.new(limit=10)
29 | if (option=="4"): resp = subreddit.controversial(limit=10)
30 |
31 | for submission in resp:
32 |     print ("=ID: ",submission.id)
33 |     print (" Title: ",submission.title.encode('ascii', 'ignore'))
34 |     print (" Score: ",submission.score)
35 |     print (" URL: ",submission.url)
36 |     print (" Text: ",submission.selftext[:100])

```

Find the top three hot, top three new, and top three controversial redds from:

- "edinburgh"
- "glasgow"

4. Now we will use Reddit for a search term. We will use the following code:

```
1 | https://repl.it/@billbuchanan/twitter02
```

```

1 | import praw
2 | import pandas as pd
3 | import datetime as dt
4 | import sys
5 |
6 |
7 | reddit = praw.Reddit(client_id='xxxxx', \
8 |                     client_secret='xxxxx', \
9 |                     user_agent='xxxx', \
10 |                     username='xxxxx', \

```

```

11         password='xxxx')
12
13 search_term='cybersecurity'
14 option="1"
15 if (len(sys.argv)>1):
16     search_term=(sys.argv[1])
17
18 if (len(sys.argv)>2):
19     option=(sys.argv[2])
20
21 print ("Search term: ",search_term)
22 print ("Option: ",option)
23
24 subreddit = reddit.subreddit(search_term)
25 resp = subreddit.top(limit=10)
26
27 if (option=="2"): resp = subreddit.hot(limit=10)
28 if (option=="3"): resp = subreddit.new(limit=10)
29 if (option=="4"): resp = subreddit.controversial(limit=10)
30
31 for submission in resp:
32     print ("=ID: ",submission.id)
33     print (" Title: ",submission.title.encode('ascii', 'ignore'))
34     print (" Score: ",submission.score)
35     print (" URL: ",submission.url)
36     print (" Text: ",submission.selftext[:100])

```

Find the top three hot, top three new, and top three controversial redds from:

- "edinburgh"
- "glasgow"

5. Now we will use Shodan to search. For this create an account on Shodan and generate an API key:

```
1 https://repl.it/@billbuchanan/shodan001
```

```

1
2 #!/usr/bin/env python
3 #
4 # query-summary.py
5 # Search Shodan and print summary information for the query.
6 #
7 # Author: achillean
8

```

```

9 import shodan
10 import sys
11
12
13 # Configuration
14 API_KEY = ''
15
16 # The list of properties we want summary information on
17 FACETS = [
18     'org',
19     'domain',
20     'port',
21     'asn',
22
23     # We only care about the top 5 countries, this is how we let Shodan
24     # know to return 5 instead of the
25     # default 10 for a facet. If you want to see more than 10, you could
26     # do ('country', 1000) for example
27     # to see the top 1,000 countries for a search query.
28     ('country', 5),
29 ]
30
31 FACET_TITLES = {
32     'org': 'Top 5 Organizations',
33     'domain': 'Top 5 Domains',
34     'port': 'Top 5 Ports',
35     'asn': 'Top 5 Autonomous Systems',
36     'country': 'Top 5 Countries',
37 }
38
39 # Input validation
40 if len(sys.argv) == 1:
41     print ('Usage: %s <search query>' % sys.argv[0])
42     sys.exit(1)
43
44 try:
45     # Setup the api
46     api = shodan.Shodan(API_KEY)
47
48     # Generate a query string out of the command-line arguments
49     query = ' '.join(sys.argv[1:])
50
51     # Use the count() method because it doesn't return results and doesn't
52     # require a paid API plan
53     # And it also runs faster than doing a search().
54     result = api.count(query, facets=FACETS)
55
56     print ('Query: %s' % query)

```

```

56     print ('Total Results: %s\n' % result['total'])
57
58     # Print the summary info from the facets
59     for facet in result['facets']:
60         print (FACET_TITLES[facet])
61
62         for term in result['facets'][facet]:
63             print ('%s: %s' % (term['value'], term['count']))
64
65         # Print an empty line between summary info
66         print ('')
67
68 except Exception:
69     print ('Error: %s' % e)
70     sys.exit(1)

```

Now find the Top 5 domains, Top 5 countries and Top 5 ports for the following:

- "apache"
- "mysql"
- "cisco"
- "openssh"
- "realvnc"