Untitled2

February 14, 2020

1 API Authentication and streaming Tweets

Below is the code to filter Twitter streams and to filter tweets according to partifular keywords.

It can help us determine what the trands in live tweets.

First of all a Twitter app is created

[33]:
```
import tweepy
import json
access_token = "778529934-We2jgtHTV3R89nBbkXiss3ominbASeCadohYcq5z"
access_token_secret = "ApOdgjY3vX9o5q6NJRleKRPFTNqCgGhdWrQ7CpirhBbu4"
consumer_key = "7aPp3lopI1WsddYeNz9vMMMLL"
consumer_secret = "PisJONATEiJHwZ6p9ezJQk2NFrYjdNuwjFu9WqBVJuGyRZ6svV"
auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_token_secret)
class MyStreamListener(tweepy.StreamListener):
    def __init__(self, api=None):
        super(MyStreamListener, self).__init__()
        self.num_tweets = 0
        self.file = open("tweets.txt", "w")
    def on_status(self, status):
        tweet = status._json
        self.file.write( json.dumps(tweet) + '\n' )
        self.num_tweets += 1
        if self.num_tweets < 100:
            return True
        else:
            return False
        self.file.close()


    def on_error(self, status):
        print(status)
```

2 Streaming Tweets

In below code Twitter stream objects are created to filter Tweets according to particular keywords.

A Stream object is created with authentication by passing tweepy.Stream() the authentication handler auth and the Stream listener l;

To filter Twitter streams, a list is passed to the track argument in stream.filter()containing the desired keywords 'clinton', 'trump', 'sanders', and 'cruz'

[35]:
```python
# Create Streaming object and authentication
l = MyStreamListener()
stream = tweepy.Stream(auth, l)

# This line filters Twitter Streams to capture data by keywords:
stream.filter(track=['clinton', 'trump', 'sanders', 'cruz'])
```

3 Loading and exploring Twitter data

Now that Twitter data is sitting locally in a text file, it's time to explore it! In this exercise, the Twitter data Now that you've got your Twitter data sitting locally in a text file, it's time to explore it! This is what you'll do in the next few interactive exercises. In this exercise, the Twitter data is read into a list: tweets_data.

[37]:
```python
# Import package
import json

# String of path to file: tweets_data_path
tweets_data_path = 'tweets.txt'

# Initialize empty list to store tweets: tweets_data
tweets_data = []

# Open connection to file
tweets_file = open(tweets_data_path, "r")

# Read in tweets and store in list: tweets_data
for line in tweets_file:
    tweet = json.loads(line)
    tweets_data.append(tweet)

# Close connection to file
tweets_file.close()

# Print the keys of the first tweet dict
print(tweets_data[0].keys())
```

```
dict_keys(['created_at', 'id', 'id_str', 'text', 'source', 'truncated',
'in_reply_to_status_id', 'in_reply_to_status_id_str', 'in_reply_to_user_id',
'in_reply_to_user_id_str', 'in_reply_to_screen_name', 'user', 'geo',
'coordinates', 'place', 'contributors', 'quoted_status_id',
'quoted_status_id_str', 'quoted_status', 'quoted_status_permalink',
'is_quote_status', 'quote_count', 'reply_count', 'retweet_count',
'favorite_count', 'entities', 'favorited', 'retweeted', 'filter_level', 'lang',
'timestamp_ms'])
```

4 Twitter data to a dataframe

Now that the Twitter data is in a list of dictionaries, tweets_data, where each dictionary corresponds to a single tweet. Next,the text and language of each tweet is extracted. The text in a tweet, t1, is stored as the value t1['text']; similarly, the language is stored in t1['lang']. A DataFrame is built in which each row is a tweet and the columns are 'text' and 'lang'.

[39]:
```python
# Import package
import pandas as pd

# Build DataFrame of tweet texts and languages
df = pd.DataFrame(tweets_data, columns=['text','lang'])

# Print head of DataFrame
print(df.head())
```

```
                                                 text lang
0                           Start with Obama and Clinton   en
1  RT @svdate: The president loves to brag that h…   en
2  @Turfline @RealJamesWoods Yep, sociopathic nar…   en
3  RT @maddow: "Barr intervened in multiple cases…   en
4  15/\nFace the truth, educate yourself, throw o…   en
```

5 A little bit of Twitter text analysis

Now that DataFrame of tweets is set up, a text analysis is done to count how many tweets contain the words 'clinton', 'trump', 'sanders' and 'cruz'. In the pre-exercise code, the following function word_in_text() is defined, which will tell whether the first argument (a word) occurs within the 2nd argument (a tweet).

[41]:
```python
import re

def word_in_text(word, text):
    word = word.lower()
    text = text.lower()
    match = re.search(word, text)

    if match:
        return True
    return False
```

[42]:
```
# Initialize list to store tweet counts
[clinton, trump, sanders, cruz] = [0, 0, 0, 0]

# Iterate through df, counting the number of tweets in which
# each candidate is mentioned
for index, row in df.iterrows():
    clinton += word_in_text('clinton', row['text'])
    trump += word_in_text('trump',row['text'] )
    sanders += word_in_text('sanders', row['text'])
    cruz += word_in_text('cruz', row['text'])
```

6 Plotting your Twitter data

Now that the number of tweets that each candidate was mentioned in is counted, a bar chart of this data can be plotted. Statistical data visualization library seaborn will be used, First seaborn as sns is imported. Then a barplot of the data is constructed using sns.barplot, passing it two arguments:

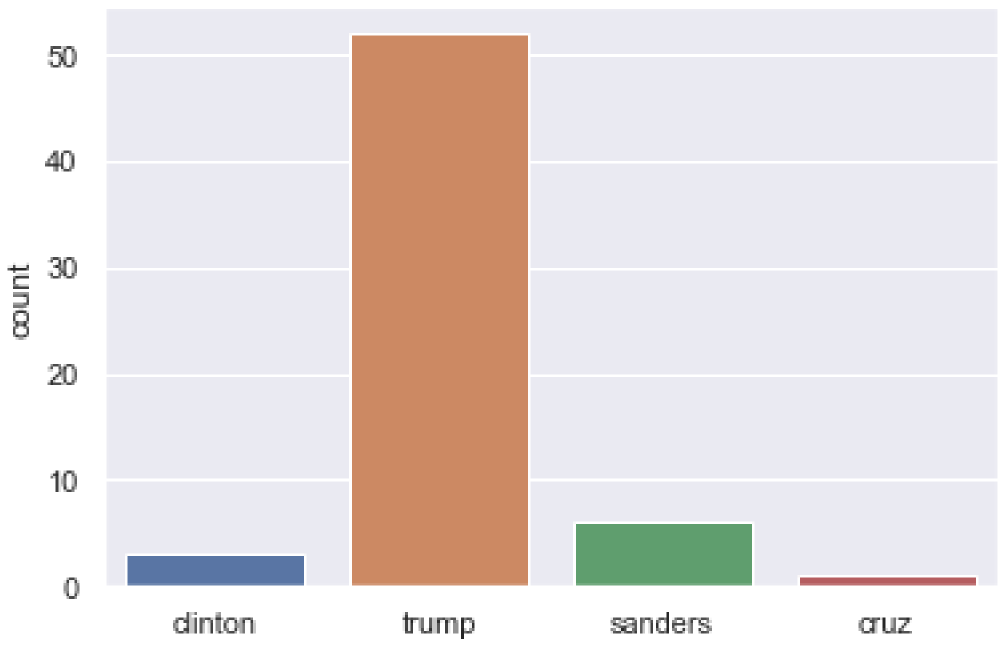
1. a list of labels and
2. a list containing the variables you wish to plot (clinton, trump and so on.)

[44]:
```
# Import packages
import matplotlib.pyplot as plt
import seaborn as sns

# Set seaborn style
sns.set(color_codes=True)

# Create a list of labels:cd
cd = ['clinton', 'trump', 'sanders', 'cruz']

# Plot the bar chart
ax = sns.barplot(cd, [clinton, trump, sanders, cruz])
ax.set(ylabel="count")
plt.show()
```


count
50
40
30
20
10
0
clinton
trump
sanders
cruz

```
[ ]:
```