#### **CHAPTER 1 :**

#### **Basics of Analyzing Twitter Data**

**Why Analyze Twitter Data?**

There are many reasons you may want to analyze Twitter data. Which of these is NOT an area of data science you could use analyzing Twitter data for?

**Answer the question**

**50 XP**

**Possible Answers**

* 

Analyzing the mentions of each political party in an election.

press1

* 

Detecting the reactions to the introduction of a new product.

press2

* 

Understanding the geographical scope of discussion of a news story.

press3

* 

Uncovering the motives of Twitter users following a hashtag. **(A)**

press4

Correct! You can't identify users unless they tweet.

**Uses of Twitter analysis**

You've been asked to identify the success (or failure) of a particular product. What Twitter analysis strategy could you use to best execute this?

**Answer the question**

**50 XP**

**Possible Answers**

* 

Collect mentions of the product and identify if people are talking about it positively.

press1

* 

Examine the size of the retweet network mentioning the product.

press2

* 

Analyzing the geographical penetration of users mentioning the product.

press3

* 

All of the above. **(A)**

press4

Correct! All of these are good ways of analyzing how a product may be received.

**Twitter APIs**

**True or False** : I could collect data from last year based on keyword searches with the Streaming API.

**Answer the question**

**50 XP**

**Possible Answers**

* 

True: The Streaming API allows historical data collection on keywords, user IDs, and locations.

press1

* 

False: The Streaming API only allows real-time data collection on ads.

press2

* 

False: The Streaming API only allows real-time data collection on keywords, user IDs, and locations. **(A)**

press3

* 

False: The Streaming API only allows access from the past week.

press4

Correct! The Streaming API only allows real-time data collection.

**Setting up tweepy authentication**

In the video, we saw how tweepy can be used to collect Twitter data with the Streaming API. tweepy requires a Twitter API key to authenticate with Twitter.

In this exercise, you will load several objects from tweepy and set up the authentication for the package.

The API keys access\_token, access\_token\_secret, consumer\_key, and consumer\_secret have already been defined for you.

**Instructions**

**100 XP**

* Import OAuthHandler and API from the tweepy module.
* Pass consumer\_key and consumer\_secret to OAuthHandler.
* Set the access tokens with access\_token and access\_token\_secret.
* Pass the auth object to the API.

from tweepy import OAuthHandler

from tweepy import API

# Consumer key authentication

auth = OAuthHandler(consumer\_key , consumer\_secret)

# Access key authentication

auth.set\_access\_token(access\_token , access\_token\_secret)

# Set up the API with the authentication handler

api = API(auth)

Great! You are now authenticated.

**Collecting data on keywords**

Now that we've set up the authentication, we can begin to collect Twitter data. Recall that with the Streaming API, we will be collecting real-time Twitter data based on either a sample or filtered by a keyword.

In our example, we will collect data on any tweet mentioning #rstats or #python in the tweet text, username, or user description with the filter endpoint.

The SListener module has already been defined and imported for you. You can find the full code for this module [**here**](https://github.com/SocialDataAnalytics-Winter2018/lab04/blob/master/slistener.py).

**Instructions**

**100 XP**

* Import Stream from tweepy.
* Set keywords\_to\_track to a list containing #rstats and #python.
* Pass the auth and listen objects to Stream.
* Set the keyword argument track equals to keywords\_to\_track.

from tweepy import Stream

# Set up words to track

keywords\_to\_track = ['#rstats' , '#python']

# Instantiate the SListener object

listen = SListener(api)

# Instantiate the Stream object

stream = Stream(auth , listen)

# Begin collecting data

stream.filter(track = keywords\_to\_track)

Good job! You are now collecting tweets.

**Loading and accessing tweets**

In the video, we loaded a tweet we collected using tweepy into Python. Tweets arrive from the Streaming API in JSON format and need to be converted into a Python data structure.

In this exercise, we'll load a single tweet into Python and print out some fields.

The tweet JSON has been loaded for you and is stored in tweet\_json.

**Instructions**

**100 XP**

* Import the json module.
* Convert the tweet JSON stored in tweet\_json from JSON to Python object using json's .loads() method.
* Print the tweet text and id using the appropriate keys.

# Load JSON

import json

# Convert from JSON to Python object

tweet = json.loads(tweet\_json)

# Print tweet text

print(tweet['text'])

# Print tweet id

print(tweet['id'])

<script.py> output:

Writing out the script of my @DataCamp class and I can't help but mentally read it back to myself in @hugobowne's voice.

986973961295720449

**Accessing user data**

Much of the data which we want to know about the Twitter data is stored in child JSON objects. We will access several parts of the user's information with the user child JSON object.

The tweet from the previous exercise has been loaded for you.

**Instructions**

**100 XP**

* Print the user handle with key screen\_name.
* Print the user follower count with key followers\_count.
* Print the user self-defined location with key location.
* Print the user self-defined description with key description.

# Print user handle

print(tweet['user']['screen\_name'])

# Print user follower count

print(tweet['user']['followers\_count'])

# Print user location

print(tweet['user']['location'])

# Print user description

print(tweet['user']['description'])

<script.py> output:

alexhanna

4267

Toronto, ON

Assistant professor @UofT. Protest, media, computation. Trans. Roller derby athlete @TOROLLERDERBY (Kate Silver #538). She/her

**Accessing retweet data**

Now we're going to work with a tweet JSON that contains a retweet. A retweet has the same structure as a regular tweet, except that it has another tweet stored in retweeted\_status.

The new tweet has been loaded as rt.

**Instructions**

**100 XP**

* Print the text of the tweet.
* Print the text of the tweet which has been retweeted, which is contained in retweeted\_status.
* Print the user handle of the tweet.
* Print the user handle of the tweet which has been retweeted, which is contained in retweeted\_status.

# Print the text of the tweet

print(rt['text'])

# Print the text of tweet which has been retweeted

print(rt['retweeted\_status']['text'])

# Print the user handle of the tweet

print(rt['user']['screen\_name'])

# Print the user handle of the tweet which has been retweeted

print(rt['retweeted\_status']['user']['screen\_name'])

<script.py> output:

RT @hannawallach: ICYMI: NIPS/ICML/ICLR are looking for a full-time programmer to run the conferences' submission/review processes. More in…

ICYMI: NIPS/ICML/ICLR are looking for a full-time programmer to run the conferences' submission/review processes. M… https://t.co/aB9Y5tTyHT

alexhanna

hannawallach

**CHAPTER 2 :**

#### **Processing Twitter text**