Information School Twitter Project Handout

Author: Ronak Sharma

Status: Student at The University of Sheffield

Date: 16-07-2020

Contents

1 Introduction

1.1 Aim

This research project is led by Dr Angela Lin and Dr Pamela Abbott in Information School at The University of Sheffield, UK . It focuses on collecting the images of home-working spaces and their captions published on Twitter with hashtags such as *#workingfromhome, #homeoffice, #WFH and #reomoteworking.* Such information will help understand the trends in how people are configuring their home-working spaces with the onset of COVID-19.

1.2 Motivation

The project will explore the different configuration of physical and digital tools and other personal artefacts used by professionals in a home-based environment to create personal digital workspaces. As a result, such insights into the home-working arrangements under mandatory requirements to work from home during the 'COVID-19' epidemic will provide practical inputs into the design of future home-based teleworking scenarios.

1.3 Objectives

The following defines the objectives for this project:

1. **Programme:** Develop a bespoke Python programme for collecting images and its meta data meeting the user defined constraints. This data shall be extracted from Twitter;
2. **Content Analysis:** Analyse the images and captions of the Twitter data obtained via the above programme. This involves devising various'categories' and then applying these 'categories' as 'codes' to images and caption data;

2   Accessing the Code via GitHub

Before accessing the code, it is recommended to follow the below two steps:

1. **GitHub Personal Account:** Create a GitHub Account for ease of usage and future modifications
2. **Environment:** Download Anaconda by clicking this link and scrolling to the bottom. All Python versions above 3.7 are deemed acceptable. The installation steps are easy to follow through

Twitter GitHub Repository Link: https://github.com/RonakSharma1/Twitter_Data_Collection.git

2.1   Information for Users

This section describes the procedure to be followed by a user for using the programme developed by the author:

1. Click and visit the GitHub repository link provided earlier. This routes you to the location where this project is saved
2. Once logged and in the repository, click on 'Code' button highlighted in 'Green'. From the pop-down menu, click on 'Download ZIP'
3. Once the above file is unzipped, you shall see the same contents as in GitHub repository
4. Open 'Anaconda' and then open 'Spyder' by pressing the 'Launch' button. Ensure that 'Spyder' version is 4.0 or above
5. From 'Spyder', using the 'File' toolbar, open the downloaded Python file i.e. *TwitterAPI_Socialmedia_Extract.py*
6. To change the colour scheme, you can click on 'Preferences' as shown in Figure 1
7. To add/remove panes, navigate to 'View' and then to 'Panes'. Ensure that you have two panes visible on the screen i.e. 'Editor' and 'iPython Console'
8. To change the layout of panes, navigate to 'View' and then to 'Lock Panes and Toolbars'. This allows to unlock the panes and move them around

Figure 1 shows the author's layout. The left pane is the 'Editor' and the right pane is the 'iPython' Console
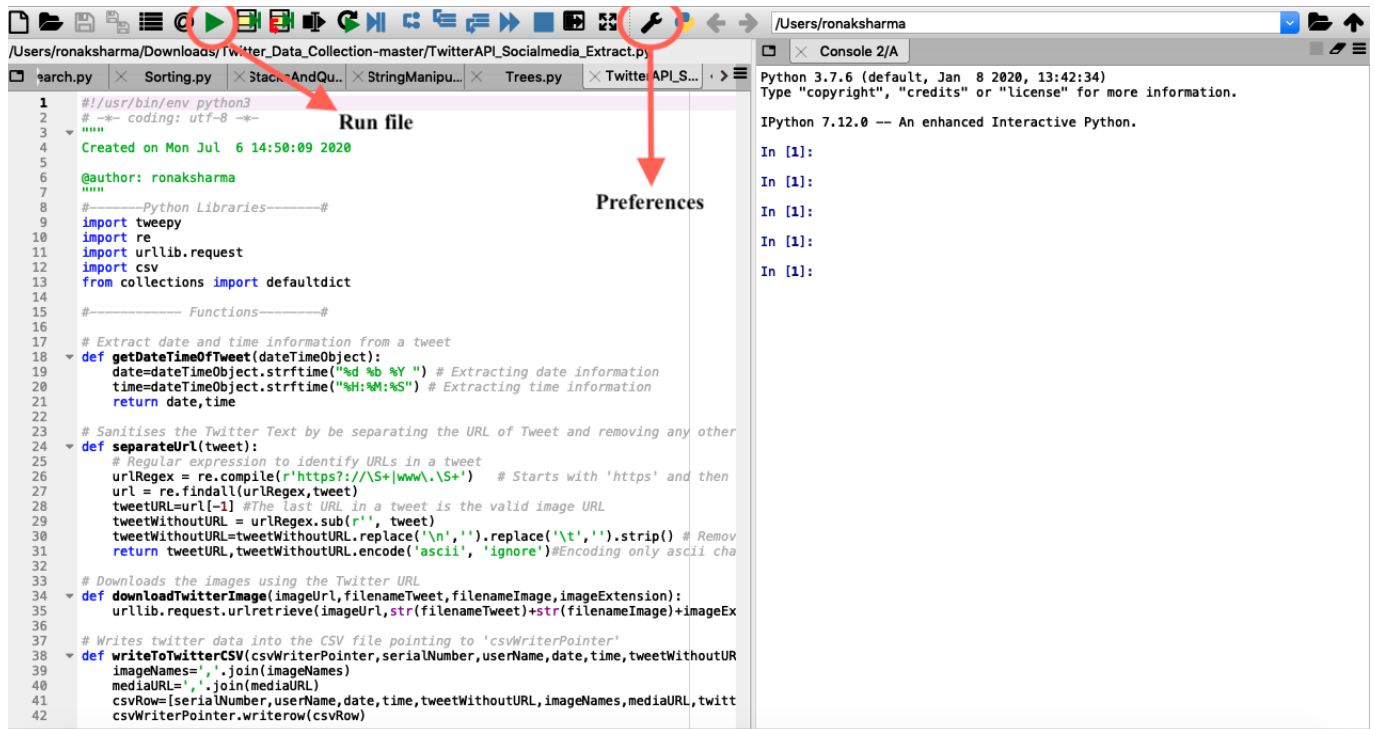
Figure 1: Spyder layout screenshot

2.2   Developer Information

This section describes the procedure to be followed by a developer for building upon the work produced by the author:

1. **Forking**: Follow this tutorial to fork the GitHub repository. This creates a personal copy of the author's Twitter repository in your own account thus allowing to making modifications

2. **Cloning:** Follow this tutorial to clone the above forked repository. This creates a local copy of this repository which now allows you to perform other GitHub features such as 'push', 'pull' and 'commit'

3   Twitter Developer Account

In order access the Twitter API, a developer account needs to be created. Follow this link How to get started with the Twitter APIs and click on 'Twitter Developer Account' to create this account. Having this account provides you with the credentials to access the Twitter API and therefore having this account is a necessity.

4   Programme Functionality

The programme accept user inputs such as start date of search, end date of search and number of tweets to return. The user also provides a list of hashtags to query. Using these query requests and the constraints, the programme extracts images associated with these hashtags. Any retweets, replies and non-image tweets were ignored during this search. For ease of understanding, any embedded URLs within a tweet were also ignored as they are out of the scope of this project. A CSV file is generated containing the following information;userHandle, date, time, tweet, imageName, image URL, tweet URL. The images are also downloaded in the same folder as of the script. An error file is also created to log any errors.

5 Programme Execution

5.1 Running the Programme

1. **Prerequisites**: 'Tweepy' library is an API wrapper which makes interface with the Twitter API easier. Launch 'Spyder' as explained earlier and navigate to the 'iPython Console' as shown in Figure 1. Enter the below command within this console to download 'tweepy'. Once installed, you'll have to restart 'Spyder'

   ```
   pip install tweepy
   ```

2. **Hastags:** The unzipped folder will contain 'hashtags.txt'. Enter the list of hashtags you wish to search and ensure that each query is separated by a ','. Only fill the first two rows of the text file. Row queries are OR(d) and then final query is formed by AND(ing). Example format:

   (a) Row 1: #WFH, #remotework (OR'd)

   (b) Row 2: setup, #workspace (OR'd)

   (c) Final Query = (#WFH OR #remotework) AND (setup OR #workspace)

3. **Credentials:** The unzipped folder will contain 'credentials.txt'. Enter the credentials from your Twitter Developer Account. Do not change any variable names, just replace the 'xx...'s with your credentials

4. **Run:** Once the environment is all setup, press 'Run file' as shown in Figure 1

5. **User Inputs:** The programme will require the following inputs:

   (a) Start search date: Past date. *Note: This cannot be more than 7 days from the current date due to Twitter policy*

   (b) End search date: Recent date

   (c) Number of pages to return per API call (PGR). Each page usually contains multiple tweets

   (d) Number of tweets to return per page (TPG)

   (e) Total tweets = $(PGR) \times (TPG)$

5.2 Output

1. A CSV file is generated called *Twitter_API_Result.csv*. Open this using Excel(Windows) or Numbers(Mac OS). The information contained has been explained earlier

2. An *Error_Log.txt* is created and overwritten each time the code is executed. This logs any errors