Sentiment Analysis Project

By Brian Mallari

Project Description

The aim of this project is to apply sentiment analysis on Tweets gathered with the Twitter API in order to answer the following question:

"Which next-gen gaming console should I acquire – the PlayStation 5 or the Xbox Series X?"

The Python script used for this project was based on an assignment for a class taught by Dr. Chirag Shah, a professor at Rutgers University, back when I was still a graduate student.

Technologies Used

- Microsoft Windows 10 Home
- Python 3.8.5, along with the following modules:
    - matplotlib.pyplot
    - pandas
    - numpy
- Microsoft Word 2016 (for tracking references and writing this report)
- Visual Studio Code (for initial script development)
- Command Prompt (for initial script testing)
- Spyder 4.1.5 (for subsequent code development) through the Anaconda Navigator of the Anaconda distribution
- Spyder 4.2.1 (for the remainder of code development, as well as the saving of plots) through the Anaconda Navigator of the Anaconda distribution
- Xfinity (for internet access)
- Paint 3D (for cutting and pasting tables and other outputs)

How to Run the Python Script

This project utilized the following files in the same folder/directory:

1. sentiment_analysis_with_twitter_data_collection.py
2. sentiment_analysis_with_twitter_data_analysis.py
3. my_own_auth.k

The first Python script was developed to collect Twitter data using the Twitter API, while the second Python script was developed to analyze that data. The decision to separate collection and analysis was so that data collection can occur once, while data analysis can occur iteratively. This helps to maintain consistency with the underlying data that are being analyzed. If a new set of data was collected each time a new analysis was attempted, then the underlying data would change between each iteration of

analysis, thus compromising the validity of the analysis. The .k file was used for authentication purposes when utilizing the Twitter API to gather Tweets.

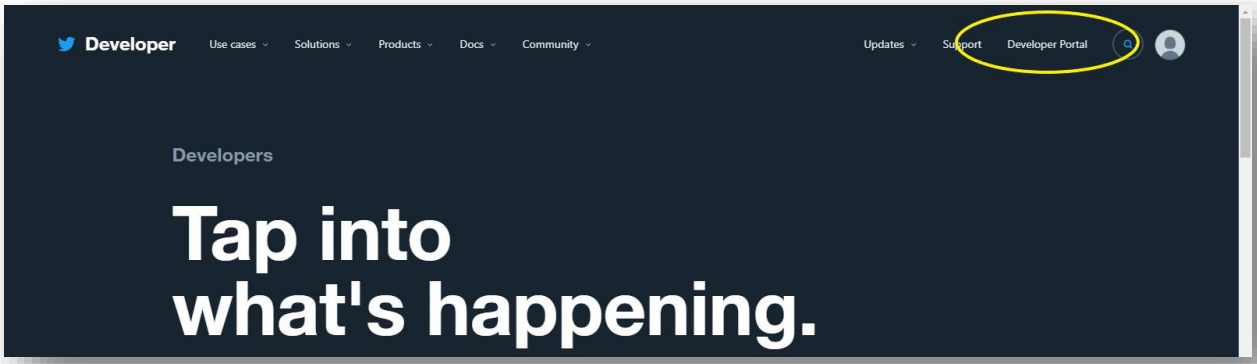
Data Collection

Before collecting Tweets from Twitter, a developer account needs to be created in order to acquire the necessary access codes for the Twitter API.

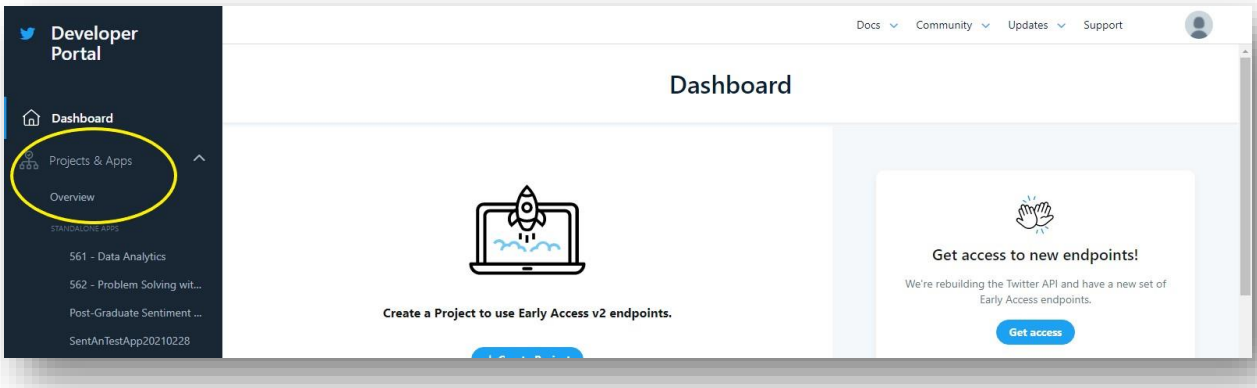
1. Make sure that you already have a Twitter account. If not, go to https://twitter.com/ and register for an account.
2. Go to https://developer.twitter.com/en and login with your Twitter account.
3. Follow the instructions on https://developer.twitter.com/en/docs/apps/overview to access the Twitter API.

Once you have your registered developer account, return to https://developer.twitter.com/en.

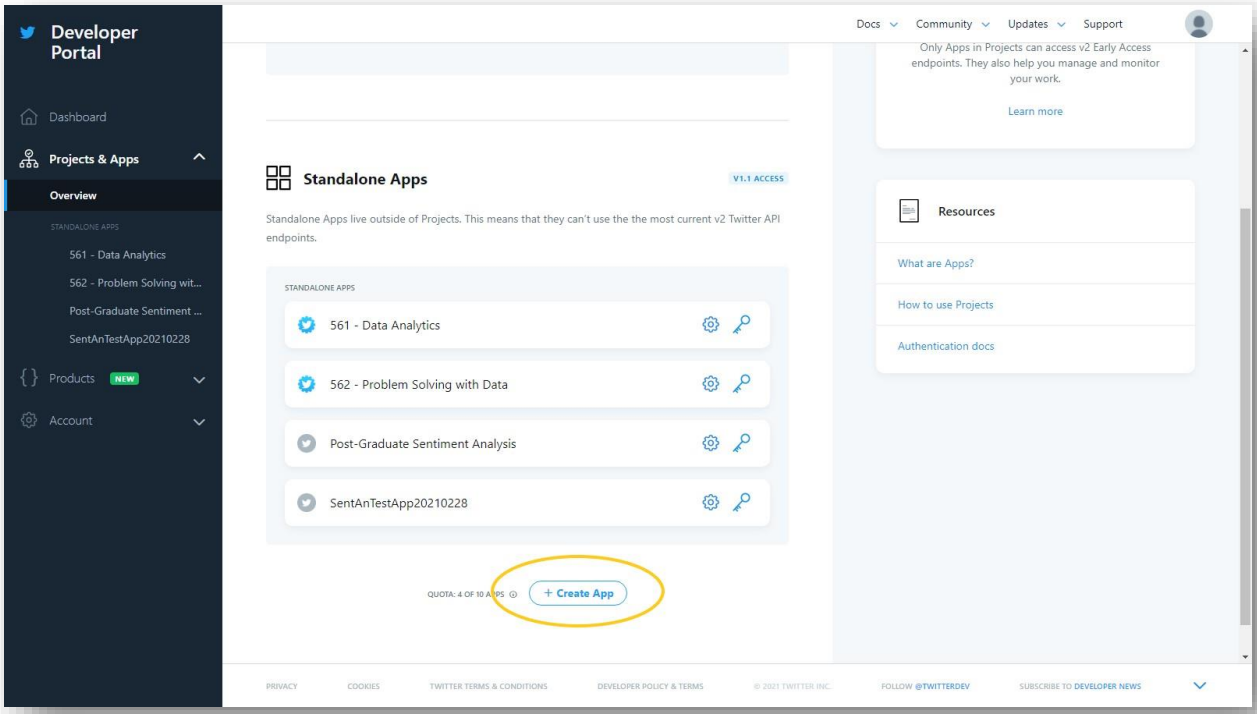
1. Make sure that you're logged in to your Twitter account.
2. Click on Developer Portal at the top-right section of the home page.



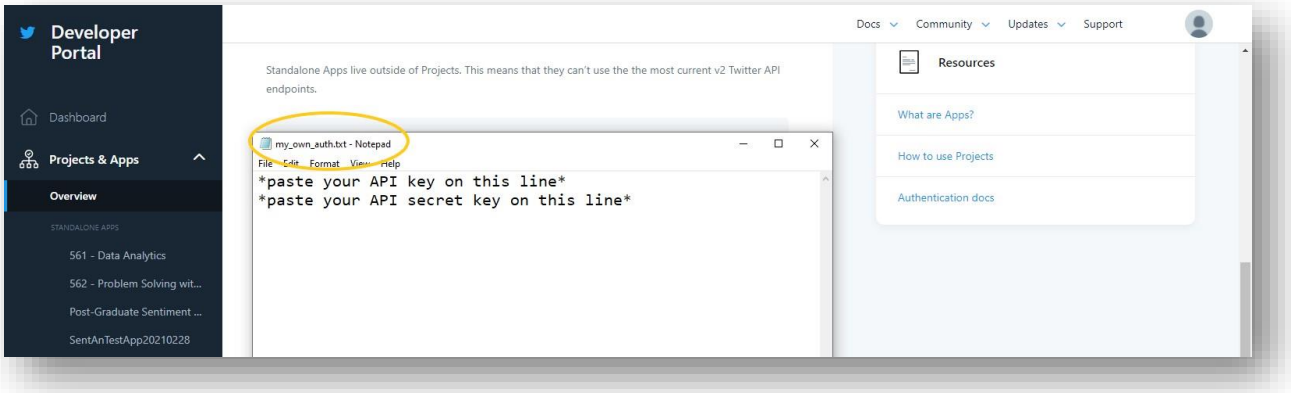
3. Go to the left navigation pane, click on Projects & Apps to expand a list, and then click Overview.



4. Go to the section on the main pane that says Standalone Apps and click on Create App.

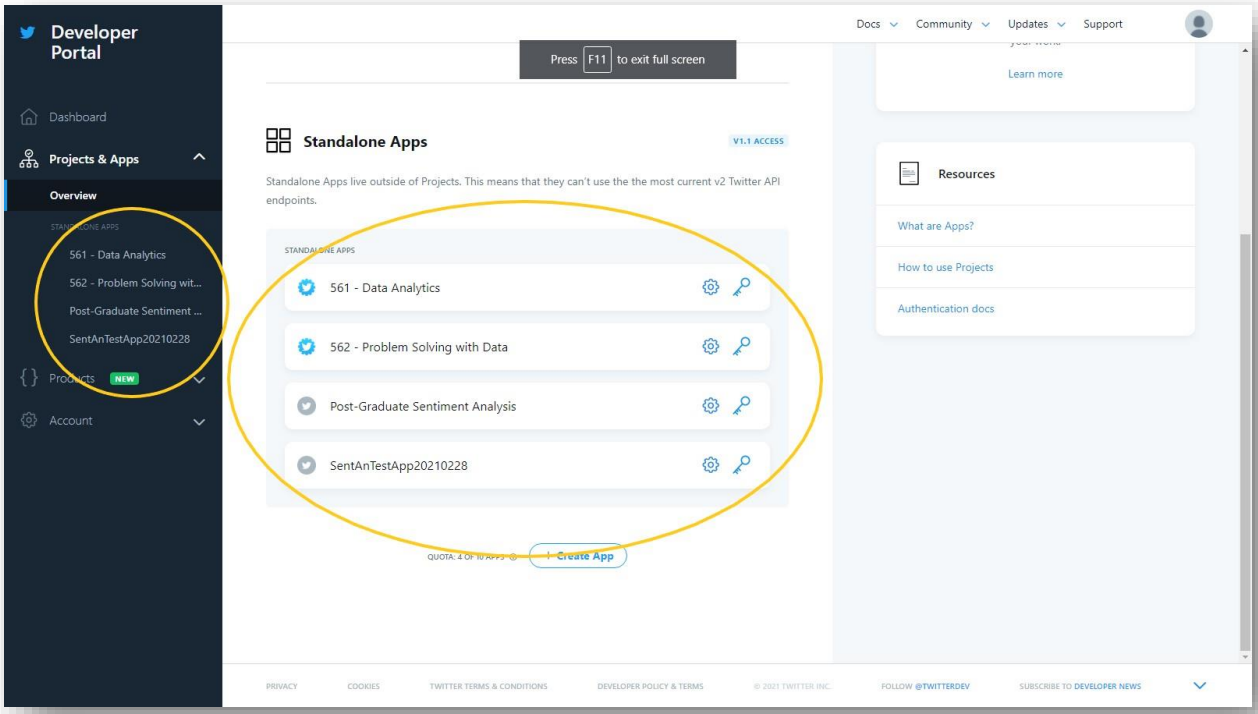


Follow the instructions for creating an app. At one point, both the API key and the API secret key will be presented to you. **Save each of them to a .txt file on separate lines, and name that file my_own_auth. You will need them later**.

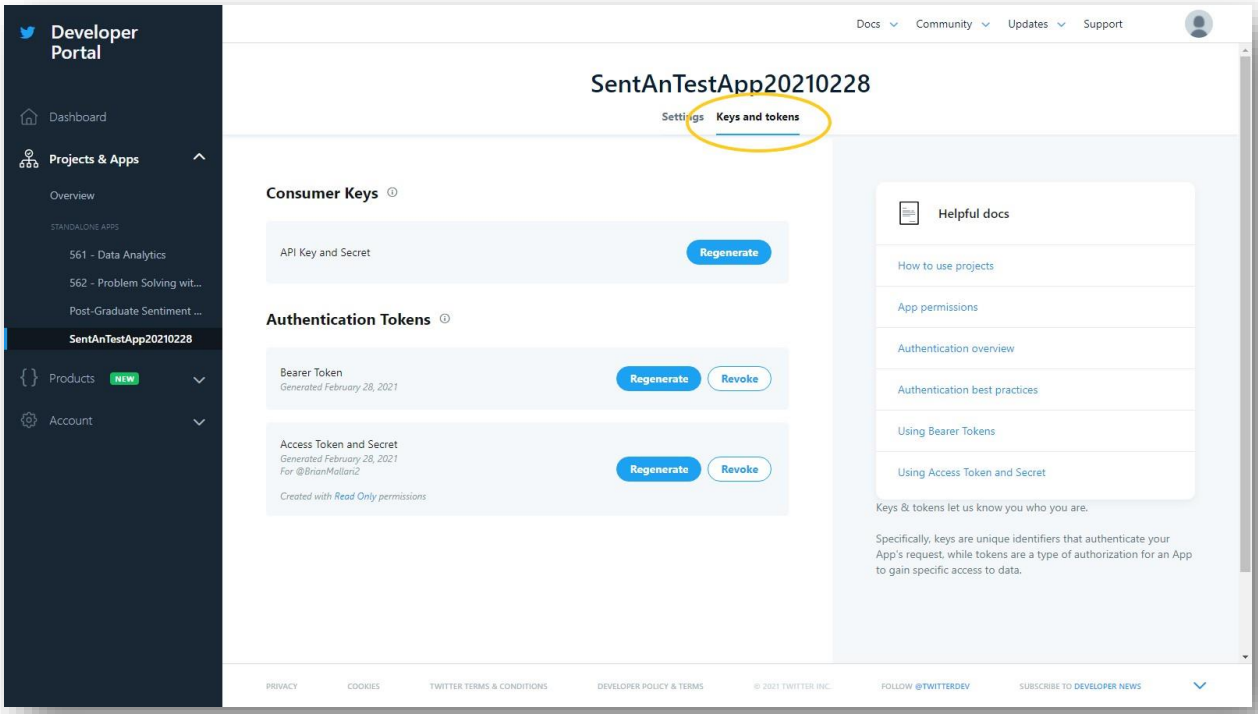


You don't have to make any modifications on the App Settings page. Read only permission for the app is fine for this project. Also, it's not necessary to set up 3-legged OAuth for this project, either. Returning to the dashboard should be sufficient.

Once your App has been created, you can find it listed underneath Standalone Apps both on the navigation pane at the left (under Project & Apps > Overview) or the Standalone Apps section in the main pane (on the Overview page).

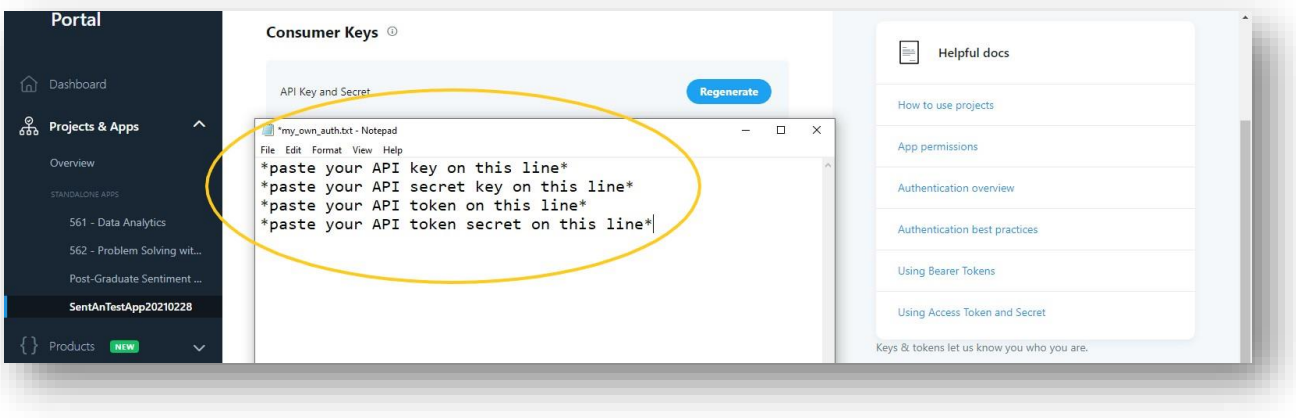


If you click on the app name in the navigation pane, you should be taken to the settings page for that app. Click on the “Keys and tokens” tab towards the top of the main pane (underneath the name of the app), and you should be taken to a page with sections titled Consumer Keys and Authentication tokens.



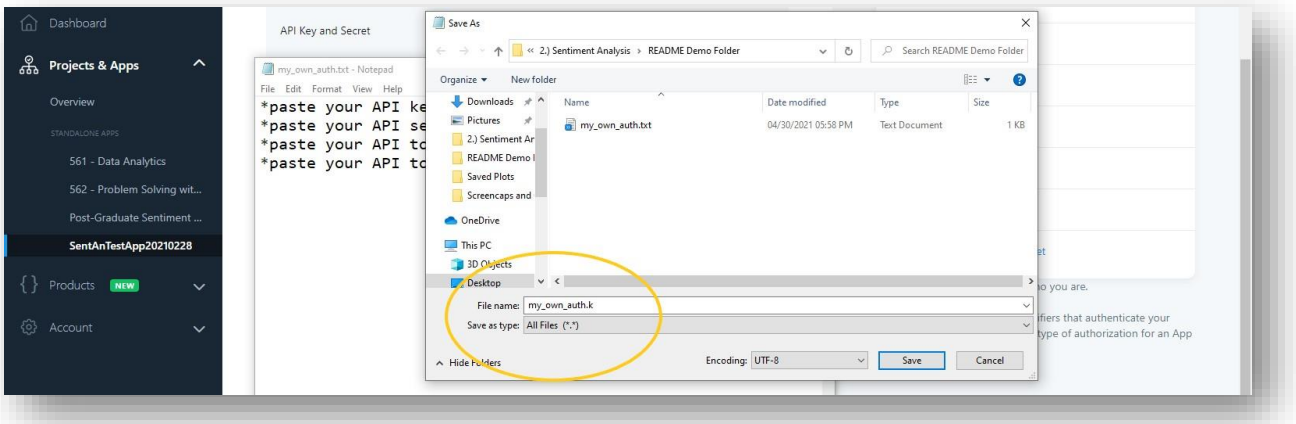
If you saved both the API key and the API secret key when the app was first created, then you can skip the Consumer Keys section and go straight to the section titled Authentication Tokens. If, however, those two pieces of information weren't saved, or if they got lost somehow, then you can gain new consumer keys by clicking on the Regenerate button under Consumer Keys, clicking on the "Yes, regenerate" button in the pop-up box, and then savings the new keys. If the old keys had been in use before, and new keys were generated, certain settings may have to be updated somewhere in order for certain scripts or applications to work properly.

Down in the Authentication Tokens section, go down to the line-item titled Access Token and Secret. If this is first time that these pieces of information are being generated, then click on the Generate button generate these values. A box should pop up with both the Access token and the Access token secret. **Save each of them on separate lines in the same .txt file used to save the API key and the API secret key. You will need these two values later as well**.



If either the Access token and the Access token secret weren't saved, or if either of them was lost, then you can return to Access Token and Secret line-item, click the Regenerate button, click the "Yes, regenerate" button, and save the new keys. Like in the case of the API key and the API secret keys, if the Access token and the Access token secret had been in use before, and new values had been generated, certain settings may have to be updated somewhere in order for certain scripts or applications to work properly.

Once the API key, the API secret key, the Access token, and the Access token secret have been collected in the my_own_auth.txt file, either change that file to a .k file by renaming the file extension from .txt to .k within the directory/folder holding that file, or by saving the file as my_own_auth.k using the "Save As…" function in your text editor. Make sure that "All files (*.*)" is selected for the field "Save as type:" in order for the file to be properly saved as a .k file.



After establishing the my_own_auth.k file, make sure that the Python script sentiment_analysis_with_twitter_data_collection.py is in the same folder/directory as the .k file. Next, run that Python script; it'll start collecting information on Twitter Tweets through the Twitter API using the four values saved in the my_own_auth.k file. Depending on certain factors such as internet connection or computer capabilities, two new .csv file should appear in the directory after a few minutes, one for Tweets related to the PlayStation 5 and one for Tweets related to the Xbox Series X.

The following data for each .csv file were collected using the Twitter API:

1. username – the name used by the Tweet author on Twitter
2. author id – a numerical ID associated with the Tweet author
3. created – the date the Tweet had been created
4. text – the content of the Tweet
5. retwc – the number of times the Tweet had been retweeted at the time of data collection
6. hashtag – any hashtag used in conjunction with the original Tweet
7. followers – the number of followers the Tweet author has at the time of data collection
8. friends – the number of friends the Tweet author has at the time of data collection

The following data for each .csv file were calculated based off of content in the text field:

1. polarity – a measure of how negative, neutral, or positive the original Tweet is; this value can range from -1 in the case of a negative Tweet to 1 in the case of a positive Tweet; a 0 indicates a neutral Tweet
2. subjectivity – a measure of the how objective or subjective the original Tweet is; a 0 indicates an objective Tweet, while a 1 indicates a subjective Tweet

The exact file name for each .csv file will depend on the date, time, and time zone that the script was run. This was to differentiate one set of collected data from another since each attempt to collect data is expected to produce different results given the ever-changing nature of what Tweets are available on the Twitter servers.

The script for analyzing data was first developed using Visual Studio Code with the remainder of the script being developed using Spyder, an integrated development environment for Python that came with the Anaconda distribution. Spyder was then used to run the final script for analysis.

The names of the files that were generated by the data collection script were copy-and-pasted directly into the data analysis script.

```
firstTwitterSearchResult = 'results_PlayStation 5_2021-01-06_13,12,31_EasternStandardTime.csv'
secondTwitterSearchResult = 'results_Xbox Series X_2021-01-06_13,12,34_EasternStandardTime.csv'
```

Plots generated when running the data analysis script have been saved as separate files, while tables generated by the data analysis script have been first saved by taking a screencap and then cropped down to isolate the desired table.