Dog Rates Data Wrangling

Introduction

The dataset that is chosen for wrangling (and analysing and visualizing) is the tweet archive of Twitter user @dog_rates, also known as WeRateDogs. WeRateDogs is a Twitter account that rates people's dogs with a humorous comment about the dog.

The goal is to wrangle WeRateDogs Twitter data to create interesting and trustworthy analyses and visualizations.

Data Wrangling

The data wrangling process consists of 3 main steps. They are:

1. **Gathering data** - The data can be gathered in many ways including web scraping, using APIs etc. The data can be gathered from a single source or from many different sources.
2. **Assessing data** - The data needs to assessed for quality and tidiness issues. This can be done visually and/or programmatically.
3. **Cleaning data** - Based on the assessment, the data is cleaned and tested to make sure all the issues identified are resolved.

Gathering Data

The data required for further analysis is collected in this step. The data may be collected from a single source or from multiple sources and combined later. The data could be provided or readily available, scraped from the web or collected using an API.

For this analysis, the required data is gathered from multiple sources. They are:

1. The WeRateDogs Twitter archive is enhanced and provided. This file (**twitter_archive_enhanced.csv**) is just downloaded.
2. The tweet image predictions, i.e., what breed of dog (or another object, animal, etc.) is present in each tweet according to a neural network. This file (**image_predictions.tsv**) is hosted on Udacity's servers and is downloaded programmatically (using requests).
3. Additional required and interesting data is obtained by querying the Twitter API (using tweepy) for each tweet's JSON data and store each tweet's entire set of JSON data in a file (**tweet_json.txt**).

Downloading Files Programmatically

The requests library is used to download the file giving the URL where the file is hosted as input. The response from the URL is written to a TSV file opened in binary write mode [Figure 1].

Further, the file is read into a dataframe using the read_csv() of pandas specifying '\t' (escape sequence for tab as the separator) [Figure 2].

```
In [2]: # storing the URL provided in a variable
        url = 'https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-predictions/image-predictions.tsv'

        # getting the response from the URL using requests library
        response = requests.get(url)

        # with keyword ensures that the file is closed immediately the desired operation is complete
        # file is opened for writing in binary mode
        with open('image_predictions.tsv', 'wb') as file:
            # content of the response is written to the file
            file.write(response.content)
```

Figure 1: Downloading file programmatically

```
In [3]: image_predictions_df = pd.read_csv('image_predictions.tsv', sep='\t', index_col=None)
        image_predictions_df.head()
```

Out[3]:

	tweet_id	jpg_url	img_num	p1	p1_conf	p1_dog	p2	p2_conf	p2_dc
0	666020888022790149	https://pbs.twimg.com/media/CT4udn0WwAA0aMy.jpg	1	Welsh_springer_spaniel	0.465074	True	collie	0.156665	Tru
1	666029285002620928	https://pbs.twimg.com/media/CT42GRgUYAA5iDo.jpg	1	redbone	0.506826	True	miniature_pinscher	0.074192	Tru
2	666033412701032449	https://pbs.twimg.com/media/CT4521TWwAEvMyu.jpg	1	German_shepherd	0.596461	True	malinois	0.138584	Tru
3	666044226329800704	https://pbs.twimg.com/media/CT5Dr8HUEAA-lEu.jpg	1	Rhodesian_ridgeback	0.408143	True	redbone	0.360687	Tru
4	666049248165822465	https://pbs.twimg.com/media/CT5IQmsXIAAKY4A.jpg	1	miniature_pinscher	0.560311	True	Rottweiler	0.243682	Tru

Figure 2: Creating dataframe from file

Creating the Twitter Developer Account

The following are the steps to create a Twitter Developer account in order to query the Twitter API for additional information:

1. Create an account/Log in to Twitter
2. Navigate to the "How To Apply" section in this page with link - https://developer.twitter.com/en/docs/developer-portal/overview and click on the "Apply for a developer account" button [Figure 3]

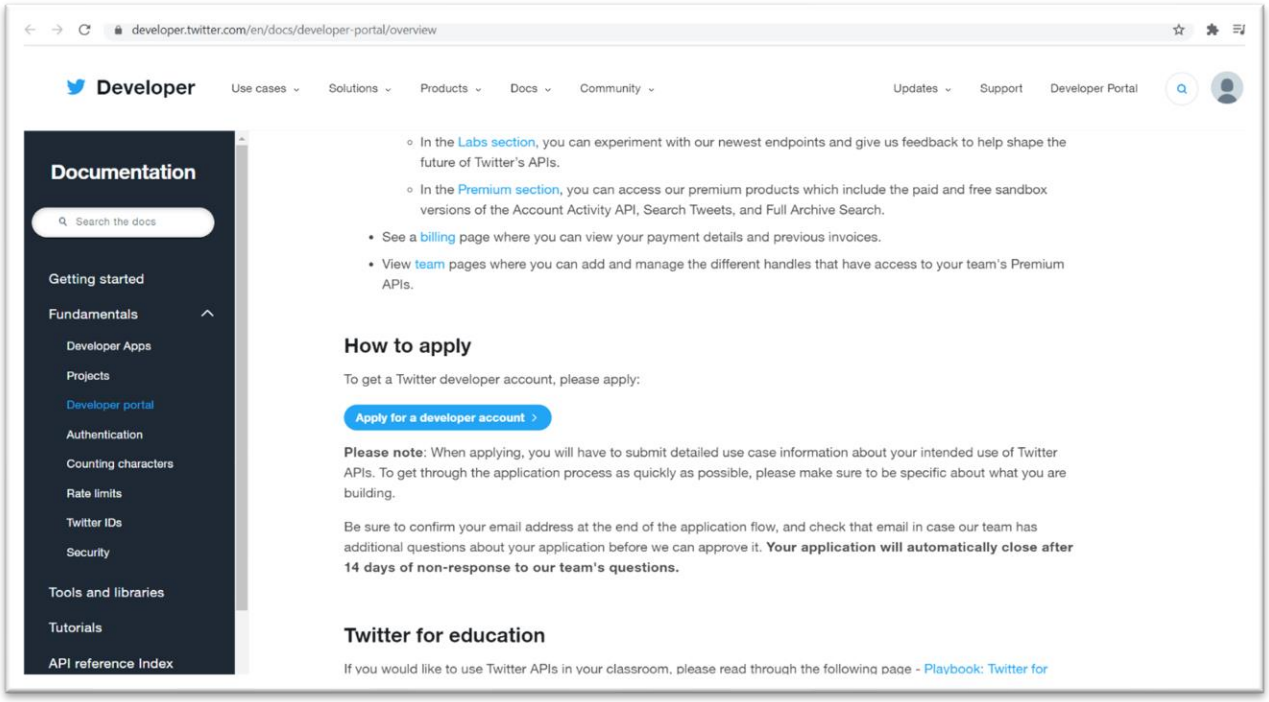

Figure 3: Home Page of Twitter Developer Portal

3. Provide the details required for the developer account creation
4. Once the details are submitted, you will receive a mail from Twitter confirming that the account has been activated [Figure 4]

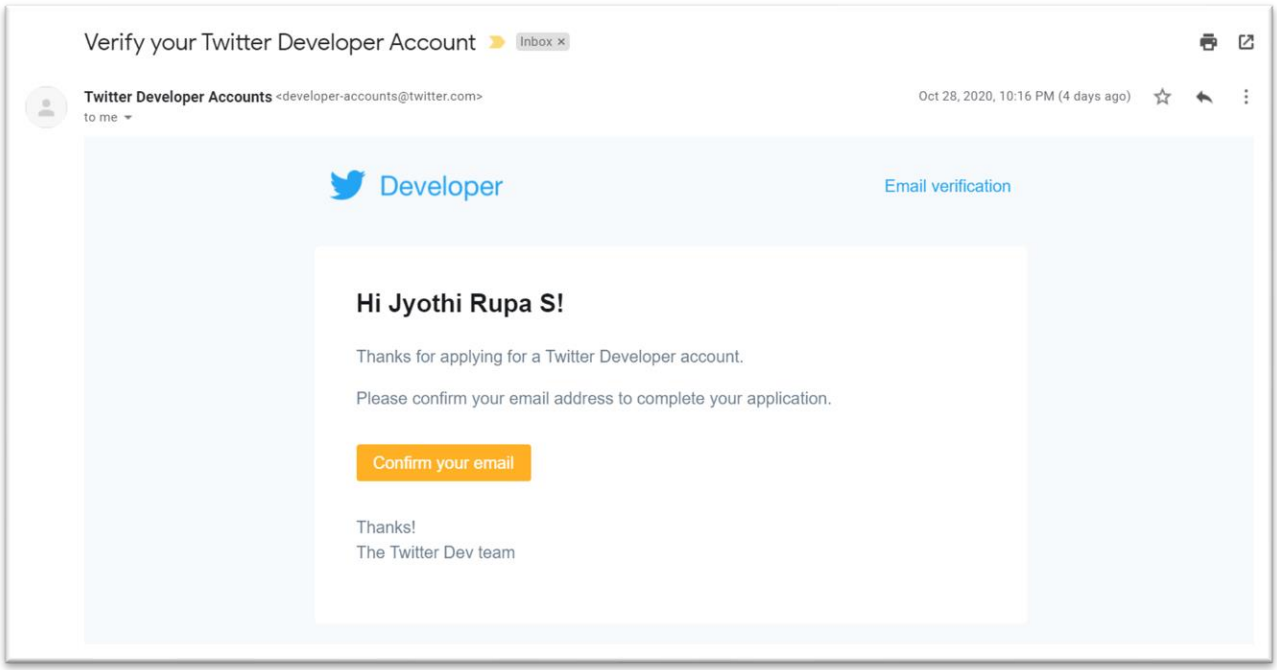

Figure 4: Twitter Developer Account Activation Confirmation Mail

5. Go to the Keys and Tokens tab on the page you are redirected to from the mail to find or generate the Consumer API keys, and the Access Token and Access Token Secret that you will need [Figure 5]

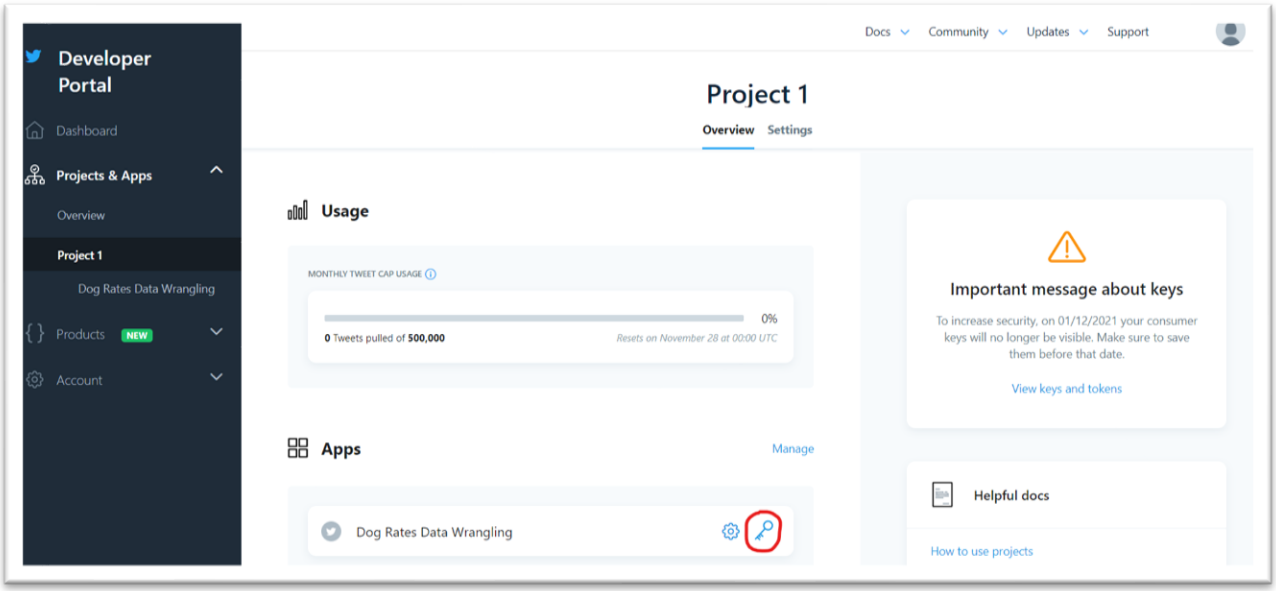

Figure 5: Project Home Page

Assessing Data

There are 2 types of issues that need to be assessed. They are:

1. **Quality issues** - Issues with content. Low quality data is also known as dirty data.
2. **Tidiness issues** - Issues with structure that prevent easy analysis. Untidy data is also known as messy data. The requirements of tidy data are:
    - Each variable forms a column
    - Each observation forms a row
    - Each type of observational unit forms a table

These issues can be assessed in 2 ways. They are:

1. **Visual assessment** - Scrolling through the data in your preferred software application
2. **Programmatic assessment** - Using code to view specific portions and summaries of the data

For the visual assessment, 5 each of the first, last and random records are considered. The records are retrieved using the pandas functions .head(), .tail() and .sample(5) respectively.

For the programmatic assessment, various exploratory functions such as .info() which gives details of missing values & data types of the fields, .describe() which gives the quartile information and other statistics of numerical fields and .duplicated() which gives us information about whether or not a record is a duplicate are used.

In this analysis, the issues identified during analysis fall under one or more of the following categories:

- Missing data
- Incorrect data types for fields
- Data does not follow the guidelines for tidy data
- Fields have few or many incorrect values
- The values of a field are inconsistent in representation (formatting required)
- Removing unwanted records/fields

Cleaning Data

There are 2 types of cleaning. They are:

1. **Manual** cleaning (not recommended unless the issues are one-off occurrences)
2. **Programmatic** cleaning

The programmatic data cleaning process includes 3 steps. They are:

1. **Define**: Our assessments are converted into defined cleaning tasks. These definitions also serve as an instruction list so others (or we ourself in the future) can look at your work and reproduce it.
2. **Code**: Converting those definitions to code and run that code.
3. **Test**: Testing the dataset, visually or with code, to make sure your cleaning operations worked.

The data cleaning is performed on copies of the original data.

The order in which the various issues are resolved depends on the impact it has on the other issues. For example, if the resolution to a missing data issue is to remove the fields with high number of missing values, and another issue is the data type of the values in of this field, it is logical to consider the missing data issue first.

Generally, first the missing data issues are resolved followed by the tidiness issues and finally the quality issues. The quality issues could be resolved first in case data is to be combined in order to bring about tidiness and there is a large amount of data making the analysis process tedious after combined.

The cleaning is performed programmatically only. Popular python data analysis libraries such as pandas and numpy are used for cleaning the data.

The final cleaned data is saved into a file called **twitter_archive_master.csv**.