Reporting: wragle_report

- Create a **300-600 word written report** called "wrangle_report.pdf" or "wrangle_report.html" that briefly describes your wrangling efforts. This is to be framed as an internal document.

In this Data Wrangling project, three pieces of data are required to be gathered using three different approaches which. Below are the approaches used and how I was able to wrangle the data.

1. **WeRateDogs Twitter archive data (twitter_archive_enhanced.csv):** this data was provided in the project lessons and was downloaded manually.
2. **Image Predictions File**: this is the second dataset required for analysis and was downloaded programmatically using the link provided on the project lesson, I use the **get()** in the requests library to download the tweet image prediction and with open function, I created an **image_predictions.tsv** file to hold the downloaded contents.
3. **Additional data**: I would have love to get the additional data by querying the Twitter API but was unable to get approval for twitter developer account which resorted me to download the **tweet_json.txt** provided in the project lesson. The downloaded **tweet_json.txt** was read line by line and **tweet_id, retweet_count**, and **favorite_count** were extracted and append to a dictionary list which were later converted to a dataframe.

Assessing Data

The datasets were both visually using **head(), tail(), sample()** and programmatically assessed using **shape, nunique(), unique(), duplicated(), info(), sum(), isnull(),** etc. to detect quality issue and tidiness issue.

Cleaning Data

After assessment, I copied the dataframe to another dataframe variables so as to get access to the original data if there is corruption of data in the course of cleaning.

I followed the cleaning procedure of Define, Code and Test and below are the cleaning done on the datasets.

1. I dropped rows that has retweeted tweet and then reset index using drop() and reset_index()
2. in_replys, status_timestamp, retweeted_status and expanded_urls columns were dropped using drop()
3. I changed **ids** datatype from int to string in the three datasets using astype() function
4. I changed timestamp dataype from string to datetime using to_datetime()
5. Both rating datatype were changed from integer to float
6. I rreplaced all lower case letters in twitter archive name column with "None"
7. I converted all items in p1,p2,and p3 to lower case for consistency using lower() function
8. I filtered out and removed all rows with False observations in p1_dog column
9. I dropped all columns having p2 and p3 using drop() function
10. None value in doggo, floofe, puppo, and pupper column were replaced with empty string "". the columns were added and assigned to newly declared variable dog_stage.

11. Joined named as a result of addition done in previous step were renamed by including comma to separate for readability and empty string were replaced with np.NaN
12. doggo, puppo, pupper and floofer column were dropped using drop() function
13. The three separate dataframes were combined using merge(), index were dropped and reset, also **id** column were dropped to avoid repetition of column.

Data Storage

The merged dataframe were saved **to twitter_archive_master.csv**, dropping row index.