WRANGLE REPORT

Goal :

wrangle WeRateDogs Twitter data to create interesting and trustworthy analyses and visualizations. The Twitter archive is great, but it only contains very basic tweet information. Additional gathering, then assessing and cleaning is required for "*Wow!*"-worthy analyses and visualizations.

Tasks :

- Data wrangling, which consists of:
    - Gathering data (downloadable file in the Resources tab in the left most panel of your classroom and linked in step 1 below).
    - Assessing data
    - Cleaning data

Gathering data

This project involved gathering of data from three different sources as listed below. For each of the data source a different method of data gathering was used namely:-

1. Importing data via csv
2. Using requests to download data off internet
3. Scrape data from an API

4. **Twitter archive file:** download this file manually by clicking the following link: twitter_archive_enhanced.csv

5. **The tweet image predictions:** i.e., what breed of dog (or other 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 should be downloaded programmatically using the Requests library and the following URL: https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-predictions/image-predictions.tsv

6. **Twitter API & JSON:** Each tweet's retweet count and favorite ("like") count at minimum, and any additional data you find interesting. Using the tweet IDs in the WeRateDogs Twitter archive, query the Twitter API for each tweet's JSON data using Python's Tweepy library and store each tweet's entire set of JSON data in a file called tweet_json.txt file. Each tweet's JSON data should be written to its own line. Then read this .txt file line by line into a pandas DataFrame with (at minimum) tweet ID, retweet count, and favorite count.

In the end, I ended up with three datasets :

- twi_archive - contains data read from provided csv
- image_predictions - contains data read (by using requests) from tsv file hosted on server
- tweet_json - contains data obtained from twitter handle by using tweepy library and creating a twitter app for oauthshape

Assessing Data

After gathering each of the above pieces of data, I assessed them visually and programmatically for quality and tidiness issues. I detected and documented **eleven (11) quality issues** and **three (3) tidiness issues** in my `wrangle_act.ipynb` Jupyter Notebook.

Quality Issues

twi_archive dataset:

- **181 retweets to be dropped.**
- **NaN values named with other names like NONE in status columns.**
- **timestamp type object. (to be changed to datetime)**
- **change source column values to these basic values : [twitter, vine, tweetdeck, iphone].**
- **convert columns doggo, floofer, pupper, puppo values into true/false.**
- **validate and correct rating numerator and denominator.**
- **Convert wrong names into nan values.**
- **Delete columns that won't be used for analysis. (in_reply_to_status_id, in_reply_to_user_id, retweeted_status_id, retweeted_status_user_id, retweeted_status_timestamp)**

image_prediction dataset:

- **drop duplicated jpg_urls**
- **extract dog breed from columns p#, p#_conf and p#_dog.**
- **Delete columns that won't be used for analysis. (img_num, jpg_url)**

tweet_json dataset:

- **tweet_id type object. (to be changed to int64)**

Tidiness Issues

- **group age stages (doggo, floofer, pupper, puppo) into one column named "stage"**
- **create dog bread column.**
- **the three datasets need to be merged into one.**

Cleaning Data

1- the steps in the data cleaning process were respected and clearly documented (Define , Code , Test).

2- Copies of the original pieces of data are made prior to cleaning.

3- All issues identified in the assess phase are successfully cleaned (if possible) using Python and pandas, and include the cleaning tasks required

4- A tidy master dataset with all pieces of gathered data is created.