Wrangle Report

Introduction:

The WeRateDogs Twitter archive contains basic tweet data for all 5000+ of their tweets, but not everything. One column the archive does contain though: each tweet's text, which Udacity used to extract rating, dog name, and dog "stage" (i.e. doggo, floofer, pupper, and puppo) to make this Twitter archive "enhanced." Of the 5000+ tweets, Udacity has filtered for tweets with ratings only (there are 2356).

Udacity extracted this data programmatically, but We didn't do a very good job. The ratings probably aren't all correct. Same goes for the dog names and probably dog stages (see below for more information on these) too. You'll need to assess and clean these columns if you want to use them for analysis and visualization.

Our 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.

Project Steps:

These are the three task of this project:

- Gathering data
- Accessing data
- Cleaning data

Gathering data

The data in this project mainly consist of three datasets. These are

- **Twitter_archive file:** This file is provide by Udacity
- **Images_predictions:** 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

- **Twitter_json & API:** 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.

Accessing Data

Once all these three datasets are gathered we need access these three dataset

- Although i did used two tool one excel to see the structure and juypter notebook to apply programming language to filter and manage dataset.
- I have access almost all column though there order in dataset presence and i have used many methods to see the structure more carefully(i.e. Info, value_counts, sample, groupby)

Through access to the datasets I have encountered many quality issues and tidiness issues like:

- there are missing value in following columns : 'in_reply_to_status_id', 'in_reply_to_user_id', 'retweeted_status_id','retweeted_status_user_id', 'retweeted_status_timestamp','expanded_urls'
- name with 745 missing value as None(not Nan)
- tweet_id is int64 type as it should be object type as no calucation is needed
- timestamp and retweet_status_timestamp is also object type
- having retweets might bring duplicates.
- max rating_numerator is 1776

For quality issues and tidiness

- Twitter_Archive, Images_Predictions, Tweet_json : Join all three dataset into one main dataset
- Twitter_Archive : contains four column(dogger, floofer, pupper and puppo) needs to make it into one
- Twitter_Archive : All Prediction columns must be packed into one single columns

Cleaning Data

This part of the task is mainly a composite of define, code and test.

First of all we have created a copy of all these dataset and then merge all these three dataset.

We melts four columns of data to one. And In the prediction columns like p1, p2 and p3 we will melt all data set to one predictiom.

Conclusion

- Use twitter API to gather data.(tweepy_of_twitter)
- Good understanding of big data is must
- Dealing with unstructured data like json.
- Handling, accessing and cleaning and visualization using code.