WeRateDogs Twitter:

Wrangling dogs and tweets

A step by step guide to the process used in preparing the data for the report 'act_report'

Gather process:

The file that was provided *twitter_archive_enhanced.csv* was directly downloaded and placed into the website. This was then loaded in as df1

An *image_predictions .tsv* file was obtained using the requests library and taken from the cloudfront host. This was loaded as df2

Finally a *tweet_json.txt* file was gathered using code in PyCharm and the file was then uploaded to the Jupyter workspace. This text was created using research of my own and not the script provided. This was loaded in as df3

Assessment:

Starting with visual assessment of the JSON file, we identify key structures we want to extract from this file.

- “id”
- “retweet_count”
- “favorite_count”

These were then extracted from the JSON data and put into a dataframe, I had considered putting other columns but realized they were not useful for any analysis as we are pulling data from a single source.

Quality issues:

- Large number of NaN's in multiple columns
  - 'in_reply_to_status_id',
  - 'in_reply_to_user_id',
  - Retweet_status_id
  - Retweet_status_user_id
  - Retweet_status_timestamp
  - ‘expanded_urls’
- timestamps are not in datetime data type
- duplicate dog names may or may not be same dog
- Mistyped data types for four categoricals ‘doggo’, etc.
- All rating denonimators should be 10, but we have at least one greater than 10, different denominators need to be normalized.
- Multiple ratings in the 100’s including one at 1776 will likely make rating comparisons difficult or useless.
- Tweet_id in df1 and df2 should be string not int
- Img_num should be string not int
- Df2 contains image guesses that are not dogs.
- Retweet_count and favorite_count should be int not object
- Different numbers of rows and columns in all 3 datasets

- Dogs with names like ‘a’, ‘an’, etc.

Tidiness

- No data for categorical options doggo, floofer, pupper, puppo
  - These 4 categories should be a single column
- df3 contains multiple observational units
  - retweet and favorite are both observational units

Cleaning process:

A copy of df1 called df1_c was created, this will be the copy that all cleaning operations are worked on.

There were 3 columns to be removed. All 3 are retweet status related. Since retweets are not being considered in this analysis, they were purged. To do that the first thing was to locate all the retweets, and drop the rows containing them. A ‘purge’ dataframe , labelled df1_p, was created so that if the need should be found for this information for some other operation later, it can be quickly restored by commenting or removal of out one line of code.

After that the next step was correcting the datatypes, in order starting with timestamp, then altered all columns that need to be strings and aren’t into the appropriate datatype. At this time df2’s appropriate column data type changes were made.

To alter category to “none”, “doggo”, etc. first verification that no one dog has more than one category assigned to it was needed. After verifying this was the case, a loop through each category and assignment to a new column of ‘dog_type’ was carried out. The data in was then altered to a category data type. The original columns were then dropped.

Normalization of the denominators, with the assumption that all should be a “10” was the next step. In order to carry this out it was neccessary to properly adjust the numerators in these categories. This was done by multiplying both the numerator and denominator by 10 and dividing by the original denominator, These results were placed into new columns called rating_numerator_norm, and rating_denominator_norm then the original columns were dropped. This resulted in a single row that had a denominator of ‘0’ initially failing. This row was removed from the dataset for calculation purposes as it isn’t really a valid measure, and there is no way to properly fix it.

Next df3_c was split into two separate dataframes, one for retweet_count and one for favorite_count. Given that these are both observations, they should be on separate tables. These dataframes are df3_c_r and df3_c_f respectively.

Because of the dataframes having different sizes, and the fact that no meaningful analysis can be done without matching datasets, the four dataframes were trimmed to all have the same number of rows, 1987 in total. It was then verified that each of the dataframes had the same exact selection of 'tweet_id' numbers so as to ensure valid comparisons could be made.

External References:

https://www.earthdatascience.org/courses/use-data-open-source-python/intro-to-apis/twitter-data-in-python/

http://docs.tweepy.org/en/latest/code_snippet.html

https://medium.com/@I_am_milica/the-beginners-guide-to-downloading-twitter-data-using-tweepy-4ec981eaba77

https://stackoverflow.com/questions/18869688/twitter-api-check-if-a-tweet-is-a-retweet

https://realpython.com/pandas-settingwithcopywarning/