Wrangle Report –

For this analysis, I wrangled three different datasets. The first was a dataset I downloaded using the requests library from https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-predictions/image-predictions.tsv. This was the dataset which included the breed predictions from the images on the weratedogs tweets.

The second dataset used was provided in the project materials and was read into a dataframe from the csv file. This was the twitter archive dataset.

2. The third dataset used was a dataset created from api requests from the Twitter API developer portal. To create this dataset, I created a twitter developer account and used the Tweepy library to make api "df_twitter_archive" columns "doggo", "floofer", "pupper", and "puppo" have "None" listed for all values. These columns hold no value and should just be dropped. (Tidiness issue because columns should store actual/useful variables and these columns store nothing).

requests of all the tweet ids from the other two dataframes.

Due to the request limit from the twitter api, I made a function which paused the requests for fifteen minutes every 900 requests.

I saved these requests in a json format in a .txt file, then created another function which read the file line by line and parsed the json format to extract only the tweet ids, favorite counts, and retweet counts for the dataframe.

I used both visual and programmatic assessment to see what needed to be cleaned in the datasets. I found the following issues and addressed each of them:

1. In "df_twitter_archive", tweet ids were listed in descending order (other two dataframes are listed in tweet id ascending order). I fixed this by changing the order to ascending to match the other two dataframes.

2. In "df_image_predictions", breed names were not uniform in their casing (some start with uppercase, some lowercase). I used the str.lower() method to change all of the casing to lowercase to be uniform.

3. NaN values in "df_twitter_archive" dataframe. I ended up dropping several columns including “in_reply_to_status_id”, “in_reply_to_user_id”, “retweeted_status_id”, “retweeted_status_user_id”, and “retweeted_status_timestamp.” These columns were ultimately dropped because the vast majority of the rows were NaN, so they were not useful for further analysis across the datasets.

4. Removed ids missing from "api_tweets_df" from "df_image_predictions." I trimmed this dataset down to match the tweet_ids in the dataset requested from the twitter api.

5. Removed ids missing from "api_tweets_df" from "df_twitter_archive." I trimmed this dataset down to match the tweet_ids in the dataset requested from the twitter api.

6. Column names in "df_image_predictions" were not descriptive. I used the project motivation documentation to find the meanings of the column headers (such as p1, p1_conf, etc) and renamed the columns to be more descriptive so they were understandable to the end user.

7. In "df_twitter_archive", "Timestamp" column was just a string/object, not a datetime type. I used the pandas to_datetime method to convert this into a datetime format instead of just a string.

8. In "df_twitter_archive" "name" column - several values were obviously not names. I manually looked through the unique values in the name column and removed any values which were obviously not names (such as “a”, “an”, “unacceptable”, etc). I replaced these values in the name column with the string “NOT FOUND,” as the name was unable to be parsed correctly.

Tidiness issues:

1. In “df_twitter_archive,” I separated the datetime object into two separate columns – date and time.

2. In "df_twitter_archive" columns "doggo", "floofer", "pupper", and "puppo" had "None" listed for all values. These columns hold no value and were dropped completely. (Tidiness issue because columns should store actual/useful variables and these columns store nothing).

3. I separated the df_twitter_archive into the following datasets – “tweet_times” with the id, posted date, and posted time, “tweet_ratings” with the id, rating numerator, and rating denominator, “tweet_text” with the id, text, and dog name, and “tweet_urls” with the id, source, and expanded urls.

I also created a cleaned master dataframe and saved it to a separate csv file, twitter_archive_master.csv

This concluded my wrangling and cleaning efforts for these datasets.