Wrangling WeRateDogs' Twitter Data

My first step was to gather the data. Using pandas' read_csv function, I downloaded 'image-predictions.tsv' and 'twitter-archive-enhanced.csv' and converted them to dataframes. Then using the tweepy library, I copied all the json data in 'twitter-archive-enhanced.csv' to a text file called 'tweet_json.txt'. I then extracted the 'tweet_id', 'retweet_count' and 'favorite_count' from each tweet in the 'tweet_json.txt' and put them in a new dataframe called 'tweet_json_data'.

My next step was to assess the data, of which there were three dataframes of which to assess. 'twitter_archive', 'tweet_json_data' and 'image_predictions'. My assessment of the three dataframes revealed 13 quality issues and 4 tidiness issues.

Quality

The first quality issue was that the 'timestamp' and 'retweeted_status_timestamp' columns from 'twitter_archive' needed to be converted to a date type. The second through fifth issues were that 'img_num' column and id columns in 'image_predictions', all id columns in twitter_archive and all id columns in tweet_json needed to be converted to a string. The sixth issue was that the 'retweet_count' and 'favorite_count' columns from 'tweet_json_data' needed to be converted to integers.

The seventh issue was to fix names in the 'name' column of twitter_archive that are not names (Ex: 'a', 'the', 'an', etc.) by either finding the true name in the 'text' column or replacing the non-name value with a NaN value. For the eighth quality issue, I replaced all 'None' values in twitter_archive_clean with a NaN value.

The ninth issue found was that twitter_archive's 'rating_numerator' and 'rating_denominator' columns sometimes had the wrong data stripped from the 'text' column. So I redid both columns from scratch. Using a regular expression, I would extract any and all fractions from the 'text' column. If there was only one fraction, I put the corresponding values in the 'rating_numerator' and 'rating_denominator' columns. If there were two fractions (there were never more than two), I put the second fraction (my assessments showed me that the second one is always the correct one) in the 'rating_numerator' and 'rating_denominator' columns.

The tenth issue was that all the ratings seem to use a denominator of 10, but there are sometimes multiple dogs rated and thereby the rating is multiplied by the number of dogs. So I reverted all denominators to 10 and the numerator to its corresponding value. I then created a 'multiplier' column that would show the number of dogs rated. (Ex: a rating_numerator of 20 and a rating_denominator of 20 rating would be changed to rating_numerator of 10, a rating_denominator of 10 and a multiplier of 2.)

The eleventh issue was to melt doggo, floofer, pupper, and puppo columns into one dog_stage column. The twelfth issue was to replace all values that have a string 'nan' with a true NaN value. The thirteenth and final issue was to remove all rows that has data in the retweeted_status_id, retweeted_status_user_id, and retweeted_status_timestamp columns since they are not real tweets but retweets, which we do not want

Tidiness

I found four tidiness issues. The first tidiness issue was to merge all data in the tweet_json_data and image_predictions_clean dataframes into twitter_archive_clean's dataframe. The second issue was that since twitter_archive's 'rating_denominator' column is always 10 (I checked. After doing fixing the eighth quality issue, there are only 10s and 3 NaNs), then that column is not needed. The third issue was to remove the 'doggo', 'floofer', 'pupper' and 'puppo' columns. Since the 'dog_stage' column has all the same data, the 'doggo', 'floofer', 'pupper' and 'puppo' columns are redundant. The fourth and last issue was to remove the 'retweeted_status_id', 'retweeted_status_user_id' and 'retweeted_status_timestamp' columns since they no longer have any data.