WeRateDogs Data Wrangling

Gathering Data

The project depends on 3 data sets.

1 - Enhanced Twitter Archive
    Solution: Downloaded Directly as a .csv file and loaded to df_archive dataframe
2 - Image Prediction File
    Solution: Downloaded programmatically as a .tsv file and loaded to df_images dataframe
3 - Twitter API
    Solution: Depending on the twitter_ids in the df_archive dataframe, twitter API was consumed and each API was saved as a line in a .txt file and needed data was loaded to df_api dataframe.

Assessing and Cleaning

During assessing and after getting familiar with the 3 data sets through visual assessment these issues appear in the programmatic assessment. Issues are listed in the same order they were solved in.

1 - df_archive and df_api were found to be the same observational unit
    Solution: merge the 2 dataframes on ‘tweet_id’ using pd.merge()

2 - in df_archive, some rows are retweets as columns 'retweeted_status_id' and 'retweeted_status_user_id' has values
    Solution: drop the non nan rows in retweeted_status_id and retweeted_status_user_id using pd.Series.isna() function

3 - in df_archive, some rows are replies as columns 'in_reply_to_status_id' and 'in_reply_to_user_id' has values
    Solution: drop the non nan rows in in_reply_to_status_id' and in_reply_to_user_id using pd.Series.isna() function

4 - some tweets do not have images as shown in the difference in rows number between df_archive and df_images
    Solution: drop rows in df_archive that don't have a match in tweet_id column in df_images using the pd.Series.isin() function

5 - after removing rows that doesn't have a match in tweet_id from df_archive, number of rows becomes less than number of rows in df_images

Solution: drop rows in df_images that don't have a match in tweet_id column in df_archive.

6 - in df_archive, missing values in expanded_urls

Solution: Solved while solving other completeness issues

7 - in df_archive 'doggo', 'floofer', 'pupper', 'puppo' are just 1 variable 'stage'

Solution:

- replace the None values with empty values to be neglected while using pd.series.add() function
- create a new series 'stage' which has all the values in doggo, floofer, pupper and puppo
- fix typo issues using the replace function
- drop unwanted series

8 - in df_archive, in columns(name,...,puppo), missing data are represented as 'None' not 'NaN'

Solution: replace None with NaN using df.series.replace function in name as other columns are fixed

9 - in df_archive, min value in 'rating_denominator' can not be 0

Solution: replace the rating_denominator of 0 to be 10

10 - in df_archive, timestamp is not datetime

Solution: change the timestamp to datetime using pd.to_datetime() function

11 - *in df_archive, rating_numerator values need to be float and the decimal rating is not properly extracted from the text.*

Solution: change the datatype of the column from int to float and then extract the values from the text column in df_archive_copy that have decimal numerator and then update the numerator column with these values

12 - in df_archive, invalid names like 'a' or 'an'. probably happens because name value was extracted after the "this is .." in the tweet. A common pattern found for such names is they all start with small letters.

Solution: check in the names series if the first letter is lower-case then replace the value with NaN using the pd.series.mask() function

13 - in df_images, false values for p1_dog and p2_dog

Solution: drop false results in p1_dog then p2_dog(to add more assertion) in df_images and merge df_images with df_archive to create df_master to be used in visualization