Wrangling Report

In this project, I used a dataset from a Twitter account known as “WeRateDogs,” which was provided by Udacity.

Gathering Data:

Data from @WeRateDogs was gathering using three different methods.

1. **Twitter archive**: the twitter archive data was a .csv file, which was downloaded manually. This file contained the bulk of data used for this project.
2. **Image predictions**: This was a .tsv file which was programmatically downloaded, and contained images and predictions for dog breeds.
3. **Twitter API / JSON**: additional data was gathered from twitter using the alternate using the alternate method provided, and using the JSON package.

Assessing:

After gather the data, I assessed it using the following methods:

1. **Visual assessment**, by looking through the three data frames.
2. **Programmatic assessment**, by using python and pandas to get a clearer picture of the data.

After assessing, issues found were then separated into “quality” and “tidiness” categories to be cleaned. I found 8 quality, and 2 tidiness issues in the datasets.

Cleaning:

Below are the issues I found, and the methods I used to correct them:

Quality #1: The “tweet_id” column in a three data frames should be string data types.

I changed the datatypes of the non-sting “tweet_id” columns to string, so that they would be in a correct user ID format.

Quality #2: Remove retweets.

I removed retweets by deleting where the” retweeted_status_id” was not null.

Quality #3: Remove all unnecessary columns from “df_archive.”

I dropped columns that were not needed for analysis, to clean up the data frame.

Quality #4: The timestamp datatype should be in the correct format.

I changed the data type to datetime, so that it would be in the accurate format.

Quality #5: Change “rating_numerator” and “rating_denominator” data types, to allow decimals.

I changed the data types to float, so that decimal ratings could be included in the future.

Quality #6: Change the naming convention of p1, p2, and p3 to be all lowercase, for consistency.

I changed the dog breed name columns to only have lowercase letters, for consistency, since there was originally a mix of both uppercase and lowercase.

Quality #7: Change dog names labeled as "a" and "None" in the name column, to "NaN."

I changed all “a” dog names to “None,” and then replaced all “None” values with null, for consistency.

Quality #8: Change the multiple columns for dog breeds, predictions, and confidence levels into just 2 new columns for dog breed and prediction confidence.

To get the first correct prediction for dog breed and confidence, I first made a function to get those answers for each dog. I then made those answers into 2 new columns (“dog_breed” and “pred_confidence”), and deleted the original columns since they were now unnecessary.

Tidiness #1: Dog stages (doggo, floofer, pupper, and puppo) should be combined into a single column.

I used pd.melt to combine the 4 columns together into a “dog_stage” column, which made the dataset more concise.

Tidiness #2: Merge dataframes into one master dataset.

I used pd.merge to combine the 3 data sets into a single master data set, which I could then save.

First, copies of the datasets were made, and then cleaning was done using the define-code-test framework on the copies. These steps were done for each issue. Some of the more challenging issues that I encountered were combining the dog stage columns in the dataset, and also making two new columns through creating a function to get dog breed and confidence level.

Conclusion:

After I cleaned the datasets, by solving the 10 issues that were outlines previously, I saved the data to a new master data frame, and then to a .csv file with UTF-8 encoding.

Cleaning the data issues turned out to be more challenging for me than I had predicted, since I was learning to better use python and pandas, and spent a lot of time figuring out how to solve issues and make my code work correctly, which is a very important lesson.