PROJECT 2 – DATA WRANGLING REPORT

Submitted by Glory Edamkue

The process of wrangling the data was something new to me when it came to the analysis, however, I followed the usual procedure to import dependencies from the beginning of the code. I imported all the libraries I would need—some of them were Pandas, NumPy, datetime, Seaborn and matplotlib. For the Twitter queries I would be making, I got tweepy, json and getpass.

For the importation of the documents, I made use of:

- the pd.read_csv ( ) method to import the twitter_archive_enhanced.tsv from the classroom
- the request library to query a website for a particular file which we were to get from the class, I got it and named it as told, image_predictions.tsv
- my Twitter API keys with the Twitter library Tweepy to query twitter data for each tweet and write/dump its JSON data to a text file called tweets_json.txt. The data being queried had the exact tweet ids as those of the twitter_archive_enhanced.csv in order to stick to the same subset of the total tweets available from WeRateDogs. Once that was done, I had to retrieve that JSON data for all the queried tweets and use them to make a DataFrame by reading it line by line in my notebook.

With that successfully done, I began the next step of wrangling, which is assessing. I first assessed the 3 files under consideration visually. I could see null and empty values, weird dog names of just one letter, and see how the schema for each file was. Then I assessed the 3 files programmatically. I made use of the .head ( ) , .tail ( ) , .describe ( ) , .sample ( ) and other code that helped me to see what work needed to be done.

I detected and corrected 9 quality issues in the dataset, and 2 tidiness issues. I began with the first quality issue, which was the need to have only original tweets. I noticed from sampling programmatically, that in the “text” column all entries that were replies started with “@” while all retweets started with “RT @”. I used regex to isolate those from the enhanced DataFrame. But some replies did not follow that categorization, so I had to redefine and re-code in order to get that cleaned up. I worked on the datatypes of the columns, and even changed the name of the one called “timestamp”, so that I could work more easily with the data if need be. I merged all 3 DataFrames into 1. Then I removed all the null values I felt would not be needed in the analysis, and adjusted the numerators to a more realistic range of ratings less than 20 to prevent outliers that would change the statistical variance of the DataFrame; I made all denominators 10 so that the weighting would be even. I made the four dog age columns into one to tidy up the DataFrame, too.

Satisfied with the cleanliness and quality of the DataFrame, I saved it as twitter_archivie_master.csv and then analyzed it.

For the analysis, I made use of grouping and Matplotlib to derive my major analyses. Some of the graphs were subplots in order to show relationships between two variables, such as the retweets column and the favorite column. For the predictions, I used percentages to show their variance much more easily.

My conclusions have limitations, due to the incomplete nature of the data. However, I went ahead with them, because the more data available, the better the end result accuracy for the ones that were complete.