WRANGLE REPORT

This document briefly describes the wrangling efforts in the **wrangle_act** notebook.

Data Gathering:

The three pieces of data required for this project required a different gathering method:

- The WeRateDogs Twitter archive data (***twitter-archive-enhanced.csv***) was downloaded directly from the Udacity classroom and read into the **df_twitterarchive** dataframe using the pandas library.

- The tweet image predictions data (***image_predictions.tsv***) was downloaded programmatically from Udacity's servers using the requests library and the get() method. It was subsequently read into the **df_imagepred** dataframe using the pandas library.

- The Twitter API code was provided by a Udacity instructor in the file - (***tweet_json.txt***) due to mobile verification issues I encountered. The json library was used to read the tweets into a .txt file which was read line by line to append the tweet_id, favorite_count, and retweet_count into the **df_tweet** dataframe.

Assessing Data:

The data was visually and programmatically assessed for quality and tidiness issues. The methods used to assess the data were: head(), sample(), info(), value_counts(), describe(), duplicated (). The duplicated() method showed the **tweet_id** column to be duplicated in the dataframes under review.

Upon assessment of the 3 dataframes, I discovered nine quality issues and three tidiness issues.

The **df_twitterarchive** dataframe contained unnecessary columns, a wrongly named column - “floofer” and it had missing values in some of its columns. It contained retweets and replies which were meant to be removed. Its **tweet_id** and **timestamp columns** were of the wrong datatype. There were a few outliers in its numerator and denominator ratings which were most likely inaccurate.The column names **'rating_numerator'** and **'rating_denominator'** needed to be shortened.

The types of dogs in columns p1, p2, and p3 of the **df_imagepred** dataframe were not presented in a uniform manner, some of them were in uppercase while others were in lowercase letters. Its **tweet_id** column was also of the wrong datatype. A lot of the top predictions in the p1 column were other animals (not dogs) or random items. The columns for the stages of dog (doggo, pupper, puppo, floof) needed to be one category column.

In the **df_tweet** dataframe, the column **id_str** needed to be renamed to **tweet_id** in order for all the tables to be merged.

Cleaning Data:

All the issues documented during the assessment of the data were cleaned using the Define, Code and Test method. In defining the issues, I stated the ways they would be resolved through coding. In the Code segment, I implemented the resolution already defined and I tested out the results in the Test segment of this section of the project.

In cleaning the data, I used a couple of functions and methods:

- The dropna( ) and drop( ) methods were used to remove unwanted columns as well as those with missing values. The join( ) method was used to merge the dataframes into one.
- The to_datetime( ) and astype( ) functions were used to convert columns to the appropriate data types. The rename( ) function was used to rename columns.
- Arithmetic operators were also used to filter the archived data (**archive_copy**) and predicted data (**predict_copy**). The iloc( ) function was used to replace wrong ratings in **archive_copy** after wrangling the dataframe using a combination of regex functions and the string( ), extractall( ) & query( ) methods.
- The str.lower( ) method was used to convert all the names in p1, p2, and p3 to lowercase letters while the replace( ) method was used to replace empty strings in the archived data (**archive_copy**)

I programmatically assessed the cleaned data to ensure that the solutions were adequately implemented. After the data had been gathered, assessed, and cleaned, the master dataset was saved to a CSV file named "twitter_archive_master.csv".