Data Wrangling Report

1. Data Gathering

Here, I gathered 3 pieces of data and loaded them in the `wrangle_act.ipynb` notebook.

- For the WeRateDogs Twitter archive data: I directly downloaded the CSV file `twitter_archive_enhanced.csv` from Udacity, then read it in a dataframe called `df_archive`.
- For the tweet image prediction data: I used the `request` library to programmatically download the TSV file `image_predictions.tsv` from the URL provided by Udacity, then read it in a dataframe called `df_image`.
- For the retweet and favorite counts: I intended to use the Twitter API to read each tweet's JSON data line by line in a txt file `tweet-json.txt`, then used the `json` library to create a corresponding dataframe. However, due to the conflict between API v1.1 and v2 in my account, I had no choice but to download the provided file from Udacity and read it in a dataframe called `df_tweet`.

2. Data Assessment

I used both **visual assessment** and **programmatic assessment** to assess 3 dataframes:

- `df_archive`: the WeRateDogs Twitter archive data (twitter_archive_enhanced.csv)
- `df_image`: the tweet image prediction (image_predictions.tsv)
- `df_tweet`: the Twitter API (tweet_json.txt)

In terms of **Quality**, besides some duplicates and irrelevant data due to retweets, most complicated issues come from the extraction of names, ratings, and dog stages from the `text` column in `df_archive`. In `df_image`, I noticed that some images are not about dogs. There are also other issues related to data types and inconsistent values (`None` vs. `NaN`, `timestamp` instead of `string`, inconsistent capitalization and underscore, etc.) that had to be fixed.

In terms of **Tidiness**, most issues originate from the redundant retweet rows and columns, unnecessary columns for the analysis like `expanded_urls`, `in_reply_` in `df_archive`. Moreover, 4 dog stage columns in `df_archive` all refer to 1 variable, so they should be merged into one column. Similarly, instead of spreading 3 predictions over 9 columns in `df_image`, we only need 4: `prediction_num`, `dog_breed`, `confidence`, and `accuracy`.

3. Data Cleaning

In this part, I created 3 copy dataframes: `archive_clean`, `image_clean` and `tweet_clean`.

My cleaning flow began with the removal of all redundant retweets in 3 dataframes and the join of `archive_clean` and `tweet_clean` (issue #1), deleted records that are not about dogs (issues #2), and turned 9 columns in `image_clean` to just 4 (issues #3). After that, I used the Quality and Tideness in part 2 to crosscheck and fix other issues.

The most challenging part was to extract the correct `rating_numerator` and `rating_denominator` from the `text` column in the `archive_clean` dataframe (issue #10). For the analysis purpose, I also created another `rating` column in `archive_clean`, found out, and handled 3 outliers.

4. Data Storage

To end my data wrangling process, I saved each clean dataset to a CSV file: `twitter_archive_master.csv` and `image_predictions_master.csv`