Data Wrangling Report

1. Gathering Data

About the Dataset(s)

The dataset I will be wrangling is the tweet archive of Twitter user @dog_rates (https://twitter.com/dog_rates), also known as WeRateDogs. WeRateDogs is a Twitter account that rates people's dogs with a humorous comment about the dog.

Based on the images in the above dataset (i.e., WeRateDogs Twitter archive), another dataset is created which consists of image predictions (the top three only) alongside each tweet ID, image URL, and the image number that corresponded to the most confident prediction (numbered 1 to 4 since tweets can have up to four images). Though no wrangling will be done directly on this image predictions dataset, it will provide some additional data for our main tweet archive dataset.

- **Gather Twitter archive CSV file.**
  Using Udacity sources in classroom, I downloaded the WeRateDogs Twitter archive manually as twitter-archive-enhanced.csv and imported this file into a dataframe (twitter_archive).

- **Gather tweet image predictions.**
  I downloaded the tweet image predictions file hosted on Udacity's servers programmatically using Python's Requests library and saved it locally to image-predictions.tsv file. Then, I imported this file into a Python Pandas dataframe (image_prediction).

- **Gather data from Twitter API.**
  (I used tweer-json.txt in Udacity classroom because I did not have access for twitter API)
  Using the tweet IDs in the Twitter archive, I accessed the entire data for every tweet from Twitter API and stored every tweet's entire set of JSON data in a file called tweet-json.txt file. Then Created a dataframe tweet_df from this JSON including only tweet_id, retweet_count, and favorite_count.

****

2. Assessing Data

Visual Assessment:

I opened the twitter-archive-enhanced.csv and image-predictions.tsv in Excel and scrolled through them, looking for quality and tidiness issues. I was able to spot the following 2 quality and 2 tidiness issues.

- **Quality issues**:

  1- unnecessary html tags in source column of twitter archive in place of utility name e.g.

  2- Rename the columns with descriptive names in image_prediction dataframe.

- **Tidiness**:

  1- doggo, floof, pupper and puppo columns in twitter_archive table should be merged into one column named "dog_stage".

  2- breed column should be added in twitter_archive table and put its values based on confidence columns and prediction is a breed of dog columns of img_prediction table.

Programmatic Assessment:

- I used pandas' info method on twitter_archive to spot erroneous datatypes and other quality issues, if any.
- I used value_counts method on rating_numerator, rating_denominator.
- during the visual assessment, I realized the twitter_archive dataframe its tweets has more than one dog-stage mentioned.

This entire activity helped me to identify the following 7 quality issues:

- **In twitter_archive dataframe:**

  - timestamp data type should change to datetime data type.

  - Data type issues in_reply_to_status_id, in_reply_to_user_id

  - rating_denominator has values less than 10 and more than 10 (like 1776)

  - some records have more than one dog stage

  - many tweet_id(s) of twitter_archive table are missing in img_prediction table.

  - As twitter_arcive contains retweets, there was duplicates.

- **in image_prediction dataframe:**

  - Remove duplicates in jpg_url column.

And 3 tidiness issues:

- doggo, floof, pupper and puppo columns should be merged into one column named " dog_stage."

- retweet_count and favorite_count columns from tweet_df table should be joined with twitter_archive table.

- tweet_archived dataframe without any duplicate (realized during cleaning), so I realized that arc_clean table have empty columns (retweeted_status_id, retweeted_status_user_id and retweeted_status_timestamp) which can be dropped

3. Cleaning Data

As all the quality and tidiness issues were to be fixed I create a copy of all these tables and named them as following (arc_clean, img_clean and tweet_clean). For each quality/tidiness issue, I performed the programmatic data cleaning process in 3 stages - Define, Code & Test. During the cleaning process, although convert the datatypes of source and dog_ stage columns of arc_clean to category datatype.

4.Storing Data

After the completion of the cleaning process, I stored the arc_clean DataFrame in twitter_archive_master.csv file.