Data Wrangling Report

1. Data Gathering

The datasets used in this project were from the tweet archive of Twitter user @dog_rates, also known as WeRateDogs. The datasets are extracted and saved as `twitter_archive_enhanced.csv`, `image_predictions.tsv` and `tweet_json.txt`.

The `twitter_archive_enhanced.csv` was given, so I downloaded it manually and created a dataframe table and called `twitter_archive`. While the `image_predictions.tsv` file was programmatically downloaded using `request.get` method. I use conditional and created a folder with the same name `image_prediction` to avoid repeating the whole process of fetching the data every time I restarted the notebook and I wrote the content as `wb` in an image_prediction.tsv file and also created a dataframe called image_prediction from it.

For the `tweet_json.txt`, I used `tweepy` library and tweet ids from twitter_archive_enhanced.csv file to query Twitter API. When extracting the json data; I use conditional and `os.path.exists` to avoid repetition, a for loop to get the json data from the API and a `try and except` function to check for success and error message and saved the data as tweet_json.txt. Out of the JSON data, I extracted retweet_count and favorite_count and converted it into a dictionary and created a dataframe table called it api_df.

2. Assessing the data

I began assessing the data, by looking at the twitter_archive.csv and image_prediction.csv datasets using google sheet and then programatically to check for quality and tidiness issues.

For the twitter_archive data; I discover that the name column contain names like 'a', 'an', 'very' and others that start with lowercase letter which is a quality issue and having four stages 'doggo','floofer','puppo' and 'pupper' which is a tidiness issue. I also checked those issues programmatically.

Programmatically, I checked the twitter_archive table for missing data, and wrong data type using the .info() method. timestamp column had an object data type instead of datetime and expanded_urls had 59 missing data.

I used the .describe() method on the twitter_archive data and I was able to discover some quality issues with rating_numerator and rating_denominator columns. The rating_denominator had values other than 10 which is not expected and further assessment of the rating_numerator and rating_denominator in comparison to the text column in which the values were gotten, it was discovered some rating values on the text are different from the values of the rating_numerator.

Checking the dog stages by joining two, three and four of te stages together, it was discovered that some records(rows) has more than one dog stage.

Manually assessing the image_prediction table, it reveals that the tweet_ids were arranged differently. So programmatically, I wrote a function `dataframe_difference` that compares the tweet_ids in twitter_archive table to the tweet_ids image_prediction table by merging the two tables. It was discovered that all the tweet ids in image_prediction table were in twitter_archive table but not all tweet ids in twitter_archive table were in image_prediction table. The reason is because image_prediction table had 2075 records.

The issues I found were as follows:

Quality issues

twitter_archive

`1.` Data type for rating_numerator should be float since some of the values from the text column are float

`2.` The rating_numerator column has a value other than 10.

`3.` The rating_denominator column have values other than 10.

`4.` Wrong data type for timestamp column.

`5.` Invalid names with lower case such as 'a','the','by' etc in the name column

`6.` The expanded_url has some missing data.

`7.` Not all records in twitter_archive are in image_prediction

`8.` Some rows have more than one dog stage(doggo and pupper)

Tidiness issues

twitter_archive

`1.` Unnecessary columns for dog stage(floofer, pupper, doggo and puppo)

`2.` Since we are only interested in original ratings(no retweets) that have image, retweeted_status_id, retweeted_status_user_id, retweeted_status_timestamp are unnecessary

api_df

- The retweet_count and favorite_count columns should be merge with twitter_archive dataframe

Cleaning the data

The cleaning process took care of the highlisted issues in the assessment

For the quality issues, these are the steps I took in cleaning the data:

- The rating_numerator data type was changed to float using .astype() method because comparing the values to those in the text column; some were in float and rating_denominator values other than 10 were removed using drop method.
- timestamp column datatype was also change to datetime using astype method, while invalid names in the name column were removed. Because of the missing data in the expanded_urls column; those tweet ids not in expanded_urls were removed. Records that had more than one dog stage were also cleaned.

For the tidiness issues; I combine the dog stages into one column and called it stages and drop other ones. I also remove those columns that are not needed such retweeted_status_id, retweeted_status_user_id, retweeted_status_timestamp, in_reply_to_status_id and in_reply_to_user_id from the archive_clean table and finally merge the api_df table to twitter_archive table using the tweet_id column as the key.

The twitter_archive is then saved as twitter_archive_master.csv using .to_csv method

References:

- (Comparing Rows Between Two Pandas DataFrames)[https://hackersandslackers.com/compare-rows-pandas-dataframes/)
- (Reading and Writing JSON to a File in Python)[https://stackabuse.com/reading-and-writing-json-to-a-file-in-python/]
- (Tweepy: a Python Library for the Twitter API - Jason Rigden - Medium)[https://medium.com/@jasonrigden/tweept-a-python-library-for-the-twitter-api-9d0537dcebd4]
- (Downloading Files using Python (Simple Examples) - Like Geeks)[https://likegeeks.com/downloading-files-using-python/]

In [ ]:

```

```

In [ ]:

```

```