Wrangle Report

Data Gathering

In this project, there are three datasets originally from different sources:
1. CSV File
This file is downloaded from the provided link in this project. The content of the file is then be imported to pandas dataframe named 'twitter_archive' by using pandas.read_csv function.
2. TSV File
This file is downloaded from the provided link in this project programmatically using python request module. The content of the file is then be imported to pandas dataframe named 'image_predictions' by using pandas.read_csv function.
3. Twitter API
This file is downloaded by tweepy package in python and is written to a text file line by line and then is saved. Then, the content of this file is imported to pandas dataframe named 'tweet_data' by using json.loads.

Data Assessing

The assessing process includes two facets: checking the quality and checking the tidiness.
The following problems are found in the process.

Quality
twitter_archive table

- Some entries in the column of rating_denominator have value not equal to 10
- Weird numbers in the column of rating_numerator(9.75 to 75, 99/90 to 99, ...)
- There are invalid values in column of name (such, an, ...)
- There are retweets data in the table
- Erroneous datatypes(retweeted_status_id, retweeted_status_user_id, in_reply_to_status_id, in_reply_to_status_user_id, timestamp, retweeted_status_timestamp, id)
- Some rows contain two stages of the dog
- There are some data from other website of which the expand_url is not www.twitter.com

image_prediction table

- Erroneous datatypes(id)

Tidiness

- One variable in four columns in twitter_archive table (dog stage is one variable)
- Tweet_data table cannot be a seperate table

Data Cleaning

Quality:

On 'twitter_archive' table:

1. Setting all entries in the column of rating_denominator equal to 10

2. Re-extracting the rating score from the text using regex, and re-calculating the score with denominator 10. Then, re-writing the new score to the column.
3. Re-extracting the dog name from the text column using regex, and re-writing to the column of dog names.
4. Subsetting the table by removing the rows in which the entry of the column of 'retweeted_status_id' is not NaN
5. Changing the datatype by using astype function
6. Finding rows which contain two stages of the dog and exploring the text of the row to revise the problem manually
7. Removing rows of which the source doesn't contain ‘twitter.com’

On ‘image_predictions’ table:

1. Changing the datatype using astype

Tidiness:

1. Dividing the dataset into two sub-dataset. The one contains the observation of which the dog stage is 'None', the other one contains the observation of which there is a dog stage. For the one does not have a stage, deleting the last four columns and adding a column named 'dog_stage' of which entries are 'None'. For the one has a stage, converting the four columns to one by using function melt(). Finally, concatenating the two dataset and sort the concatenated dataset by tweet_id.
2. Extracting 'favorite_count' and 'retweet_count' from the downloaded json data and adding them as column to twitter_archive table.