I. Data Gathering

In this project, I gathered data from three sources. First, I downloaded the “twitter-archive-enhanced.csv” file manually, and uploaded it to the Jupyter workspace. I then used the read_csv function of pandas to access the data. Second, in terms of the image prediction, I downloaded the tsv file programmatically. Due to its file format, I then changed the separation mark from colon to tab. In terms of the favorite and retweet counts, I couldn’t obtain the approval from Twitter API, and therefore I used the code provided from the “Twitter API” section. I named these three dataframes df_1, df_2, and df_3, respectively. Upon finishing gathering data, I combined all three dataframes into one. I set up the index as “tweet_id” for all three dataframes, and merged them by index, which returned df_5.

II. Data Cleaning and Accessing

After gathering all the data, I started to play around with the dataframe by looking into the datatypes, null values, outliers, incorrect values, etc. After that, I identified 8 issues with data quality and 4 issues with data integrity.

For data quality:

1. Two rows (2344 and 2355) have missing values in “favorite_count” and “retweet_count”.

Identified the two rows where these two columns have missing value, and then deleted them.

2. Incorrect datatype: “favorite_count” and “retweet_count” should be integers.

Converted both columns into integers by using the “.astype(int64)” command.

3. Incorrect datatype: “timestamp” should be datatime.

Converted it into datetime format by using “pd.to_datetime” command.

4. Incorrect datatype: “img_num” should be integer.

First, filled the missing values in this column with 0, and then converted it into integers.

5. “rating_denominator” should all be 10.

I first replaced all the values in this column as 10, and then converted the datatype into integers.

6. Incorrect values in the “rating_numerator” column.

Extracted ratings from the text, then filled the rows without ratings with 0, and lastly converted the column into integers.

7. Except for 394 tweets, the others' “dog stages” are not identified.

Extracted dog stages (“doggo”, “floofer”, “pupper”, “puppo”) from text, and then compared the number of dog stages extracted with the number of dog stages provided from original dataset. Replaced the dog stages accordingly.

8. Incorrect names.

Since names are very difficult to extract, I deleted the ones whose names are obviously incorrect, such as “the”, “a”, “None”.

For data tidiness:

1. “doggo”, “floofer”, “pupper”, and “puppo” should combined into one column “stages”.

For these four columns, fill the null values with “”, and then add these four columns together in a new column “stage”. After combining them, there are some entries with multiple dog stages, separate the stages with “,”. Lastly, deleted the four columns representing dog stages.

2. Redundant information:
    a. “in_reply_to_status_id” and “in_reply_to_user_id” both indicate whether the tweet has replies.
    b. "retweeted_status_id” and “retweeted_status_user_id” both indicate whether it was retweeted.

Created two new columns, “retweeted” and “replied”, to record whether these tweets were retweeted and replied, respectively. Then filled these two columns by identifying whether there were entries in the “in_reply_to_status_id”, “in_reply_to_user_id”, "retweeted_status_id” and “retweeted_status_user_id” columns. After filling the new columns, deleted these four original columns.

3. Duplicate columns: “tweet_id”.

Deleted the last two tweet id columns, and renamed the first one, “tweet_id_x”, as “tweet_id”.

4. Breed information should be in one column.

First, identified the breed of each dog:

    a. Using the query function to extract the ones correctly recognized at the first attempt, second attempt, and third attempt, respectively. Then created a new “breed” column, and filled in their correct breeds.
    b. For the ones the failed to correctly identify, and the ones without image predictions, extracted them by the query function as well, but label the “breed” column as “Unknown”.

Second, deleted all the image recognition columns, keeping the “breed” column only. Then, appended the dataframes together.