Reporting: Wrangle Report

Quality issues:

1. ‘df’ dataframe includes retweets

Problem: Visual analysis revealed that ‘df’ dataframe contains retweets. This was indicated to us by retweeted_status_id. Programmatic analysis (pandas.info() was used) showed, there are 78 available values out of 2,356. Also, same analysis showed there was no missing data under rating columns, which indicates those values aren’t accurate. There is data **accuracy issue**.

Solution: Rows that contain retweeted rows need to be deleted.

2. ‘df’ dataframe contains data beyond 1st August of 2017, for which there is no image prediction data in ‘df_img’ dataframe

Problem: According to provided information, data after 1st August 2017 don’t contain image prediction. Image prediction was supposed to help us identify breed of dogs. We have missing data, which is **Completeness issue.**

Solution: In ‘df’ dataframe, we will need to change format of values under timestamp column. Pandas.info() revealed the format is object, aka strings. Firstly, the format will be changed to datetime using pandas.to_datetime(), after which we will filter out and drop rows after 1st August 2017.

3. In ‘df’ not all tweets are regarding dogs

Problem: In ‘df_img’ dataframe visual analysis revealed, that there are tweets about nondogs. The same tweets can be found in ‘df’ dataframe. The data has **validity issue**.

Solution: Identify non dog tweets by comparing p1_dog, p2_dog and p3_dog column values in ‘df_img’ ‘df’, if at least 2 out of these 3 is false, then the tweet_id will be appended to ‘non_dogs’ list. To calculate easily that more than 2 values is false, a function should be defined, if value equals true will return 1, if false – 0, and if sum will be less than 1, it means the tweet refers to nondog. Later the ‘tweet_id’s will be used to drop rows in both ‘df_img’ and ‘df’ dataframes.

4. In ‘df’ under `source` column, links have html tags and they should be removed

Problem: At first visual analysis revealed unnecessary html tags. Programmatic analysis (value_counts) showed that the source shows what device was used for tweeting. It was iphone, vine, web client and TweetDeck. The data isn’t missing or wrong, but this is **consistency issue**.

Solution: A for loop and nested for loop will be used to loop through each of the values under ‘source’ column and list made of values we mentioned above. Using .find() function, will identify if cell contains the word from the list and replace the old value with short version.

5. In ‘df’ dataframe need to remove rows, where values are not Null under `in_reply_to_status_id`, because they don't provide actual rating

Problem: Further analysis programmatic and visual analysis revealed that `in_reply_to_status_id’ doesn’t provide any rating just like in first issue. This also is **accuracy issue.**

Solution: Drop the rows where reply status column provides non null values.

6. `rating_numerator` and `rating_denominator` can be expressed in single column in `’df’` dataframe

Problem: These two columns can be provided as single row, by dividing rating numerator by rating denominator, since rating denominator is 10 everywhere. The issue in this case is **consistency** of the data.

Solution: Define a new column named ‘rating’, values of which are division of rating_numerator by rating_denominator.

7. Drop `name`, `doggo`, `floofer`, `pupper` and `puppo` colums. Big chunk of data is missing. Instead will merge breed name with highest confidence coefficient from `’df_img’` dataframe and also image url

Problem: There is data completeness issue. Combining last four columns wouldn’t make sense, because there still be left missing data. But we still will need dog classification.

Solution: From df_img dataframe use breeds with highest _conf value. During this operation we notice, that some breeds with False value under _dog columns have highest confidence coefficient. This requires us to delete those tweets too at the same time.

8. Values under `tweet_id` column in `’df’` dataframe are in int64 format, while they should be strings

Problem: values under tweet_id are int64 format.

Solution: Change format to string.

Tidiness Issues:

1. Add `retweet_count` and `favorite_count` columns from `tweepy_json` to `df` dataframe

From tweep_json we will require retweet_count and favorite_count values into single table, to make analysis.

2. Drop redundant and unnecessary columns from `df` dataframe

Drop unnecessary columns: 'in_reply_to_status_id', 'in_reply_to_user_id', 'retweeted_status_id', 'retweeted_status_user_id', 'retweeted_status_timestamp', 'dog_test'.