Data Gathering

There are 3 datasets which are gathered by separate methods.

1) twitter-archive-enhanced.csv >> This dataset is readily available in csv format. It contains the tweet id, retweet information, tweet text, image URL, and extracted data of dog name, rating_numerator, rating_denominator, and dog type.
2) image_predictions.tsv >> This dataset can be retrieved by using requests.get method. It contains tweeted, image URL, predictions whether the image is a dog image, dog breed, and corresponding probability.
3) tweet_json.txt >> This dataset can be retrieved by connecting to Twitter API using tweepy libray. The queried data are in JSON format. It contains tweet data, including tweet id, favourite count and retweet count.

Data Assessing

From visual assessment, I can see a few issues, including:

- The dataframe twitter-archive-enhanced includes retweets, and reply tweets.
- Float ratings are not correctly displayed.
- Some are not dog images.
- Some images are group of dogs, instead of individual dogs.
- Untidy columns for dog stage

From programmatic assessment, I can see a few issues, including:

- Out of range rating_numerator
- Rating_denominator that are not equal to 10
- Wrong dog names

Data Cleaning

Data Tidiness

I would like to have a single dataframe that contains each tweet's text, image, dog name, dog breed, dog rating, and the tweet's popularity in terms of favourite counts and retweet counts.

1) After merging twitter-archive-enhanced dataframe with image_predictions dataframe using tweet_id, there are many irrelevant columns to our analysis. Therefore I decided to remove those columns: in_reply_to_status_id, in_reply_to_user_id, retweeted_status_id, retweeted_status_user_id, retweeted_status_timestamp, p2, p2_conf, p2_dog, p3, p3_conf, p3_dog, doggo, floofer, pupper, puppo.

2) Merge with favorite count, retweet count info so that the final table contains the rating and popularity info of each tweet.

3) Melt the 4 dog stage columns

Data Quality

1) Currently if there are more than one string with the format ##/##, the first string that matches with the format is extracted as the rating. This can be wrong. For example, for the text **"*After so many requests, this is Bretagne. She was the last surviving 9/11 search dog, and our second ever 14/10. RIP https://t.co/XAVDNDaVgQ*"**, 9/11 is extracted as the rating. However, the correct rating should be 14/10.

We can improve this by adding a condition. Not only the format must be ##/##, it must also be ##/10.

2) Numerators that are floats should include digits before '.' For example, ***"This is Logan, the Chow who lived. He solemnly swears he's up to lots of good. H*ckin magical af 9.75/10 https://t.co/yBO5wuâ€¦"***, the numerator should be 9.75, not 75.

For 1) and 2), we can make use of the re library to search for the rating string more accurately.

3) The entries whose denominators do not equal to 10 seem to contain dog gang image. I decided to remove these entries as I wanted to analyze individual dogs, not groups of dogs.

4) twitter-archive-enhanced.csv contains retweet and reply tweet entries, which may cause duplicates of dogs. I had to remove these entries.

5) Remove joke ratings (outliers). For example, the tweet ***"This is Atticus. He's quite simply America af. 1776/10".*** 1776 is simply the year the USA was founded. Another tweet I found is Snoop Dogg, the artist.

6) Since my interest is in analysing dogs, I decided to remove non-dog entries based on P1 prediction data in image_predictions.tsv. I had to merge twitter-archive-enhanced dataframe with image_predictions dataframe using tweet_id.

7) I also decided to remove dogs whose breed cannot be identified from P1 prediction data.

8) It seems that the logic of dog name extraction is to fetch the word immediately after the first ‘is’. This results in many wrong name extractions. I decided to correct the names of the dogs 'the', 'a', 'an', 'actually'. There maybe more to be corrected, however this is not relevant to our analysis.

9) Convert tweet_id to string