briefly describing the wrangling in this project:

I started to gather the three datasets (archive, tweet and images)

Then I merged the data set archive to tweet to then dropped unneeded columns.

In images table:

- I dropped any value that have three false prediction confident form the algorithms to avoid any undog rating retweet.
- Then created nested loop to extract the dog type in each index and put it in new column called dog_type
- Then I merged the data set images to tweet to avoid unrelated tweets then dropped unneeded columns.

in tweet table:

- Changed the data type of (id and rating_numerator)
- Fixed inconsistent format (text)
- Dropped tweets that are retweeted

- Rating was extracted by the first slash which led to some invalid data

  I tried to discover a pattern to fix all rating so here what I conclude:

  - rating_denominator :

  1- tweets that have more than 40 in rating_denominator include more than one dog (except one fixed manually).

  2- tweets that have less than 40 and not equal to 10 rating_denominator contained slash (date or not actual rating) before the rating slash. (Except one that has no rating)

  - rating_numerator:

  1- some tweet that has high rating_numerator are decimal value and the number after decimal was taken.

  2- The lowest rating_numerator is zero it was not rating dog it just was calling out other account, so I was dropped.

- There were names that (a) , (an) or (this) so after visualization I notice that tweets was extracted by the next word after “this is” but they were many tweet that have this is followed by adjectives not the name however I notice in many text of them the name was after word “named” so I extracted some name the other I find no names in the tweet so I assign them to null.

- Dog stages (doggo,floofer,pupper,and puppo) I notice that the number of each one doesn’t match the one in the text so I extracted them again. Then I created new column called Dog stages to store them.

- Finally, I extracted the tweet URL form extended_entities in order to go directly to the source if needed, Also extracted month and year in created_at so I can used it later in visualizations.