Wrangling

Gatherings Steps

- Downloaded twitter-archive-enhanced.csv manually, and converted the .csv file to a dataframe.
- Downloaded image_predictions.tsv programmatically using the requests library and convered the .tsv file to a dataframe.
- Using tweepy, twitter api to attain the json objects contained in the archived-enhanced dataframe. Used the twitter id to download the content, captured deleted_tweets programmatically using try except constructs. Filtered exceptions and errors (whether twitter time delay error or other program error at first) and generated necessary delays in addition to using (wait_on_rate_limit= True, wait_on_rate_limit_notify = True) while accessing the twitter api to satisfy the time limits of access.
- Reading the tweet_json wrangled file to a dataframe, extracting necessary favorite_count, retweet_count information using pandas capabilities.

Difficulties:

- Rate limit leading to much time consumed in accessing the twitter api with every error in code.
- Understanding the underlying meanings of the most important tweet json attributes like entities, extended_entities. How retweets represented, user objects,etc.

Assessing Steps

- First, data is assessed visually, looking for multi-valued column like source, expanded_urls, any pattern in the text like “This is ..” to estimate how the other columns like name, ratings are computed in the dirty dataset, for any possible valued column names like pupper, puppo, doggo.
- Second, assessed data programmatically looking for incorrect datatypes .e.g object for datetime, int for string, any quality issues counterintuitive to visual assessment .e.g. Dog stages, floofer column containing no null, despite they do visually
- Third, iterating over value counts of interesting columns identifying quality issues like ‘a’, ’the’ and adjectives for names, tidiness issues as text column contained multiple information .e.g. urls, hashtags, mentions, retweets additions.
- Assessing why some columns contain many null .e.g. name, puppo, pupper, floofer columns figuring out how possible those columns were computed.
- Using number summaries to assess numerical column (rating_numerator, rating_denomunator) capturing quality issues .e.g. Non 10 rating denominator or unrealistic very high or very low valued rating numerators.
- Finally, assessing continued while cleaning data if a new quality problem is encountered.

Difficulties:

- Assessing name column was a bit difficult as even there is many names and many quality issues not restricting to ‘None’ objects.
- Programmatic assessment needed visual assessment almost always leading to much time wasted.

Cleaning Steps

- Text: separated text and the short t.co urls. Cleaned from truncated urls, retweets additions.
- Source: extracted source application and source application urls from the html tag. Dropped ‘rel’ no follow attribute as it’s not important
- Expanded_urls: separated urls from the main dataframe to a new dataframe containing individual urls, gathered image urls and cleaned unnecessary repetitive links.
- Name: cleaned from false names, extracted and recovered correct names for some entries.
- Converted columns to their appropriate semantically correct types (ids to strings, object dates to datetime)
- Removed many non dog tweets and cleaned rating numerators and denominator
- Retweets, in-reply tweets removed
- Merged data frames into one main dataframe

Difficulties and Notes:

- Some tweets and ratings were other users quoting WeRateDogs tweet non rating tweet and those users provided the ratings .e.g. athletes so there is some sort of delegation
- Some photos included multiple dogs, preventing me from converting pupper, puppo, doggo to a dogstage column.
- Extracting correct dog stages and floofer columns contain many nulls
- Some epanded_urls link contained tweet status ids not equal to any of the tweet or retweet or in_reply status ids
- When gathering images urls from json, some tweet entities or extended_entities object didn’t contain ‘media’ attribute although the tweet contained photos.