wrangle_report addition

Gathering data

There were 3 different sources for gathering the data. Thus, three different methods.

1. Source: Ready to download as 'twitter-archive-enhanced.csv'.
   Method used to read it: Reading the csv file it into a DataFrame using read.csv function.
   For what: To have ratings and etc..

2. Source: A URL containing ‘image_predictions.tsv’.
   Method used to read it: Getting the tsv file via *request ing it. Opened a file location in memory for (kind of) saving the file there. Reading the saved tsv file into a DataFrame* using read.csv function.
   For What: To get the name of the breeds and etc..

3. Source: Twitter API
   Method:
   1. Building a Twitter developer account →Requesting for an elevated access→ Conider the tweet IDs in twitter_archive.csv→ query the Twitter API for each tweet's JSON data using Python's Tweepy library→ store each tweet's entire set of JSON data in a file called `tweet_json.txt` file→ Read `tweet_json.txt` into a pandas DataFrame.
   2. Twitter was asking too many questions and lingering for providing me with the API so I chose to go with the second method:
      download the `tweet_json.txt` → uploeaded it in my space→ read it line by line and (chose to get) got ‘likes’, ‘retweets’ and ‘ids’ for each tweet→ saved it to a list→ Reading the data into a pandas DataFrame called ‘additional_archive’.

   For What: To have the most liked one(breed) and etc. .

Assessing data

I first started with twitter_archive table, wrote all the issues about it. Then, furthered with other tables. I did almost the same steps for three tables.

1. I searched for the datatype issues using df.info().

2. I looked up for NaN values.

   Some columns had lots of them which made me thinking why. I noticed that the reason is those columns are mostly data about retweets and replies not the tweets. Here I note down to delete the rows with tweet ids that belong to retweets and replies. Then, delete the columns of retweet and reply data.

3. I saw lots of None values which should be replaced with NaN.
   For ‘name’ column I could use replace method.
   For growth stages, this issue will automatically be rectified when combining the columns into one column called ‘growth stage’.

4. Using df.name.value_counts I noticed that there are wrong names in the column. Since ‘name’ is not a decisive variable in our study case, the best treatment was to change them to NaN. (Instead of searching the text for correct names)

5. I googled about WeRateDogs rating system and noticed that denominators must be 10 which in some cases were not. Since troubleshooting this problem was obvious (changing all numbers into 10), I didn’t look any closer to the numbers. I also noticed weird numbers in rating nominators. Since there is no specific rule or hint for the *accurate* number, the best way is to check them with the real text. I looked up in the text and noticed that some numerator ratings had been given in decimal format so the first change in cleaning process should be changing the rating_numerator column into float

6. Tables can be merged.

Cleaning

1. Changed the datatypes to avoid any errors in the rest of the process.
2. Combined dog growth stages into one column to get finished with tidying the first table.
3. Merged tables on tweet_id.
4. Removed retweet and reply rows and then columns to get faster and redundancy-free result in next steps.

Order of the cleaning, from now on, is not a big issue.

The biggest effort this cleaning took was correcting the numerators rating using both regex expressions and manual practice. The algorithm didn’t consider decimal points and people entering more than one ‘/’ while writing their ratings. This was the reason why numerators in the table differed from that of the text.

Mojde Bay