wrangle_report

January 18, 2023

0.1 Reporting: wrangle_report

- Create a **300-600 word written report** called "wrangle_report.pdf" or "wrangle_report.html" that briefly describes your wrangling efforts. This is to be framed as an internal document.

The subject dataset to be worked upon is the tweet archive of Twitter user @dog_rates, also known as WeRateDogs. WeRateDogs is a Twitter account that rates people's dogs with a humorous comment about the dog. These ratings almost always have a denominator of 10. The numerators, though? Almost always greater than 10.

0.1.1 Data Gathering

The dataset were to be gathered from 3 different sources. 1. Directly from WeRateDogs twitter archive data that was provided in the project description. 2. I also had to use the request library to access WeRateDogs tweet image predictions hosted on udacity 3. The third dataset of WeRateDogs twitter account was directly acccessed via Twitter's API or each tweet's JSON data using Python's Tweepy library and stored each tweet's entire set of JSON data in a file called tweet_json.txt file. Each lines was then read and converted to a dataframe with tweet ID, retweet count, and favorite count as columns.

0.1.2 Data Accessing

The first dataset,WeRateDogs twitter archive data contain 2356 entries, 17 columns

The columns including Columns: tweet_id, in_reply_to_status_id, in_reply_to_user_id, timestamp, source, text, retweeted_status_id, retweeted_status_user_id, retweeted_status_timestamp, expanded_urls, rating_numerator, rating_denominator, dog name, doggo (dog stage), floofer (dog stage), pupper (dog stage), puppo (dog stage) The second dataset of WeRateDogs tweet image predictions contains 2075 entries, 12 columns

Columns: tweet_id, jpg_url (dog prediction image), img_number (image number of tweet), p1 (p1 is prediction order 1 out of top 3 image predictions), p1_conf (prediction confidence), p1_dog (whether the prediction is dog breed or not),p2, p2_conf, p2_dog, p3, p3_conf, p3_dog The third dataset contains 2354 entries, 3 columns

Columns: tweet_id, retweet_count, favorite_count After visual and programmic assessment of the 3 dataset the following quality and tidiness issues were detected.

Quality issues

1. Completeness: There are significant number of missing values in the twitter archive dataset
2. The name column has a lot of duplicated values and some names are None
3. Tweet ids are currently in integer but they should be strings
4. The timestamp is in integer format rather than datetime
5. Accuracy: Nulls represented as None in name column
6. Rating_denominator contains values other than 10, minimum is 0
7. The url still has source display
8. There are Urls in the end of the ’text’.

Tidiness issues

1. The dog stages are in separate columns but they should be in one
2. The three dataset have tweet id in common, hence could be combined

0.1.3 Cleaning Data

A copy of the original set of data were made and the following set of issues were solved and the approach

Issue 1: Multiple DataFrame Approach: Merging the dataframes based on tweet_id

The WeRateDogs twitter archive data and WeRateDogs tweet image predictions were merged first * df = pd.merge(data, df_req, how=’outer’, on=’tweet_id’, sort=True)

and then the merged data set is merged with the WeRateDogs tweet ratings.

- df = pd.merge(df, df_json, how=’outer’, on=’tweet_id’, sort=True)

Dimensions of the final cleaned dataframe * Dimensions: (2005 rows, 25 columns

Issue 2: Missing Values Approach: fixing Null and None values by replacing with 0 where appropriate and dropping where appropriate

Issue 3: Source Html Tag in Url Approach: By Extracting only URL using regex characters

- df.source = df.source.str.extract(’"([ˆ"]*)" ’)

Issue 4: rating_denominator column contains values less and larger than 10 Approach: Fixing denominator values other than 10

Issue 6: Erraneous datatypes in (favorite_count, retweet_count, p_dog, img_num and timestamp) columns Approach: Conversion to int, timestamp where necessary

- df.favorite_count = df.favorite_count.astype(int)
- df.retweet_count = df.retweet_count.astype(int)

Issue 7: name column Nulls represented as None Approach : Replacing None values in the name column with NaN

- df.name = df.name.replace('None', np.nan)

Issue 8: name column have multiple invalid values ('a', 'an', 'the') Approach: Replacing the invalid names with NaN

Issue 9: dog stage columns contain multiple dog stage at once Approach: Cleaning the records that contain more than one stage value referring to their text

Issue 10: multiple dog stage in different columns Approach: Merge all dog stages into a column and convert to categorical variable

Issue 11: dog_breed must be added based on the img_num and p1_dog Approach: Creating new dog_breed column based on the img_num prediction then dropping all prediction column

Storing Data The cleaned master DataFrame is then stored in a CSV file with the main one named twitter_archive_master.csv * df.to_csv('twitter_archive_master.csv', index = False)