wrangle_report

September 18, 2022

1 WeRateDogs Wrangle Report

BY: AKOLADE SOFIYYAH IWALEWA

1.0.1 OBJECTIVE: Wrangle a data from @WeRateDogs twitter acount to derive meaningful insights and visualizations.

1.0.2 DATASET INTRODUCTION:

This datasets contains 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. Almost all the numerators are always greater than 10. For example, 11/10, 12/10, 13/10, etc. The numerator are mostly bigger Because "they're good dogs Brent." WeRateDogs has over 4 million followers and has received international media coverage. This unique rating system is a big part of the popularity of WeRateDogs.

I made use of these data wrangling processes. I will discuss each process in details - Data Gathering - Data Assessing - Data Cleaning - Data Storage

1.1 DATA GATHERING

I started my data gathering by importing all neccessary packages that I would be using for analysis, such as pandas, numpy, scipy, os, json, request and others. The datasets used in this analysis were gathered from three different sources

1. The twitter_archive_enhanced.csv file, which was downloaded manually and loaded into my jupyter notebook using using pandas (pd.read_csv).
2. The image predictions file which was downloaded from this url "" using the request libarary.
3. The last dataset was gotten by querying the Twitter API for additional using Python tweepy libarary.

After I have successfully gathered the three datsets, I moved on to assessing the three datasets

1.2 DATA ASSESSSING

I assesed the datasets visually and programatically, that is I scrolled down to check the datasets and also wrote some codes to check if there are any quality or tidiness issues with the datasets. I made use of some pandas codes such as head(),info(),shapes(),unique(), describe(), tail(), value_counts() and so on. After I ahve assesed progamatically and visually, these are the quality and tideness issues that I observed.

1.2.1 Quality Issues

- Irrelevant column: Retweet status Id
- Rows with invalid denominator
- Source column is a combination of url and text (only text is needed)
- Incorrect dog names in name column
- Irrelevant column: img_num
- Missing values in the following columns: in_reply_to_status_id, in_reply_to_user_id, retweeted_status_id, retweeted_status_user_id, retweeted_status_timestamp, expanded_urls columns
- Redundacy in text column
- Incorrect data types in the following colums: tweet_id, timestamp, dog_stage, p1_dog, p2_dog and p3_dog.

1.2.2 Tideness Issues

- Dog_stage in Twitter Archive table(df_1) is in 4 columns instead of 1.
- Twitter archive data (df_1), image Predictions data(df_2), and Tweets data (df_3) should be a single dataframe

After observing all these issues with the datasets, I cleaned the datasets so I could use high quality data for my analysis.

1.3 DATA CLEANING

Before I began the data cleaning, I made original copies of the three datasets. Then I wrote codes to perform the following cleaning:

- Drop the rewteet status id, because the information is not needeed
- Drop rows where rating_denominator is not 10 (the rows to be dropped are 23 as shown above, the number rows to be drop are relatively very small to the total number of rows)
- Extract source text from source column in twitter achive data(df_1_clean)
- Remove rows where dog names are not real (I.e. the dog names are in lower case)
- Drop the img_num column in df_2_clean, It's not needed for analysis

- Drop columns in_reply_to_status_id, in_reply_to_user_id, retweeted_status_id, retweeted_status_user_id, retweeted_status_timestamp, expanded_urls (they contain missing values and they won't be needed for analyis)
- Combine doggo, floofer, pupper, and puppo columns into a single column called 'dog_stage', using lambda, Then drop the columns
- Combine Twitter archive data(df_1_clean), Image Predictions(df_2_clean), and Tweet data (df_3_clean)
- Drop the text column
- Convert tweet_id to string, timestamp to datetime, dog_stage to categorical, p1_dog, p2_dog and p3_dog to categorical datatype.

  After writing codes to carry out all the above, I also wrote testing codes to ensure all the issues have been resolved. Then I proceed to saving the datasets.

1.3.1 DATA STORAGE

After Cleaning, I saved and combined all the datasets as a single dataframe.