WeRateDogs Project: Wrangling

Introduction

WeRateDogs (https://twitter.com/dog_rates) 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. However, the numerators are almost always greater than 10. (11/10, 12/10, 13/10, etc.) Why? Because "they're good dogs Brent."

WeRateDogs has over 4 million followers and has received international media coverage. So, what to do with these ratings? With dogs the best common question has to be which is the most popular dog? Can we get any relationship between Retweets, Favorites and Ratings? I have wrangled the WeRateDogs twitter data and have analyzed it to get answers of these questions.

Here, I will outline the steps taken to gather, assess and clean the data.

Data gathering

The data for this project consist on three different dataset that were obtained as following:

- Twitter archive file: the twitter_archive_enhanced.csv was provided by Udacity and downloaded manually.
- The tweet image predictions: This file (image_predictions.tsv) is hosted on Udacity's servers and was downloaded programmatically using the Requests library and URL information. It contains three predictions of dog breeds.
- Twitter API & JSON: by using the tweet IDs in the WeRateDogs Twitter archive, I queried the Twitter API for 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. I read this .txt file line by line into a pandas dataframe with tweet ID, text, favorite count, retweet count, source, retweeted status and url.

Data assessment

After gathering the three datasets, I assessed them visually and programatically.

- Visually, I set the display of columns to maximum in jupyter notebook.
- Programmatically, I used different pandas methods such as .info(), .describe(), .duplicated() etc.

Quality issues

- Some dogs names are invalid. Example 'a'
- The tweets sources in source are embedded in a url

- Missing values in names, doggo, floofer, pupper and puppo are represented as None instead of NaN
- tweet_id should be string object while timestamp, and retweeted_status_timestamp should be datetime objects
- Some columns have missing data
- The maximum value of rating_numerator and rating_denominator is outrageous
- Some of the ratings seem valid. Examples are those with index number 185, 285 etc.
- Others are incorrect. It appears that some ratings are in decimals and only the digits after the . were recorded while some are cummulative ratings of more than one dog.
- Some digits extracted as ratings were not ratings. An example is the rating with index 516 which was referencing 24/7 and index 342 which was referencing 9/11 as a date
- There are some invalid names such as O, a, actually, all, an, getting, by, his, incredibly, infuriating, just, life, light, mad, my, not, officially, old, one, quite, space, such, the, this, unacceptable, very
- There is inconsistency in the naming, some were written in lowercase while others were in Sentence case
- Missing names were represented as None instead of NaN
- The dog with name O real name's O'Malley
- The name of some dogs were not included in the tweet, just the type of dog while other dog's name were mentioned after 'named'
- Some of the tweets were not dogs

Tidiness issues

- Doggo, floofer, pupper and puppo are different stages of dogs and should be in one column
- All the 3 datasets should be merged together

Data cleaning

Before diving into cleaning, I made a copy of each dataset.

Data cleaning is divided into three parts: Define, code and test the code. These three steps were on each of the issues described in the assess section. Each of the issues raised in the assessment stage was fixed. Some data were dropped because they do not have pictures.

doggo, floofer, pupper and puppo columns were put in one column; `dog_stage` . This was done by extracting the dog's stage from the each tweet. Afterwards, the three datasets were merged to one dataset. Dog ratings and names were also fixed.

Conclusion

Data wrangling provides a clean data for analysis. At the end of the analysis, three datasets from different sources was merged to one. This cleaned data will be used in future analysis

In [ ]: