Introduction

Real-world data rarely comes clean. Using Python and its libraries, you will gather data from a variety of sources and in a variety of formats, assess its quality and tidiness, then clean it. This is called data wrangling. You will document your wrangling efforts in a Jupyter Notebook, plus showcase them through analyses and visualizations using Python.

The dataset that you will be wrangling (and analyzing and visualizing) is the tweet archive of Twitter user dog rates, also known as WeRateDogs is a Twitter account that rates people's dogs with a humorous comment about the dog

Project Motivation

Context

Your goal: wrangle WeRateDogs Twitter data to create interesting and trustworthy analyses and visualizations. The Twitter archive is great, but it only contains very basic tweet information. Additional gathering, then assessing and cleaning is required for "Wow!"-worthy analyses and visualizations.

The Data Gathering process

Enhanced Twitter Archive

The WeRateDogs Twitter archive contains basic tweet data for all 5000+ of their tweets, but not everything. One column the archive does contain though: each tweet's text, which I used to extract rating, dog name, and dog "stage" (i.e. doggo, floofer, pupper, and puppo) to make this Twitter archive "enhanced." Of the 5000+ tweets, I have filtered for tweets with ratings only (there are 2356).

Additional Data via the Twitter API

Back to the basic-ness of Twitter archives: retweet count and favorite count are two of the notable column omissions. Fortunately, this additional data can be gathered by anyone from Twitter's API. Well, "anyone" who has access to data for the 3000 most recent tweets, at least. But you, because you have the WeRateDogs Twitter archive and specifically the tweet IDs within it, can gather this data for all 5000+. And guess what? You're going to query Twitter's API to gather this valuable data.

Image Predictions File

One more cool thing: I ran every image in the WeRateDogs Twitter archive through a neural network that can classify breeds of dogs*. The results: a table full of image predictions (the top three only) alongside each tweet ID, image URL, and the image number that corresponded to the most confident prediction (numbered 1 to 4 since tweets can have up to four images).

Assessing and cleaning the data

Archive Data frame

1. Timestamp is object but it should be a datetime as it represent time on which tweet uploaded

2. retweeted_status_timestamp is object but it should be a datetime as it represent time on which tweet is retweeted

3. name column should be convert to dog_name as it may confuse reader as it is dog name or owner name

4. Rating_denominator ranges from 0-10 but this cause issues when we calculate rating ratio as diving number by zero lead to mathmatical issues this should be changed

5. there is outliers in denominators as the maximum value is 177

6. doggo, floofer, pupper, puppo columns contain 'None' value where NaN should be used.

7. there are 9 dogs with non-english names so i will drop them as in this analysis i intersted in english dogs only

8. Missing values in 'name' showing as 'None'

image_preds Dataframe

1. change columns name to something informative for example p1 to first_predection / p1_confd to first_prediction_confidence

Tideness

1. img_num column in image_preds contain number for images in tweet i think this is un-useful info so i will drop this column
2. doggo, floofer, pupper, puppo columns are redundant as they describe same thing
3. we need to join dataframes to make clear and complete picture for the entire information needed for the next step