Data Analyst Nanodegree – Jonathan Ravel – Project 3

We Rate Dogs

Section 1 – Gathering Data

We begin by gathering the data from three different datasets.

Twitter-archive-enhanced.csv was simply loaded with the read csv function.

The image-predictions.tsv file was obtained by getting the information from the cloudfront url, then similarly loaded into a csv with a delimiter.

Lastly the Twitter archive was accessed through its API. For the purposes of the Jupyter Notebook, coding steps are shown but keys hidden, and these steps are commented out. This data was loaded into a dataframe then saved as a csv.

Section 2 – Assess

Now we perform a simple visual assessment of the 3 datasets.

Twitter Archive

We will have to investigate the in_reply_to_status and in_reply_to_user_id columns. We may not want to include tweets that are replies. The timestamp may need to be split into date and time if we want to analyze ratings by year. The meaning of the source column isn't entirely clear at first glance. The retweeted_status_id will be used to remove retweets per instructions, and then we won't need the associated columns of retweeted_status_user_id and retweeted_status_timestamp. At this point it is not clear if we will be able to utilize the expanded_urls field but we can investigate. The rating_numerator and rating_denominator fields invite programmatic assessment. It would be fun to run value counts on the name field. Lastly some of the stages of dogs might be able to be combined into one column, but only if a dog can be only one of these stages.

Image Predictions

This is a fun side project to see if the image classifier was able to do a good job identifying if the images were dogs or not. However, there are no dog breeds in the Twitter Archive or API data so we won't be able to do that much with this information. Some data is capitalized but others lower case, so we can fix this for consistency. Also, the confidence percentage would be more readable if it wasn't six digits.

Twitter API

This data gives us important engagement metrics that will be interesting to investigate with other variables. The user_count information is not clear.

Using functions, now we will perform a programmatic assessment of the data and list items to clean for quality and tidiness.

Quality

Twitter Archive

- tweet_id is an int64 field and needs to be a string
- in_reply_to_status_id and in_reply_to_user_id have 78 non-null values that can be removed
- retweeted_status_id contains 181 non-null items that can be removed
- timestamp could be more helpful if split into date and time
- many ratings are incorrect; defined as greater than 15 in the numerator
- many denominator values aren't 10 (since 10 is always the scale)

Image Predictions

- tweet_id is an int64 field and needs to be a string
- capitalizations are not consistent
- confidence percentage format of 6-digit decimal is hard to read

Twitter API

- tweet_id is an int64 field and needs to be a string

Tidiness

Twitter Archive

- remove columns that will be left with null data: in_reply_to_status_id, in_reply_to_user_id, retweeted_status_id, retweeted_status_timestamp
- combine the floofer, pupper and puppo fields into one column (all dogs can be doggos as well as the other categories)

Twitter API

- drop user_count

Combined

- join data on tweet_id into one dataset

For the cleaning steps and tests, please see the Jupyter notebook.