Wranglin We Rate Dogs

Lauriston Nunes

The first step in the data wrangling process is to gather the data. We will be working with three data sources, the WeRateDogs Twitter archive, the tweet predictions, and tweet JSON which contains each tweet's retweet and favorite count. First, we'll gather the WeRateDogs Twitter archive. This is done by simply saving the twitter-archive-enhanced.csv locally to my desktop, uploading it to the Udacity workspace, and then reading the csv into a dataframe. In order to gather the tweet predictions we'll use the requests package to pull the content from https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-predictions/image-predictions.tsv.
In order to gather the tweet's retweet and favorite tweet counts, we'll query Twitter's API for JSON data for each tweet ID in the Twitter archive and save the results to tweet_json.txt. From the tweet_json.txt we'll parse out the id, *favorite_count*, *retweet_count*, and *full_text* for us to analyze. I had to manually remove a comma at the end of the file and add brackets [] to ensure the file was in valid JSON format for me to parse.

The second step in the data wrangling process is to assess the data. We can assess the data visually and programmatically. Here are the issues I found using both assessment methods:

Quality

twitter_archive table

- 2175 records have not been retweeted
- Name has values like \`None\` and \`the\`, \`a\`, \`an\`
- Some ratings are not out of 10 ex, 666287406224695296
- Timestamp should use the datetime64 data type
- \`None\` string in Dog stage data
- Missing Dog stage data
- Dog stage data should use the category data type

image_predictions_raw_data table

- Missing images

Tidiness

- Dog stage data should use the category data type and be consolidated under one column
- *tweet_id* column in **twitter_archive** table duplicated in **image_predictions_raw_data** and **tweet_json** tables
- *favorite_count* and *retweet_count* should be part of **twitter_archive** table

The third step in the data wrangling process is to clean the data. We will also need to go back to the gather step to improve data quality and data tidiness for cleaning. For the *2175 records have not been retweeted* issue, we only want to select records that have not been retweeted. For the *Name has values like `None` and `the`, `a`, `an`* issue, we'll remove records that have names with values like None and that start with a lower-case letter. It looks like the name was pulled from regular expression that pulls "This is ". In the case where the regular expression returned *None*, the "This is ..." expression was not found. In the case where the expression was found, but with a lowercase letter, the string was not a name. For the *timestamp should use the datetime64 data type*, we will convert timestamp to datetime64 data type. For the *Some ratings are not out of 10* issue, we will update the *rating_denominator* to 10. For the *`None` string in Dog stage data*` issue, we will replace `None` with NaN. This will give us a better idea of how many dog stage values we actually have. For the *Missing Dog Stage* data issue, we can parse the key word from the text. Another issue with the dog stage data is that it should use the category type. We can convert this from a string object to a category object since the same values are repeated. Last but not least, we tidy our data. We achieve this by merging all of the tables together based on the *tweet_id* and dropping unused columns, such as the *retweeted_status_id*, since we will not be analyzing retweets.