WeRateDogs Project: Wrangle Report

David Hundley
Udacity - Data Analyst Nanodegree
March 2019

Introduction

Throughout this report, I'll walk through what I did to get the datasets, the sorts of assessments I did on them, and what specific things I went about clean up. Let's go ahead and jump into it!

Gather

There are three different raw datasets I pulled from for the purposes of this project. This included the following:

- **Localized Twitter Archive**: This dataset was specifically provided to me by Udacity. After uploading this to the Jupyter workspace, I simply read it into a DataFrame as local_df.
- **Twitter Info from Tweepy API**: Utilizing Twitter's own Tweepy API, I extracted the necessary information utilizing the IDs from the Localized Twitter archive and stored them into its own JSON file. From there, I built a DataFrame from that JSON file and called it tweet_json_df.
- **Image Predictions Dataset**: Finally, we got this dataset from a URL provided by Udacity. Specifically, I used the Requests package to gather the content from the URL and write it into a DataFrame called image_predictions_df

Assessment

Assessment of these datasets in their raw form was pretty standard, run of the mill stuff. Specifically, I assessed these datasets in the following manners:

- **Manually**: This was a matter of loading up these DataFrames in their raw form, looking at their data, and making notes of things that looked off kilter.
- **Programmatically**: Knowing that manual assessment would only go so far, I ran some methods on these DataFrames to glean additional insights that wouldn't be readily available from simply glancing at the data

Cleaning

Finally, I went through the datasets and cleaned up all the data for later analysis. This data was cleaned for both quality issues and tidiness issues, and the final cleaned dataset was stored into twitter_archive_master.csv. Below are the specific issues I cleaned for:

Quality Issues¶

- Cleaning up the source column in local_df
- Cleaning up the source column in tweet_json_df
- Extracting the rating from the text in tweet_json_df
- Dropping rows with missing values for rating in tweet_json_df
- Adjusting the timepstamp in local_df to only reflect date and not time
- Dropping retweeted rows from local_df (since they are essentially duplicating information)
- Dropping retweeted rows from tweet_json_df (since they are essentially duplicating information)
- Providing more descriptive column names in image_predictions_df since I found them confusing
- Adjusting the letter case on each value in the prediction columns to match all upper/lowercase names to be a consistent format
- Dropping missed rows from all tables where the Twitter API failed to gather information on a specific ID
- Adjusting ID column name in the image predictions dataset

Tidiness Issues¶

- There are many columns that can be dropped from the datasets altogether as they won't be used in our final assessments. These columns include...
  - From local_df
    - in_reply_to_status_id
    - in_reply_to_user_id
    - expanded_urls
    - text
    - rating_numerator (since we'll merge in the one from the API later)
    - rating_denominator (since we'll merge in the one from the API later)
  - From tweet_json_df
    - retweeted
  - From image_predictions_df
    - img_num
- The dog categories like floofer, puppo, and more are each its own column and can really be combined into one column with different representing values
- Finally... all three of these datasets can really be combined into a singular one due to the nature of their adjoining content