Project: Wrangle and Analyze WeRateDogs Tweets

Muthukumar Palavesam

Introduction:

The dataset that I will be wrangling (and analyzing and visualizing) is the tweet archive of Twitter user @dog_rates, also known as WeRateDogs. WeRateDogs 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. The numerators, though? 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.

In this project, I am using Python and its libraries. This project includes Four major steps to do the data wrangling:

1. Gather the data from a variety of sources and in a variety of formats
2. Assess its quality and tidiness
3. Clean the data
4. Store the Data

Gather the data:

For this analysis I gathered data from three different sources:

- Direct `CSV File` download
- Using `requests` library in python
- Extract the data from Twitter using `tweepy` library in python

Direct `CSV File` download:

Using the link provided by Udacity, I downloaded the WeRateDogs Twitter archive manually from the link (https://d17h27t6h515a5.cloudfront.net/topher/2017/August/59a4e958_twitter-archiveenhanced/twitter-archive-enhanced.csv). Then I copied this file and paste it into my local project folder. After that, I used the pandas read_csv function to read the stored CSV file and named it as 'tweet_archieve'

Using `requests` library in python:

I downloaded the tweet image predictions file hosted on Udacity's servers programmatically using Python's Requests library and saved it locally to image_predictions.tsv file. Then, I imported this file into a Python Pandas dataframe (img_predict).

Extract the data from Twitter using `tweepy` library in python:

Using the tweet IDs in the Twitter archive, I accessed the entire data for every tweet from Twitter API and stored every tweet's entire set of JSON data in a file called tweet_json.txt file. Created a dataframe tweets_data from this JSON including only tweet_id, favorite_count, retweet_count.

Assess its quality and tidiness:

After gathering the above pieces of data, I assessed those visually and programmatically. I looked for quality and tidiness issues. The issues I found are the following.

Quality:

Dataset Name: tweet_archieve

- Need to remove `+0000` from timestamp column
- Extract Date and Time from the timestamp column and Convert tweet_date column datatype from object into DateTime
- tweet_id column should be a string datatype
- `source` column is lengthy and unwanted HTML tag should be removed
- Incorrect rating_nominator and rating_denominator values
- tweet `text` column is showing with URL.
- Some of the data in `name` column are not the actual name (Example: a, an, actually, by). Need to correct this
- Retweets should be removed and Retweets associated columns not needed

Dataset Name: img_predict

- tweet_id column should be a string datatype
- dog breed predictions (p1, p2, p3) column:
- dog breed predictions (p1, p2, p3) should be represented as categorical datatype
- `p1, p2, p3` columns there are Underscore (_) between words
- `p1, p2, p3` columns entries are not all capitalized

Dataset Name: tweets_data

- favorites and retweets showing as object, need to change the datatype into int64

Tidiness:

- doggo, floofer, pupper, and puppo columns in the tweet_archieve_clean table should be merged into one column named "dog_stage" and Convert the dog_stage column datatype from string to categorical
- img_predict_clean table contains no dogs' records, so we need to exclude those records and adding the dog breed with the highest confidence value (i.e. `p_conf` values of each prediction)
- Join all 3 datasets (tweet_archieve_clean, img_predict_clean, dog_breed_df) into one dataframe called twitter_archive_master
- Remove the unrequired columns
- Rename the columns into the meaningful description and reorder it

Clean the Data:

Once I had successfully gathered all the data, I created a copy of all the dataset and started the data cleaning processes. I looked into the quality and tidiness issues mentioned above one by one and then set about fixing them. For each quality/tidiness issue, I performed cleaning systematically, using "Define," "Code," and "Test" sections for each issue identified during the assessment. This ensures that results are as expected.

Storing the Data:

I stored the cleaned merged DataFrame in twitter_archive_master.csv file into my local project folder