Data Wrangling report

WeRateDogs Twitter archive data

Libraries Used

os, requests, pandas, tweepy , json, timeit , re, numpy, matplotlib

Gathering Data

- Data was gathered manually and programmatically using the ‘Requests’ library.
- All the data gathered was transformed into pandas dataframes.
- The data gathered through the ‘twitter_archive_enhanced.csv’ file was however incomplete and thus the need for getting the additional data fields using Twitter API.
- In order to authorize our app to access Twitter on our behalf, we need to use the OAuth interface (using tweepy). To do this following were obtained from my Twitter account:
  - consumer_key, consumer_secret, access_token and access_secret.
- Using the tweet IDs in the WeRateDogs Twitter archive (i.e. twitter_archive_enhanced.csv), the Twitter API was queried for each tweet's JSON data using Python's Tweepy library.
- All this json data for all the tweets was written to a file called tweet_json.txt file. Each tweet's JSON data should be written to its own line. This took about 27 minutes for me. I struggled a lot for this step as I was able to write the json data into a text file, but not line by line.

Wrangling Data

- Using the json.load function (from the json library) the json data from the tweet_json.txt was written to a data object.
- Using the data object created in above step, I extracted tweet_id, retweet_count, favorite_count, retweeted_status, and even full_text columns. I added these as columns in df_tweets pandas dataframe.
- I added data from each file to respective pandas dataframes for:
  - twitter_archive_enhanced,
  - image_predictions
  - 'df_tweets' (subset of json data obtained using Twitter API)
- I then merged the three dataframes ('df_tweets', ‘img_pred’ and 'twitter_archive_enh') to form a master dataframe 'twitter_archive_master'.

- I assessed the master dataframe using .info(), .head(), .tail(), .sample() methods
- I noted down all the issues upon discovery one-by-one and categorized them as ‘Quality’ issues and ‘Tidiness’ issues.
- In order to clean the dataset I first created a copy of the 'twitter_archive_master' dataframe.
- Each issue was then addressed first by elaborating the issue definition with steps to resolve it. This was followed by code and then by test code.
- Following are the quality and tidiness issues I discovered and resolved:

# Assessment Summary
## Quality issues

1. Name for tweet with ID = 778039087836069888 should be ‘Max’.

2. Record with Tweet_id = 887517139158093824 has incorrect name = 'such'.

3. Record with Tweet_id = 887473957103951883 has name = NaN instead of 'Canela'.

4. Many rows have incorrectly extracted names such as 'a','the','an'.

5. Some rows have rating_denominator != 10.

6. The same rows above have incorrect numerators too.

7. Record with Tweet_id = 828011680017821696 actually has two dogs Brutus and Jersey. Need to capture info for both and distinguish the two.

8. Record with Tweet_id = 678396796259975168 actually has two dogs (names not provided). Need to capture info for both and distinguish the two.

9. Record with tweet_id = 825147591692263424, has name 'sweet pea' and not just 'sweet'.

10. Utility used to post the tweet is embedded within the url in source column. Need to extract it.

11. Most missing values are indicated by 'None' string instead of np.NaN

11. Datatypes for following columns are incorrect:

- tweet_id is int instead of string
- in_reply_to_status_id, in_reply_to_user_id must string
- timestamp, retweeted_status_timestamp should be datetime
- retweeted_status_id, retweeted_status_user_id must be string

## Tidiness issues
1. The dog stages 4 columns should actually be just one column 'dog_stage' with options viz. pupper,puppo,doggo,floofer with datatype category

2. The expanded_urls column has same url listed once/twice/thrice depending on number of images uploaded.

- After addressing above issues I wrote the clean dataframe to a .csv file
- Using the clean dataframe, I created data visualizations and drew insights.