Wrangling Report for the WeRateDogs Udacity Project

Overview:

This report is a submission for a Udacity Project that is part of the Data Analysis Nanodegree that they offer. The process of the wrangling report involves gathering data from different sources, assessing the quality and tidiness, cleaning what we have assessed, performing analysis of the newly cleaned dataset.

Data Gathering:

For the project, the data was gathered from three different resources:

1. twitter_archive_enhanced.csv: A downloadable file provided by Udacity from The WeRateDogs Twitter archive.
2. image_predictions.tsv: The tweet image predictions, i.e., what breed of dog (or another object, animal, etc.) is present in each tweet according to a neural network. This file is hosted on Udacity's servers and should be downloaded programmatically using the Requests library.
3. The Twitter API: By requesting data from it.

Firstly, the twitter_archive_enhanced.csv file was downloaded normally and the uploaded to the Jupyter notebook. Secondly, image_predictions.tsv file was downloaded programmatically using the requests library. Thirdly, the Twitter API was accessed using the tweepy library.

To access the Twitter API, I had to fill an application on the Twitter developers' website in order to be able to access the data. After filling, I was able to access the keys they provide to authenticate. Then, I used Tweet IDs in the twitter_archive_enhanced.csv to query the API for information relevant to analysis goals like retweet counts and favorite counts and saved as a txt file, "tweet_json.txt" in JSON format.

After collecting the data, they were imported into Pandas' DataFrames.

Assessing Data:

After importing data, I printed each DataFrame to inspect them visually. Firstly, I was trying to understand each column in detail, what does it represent and what does it mean? Then I started to check the values of the column. I used the sample pandas' method to have a random sample of each DataFrame and see if there is something wrong.

In the twitter Archive DataFrame, Firstly, I noticed that the `retweeted_status_id` column has some values for the id of users which we do not need for analysis. We are only looking for original tweets. Secondly, I noticed that these columns: `in_reply_to_status_id`, `in_reply_to_user_id`,

`retweeted_status_id`, `retweeted_status_user_id` and `retweeted_status_timestamp` are not necessary for the analysis, so they should be dropped.

Then I used the Programmatic Assessment to check more for quality and tidiness issues with the data. When I used the .info Pandas' method, I found that some columns have wrong data types like the tweet_id is treated as int while it should be a string as it is a unique ID. Additionally, I found that timestamp is treated as an object while it should be a DateTime.

Moreover, the `rating_numerator` and the `rating_denominator` type were integers and it should be floats instead. Then I started to check for specific columns to check its quality. I found that The rating_denominator column has some values larger than ten which is not possible of course. Also, the dog names had lots of mistakes in the data. I have seen some weird values like a, an, they ..., others have None values in the column name.

In the `tweet_json` DataFrame at "retweeted_status" column, I found using .unique Pandas' method that it has 2 values, the first one is Original tweet and the second one is This is a retweet. Since we should only work with original tweets, the retweet values should be dropped.

Lastly, I checked for tidiness and I found out that 3 main things. The first is that the 3 DataFrames should be merged together as one. The second thing is, the dog stages have three columns and it should be melted into one. The third and last thing is that each of the image prediction and the confidence level has three different columns and the analysis would make more sense if we had only one for each.

After finishing assessing, the next step is cleaning the data.

Data Cleaning:

Firstly, I made copies of each DataFrame using the .copy() method, just to make sure if something wrong happened I can go back to my original DataFrame again. This was my findings:

Quality issues:

1. Twitter Archive Dataset

1. Keeping original ratings only (no retweets) that have images
2. in_reply_to_status_id, in_reply_to_user_id, retweeted_status_id, retweeted_status_user_id and retweeted_status_timestamp are not necessary for the analysis, so they should be dropped.
3. The type of the timestamp should be datatime not object
4. The rating_denominator column has some invalid values
5. Some dogs have wrong names like: a, an, they ..., others have None values
6. tweeted_id should be converted to string

rating_numerator and rating_denominator should be converted to float.

2. Tweet Image Prediction

7. tweet_id should be converted to a string

3. Twitter API & JSON

8. Dropping retweets and keeping original tweets only.

Tidiness:

9. Melting dog stages into one column
10. Merging the 3 datasets
11. Creating a column for the image prediction and a column for the confidence level

For every issue note with the data, I followed the most frequent way of cleaning. Firstly, I define every problem, then I code to solve it and after this, I test my code to check if the problem was solved or not. I kept repeating this until I fixed all the noted issues with the data.

Storing Data:

The last step of wrangling was to save the merged DataFrame into a CSV file named `twitter_archive_master.csv` using .to_csv method.