WeRateDogs - Wrangle report

Here I outlined all the wrangling steps I took through the analysis of the WeRateDogs Twitter archive data.

Gathering Data

Data was gathered from three different sources individually and stored before wrangling. The datasets include:

1) **Twitter Archive Data** : This was already available as a CSV file which I downloaded for analysis manually from Udacity

2) **The Image prediction file :** The image prediction file was obtained programmatically by writing the .tsv file from Udacity into a new file/csv document for wrangling.

3) **Twitter API / Json:** I could not obtain data via the twitter API as my request was not approved by Twitter. However, I downloaded the alternative json file and utilized it for the analysis.

All three data were loaded in the jupyter notebook as ***df_archive, img_files,*** *and* ***api_json.***

Assessment and Cleaning

The Twitter Archive Data: I explored this dataset checking for data types of each column, null/missing rows, cleanliness of data and observed some issues with the quality and tidiness.

- The **timestamp** column had an incorrect data type format and was converted to datetime data type
- Rows with non-null values in i**n_reply_to_status_id, in_reply_to_user_id, retweeted_status_id, retweeted_status_user_id, retweeted_status_timestamp** were dropped.
- Columns containing retweets and replies were dropped based on the project requirements for only original tweets. They are: **in_reply_to_status_id, in_reply_to_user_id, retweeted_status_id, retweeted_status_user_id, retweeted_status_timestamp.** Rows with missing values in these columns were also dropped
- The **rating_numerator** had ratings with different distributions of numbers. Using a plot, I saw the distribution was between 0-14 and the rest as outliers. Rows with values greater than 14 were dropped.

- The **name** column had a lot of ambiguous names. The wrong entries were characterized by being conjunctions and prepositions that do not begin with a capital letter. They were removed based on this condition.

- The **Source** column had html tags which made the data untidy. Using an extract method I was able to remove all the special characters so we had only the text to show visitors devices.

The Image Prediction Data: I explored this dataset and noticed mainly tidiness issues. There were multiple columns for predicting if an image is a dog, the dog breed, and the confidence level.

- The columns containing predictions whether an image is a dog ***p1_dog, p2_dog, p3_dog*** were used to select the condition to collapse the other columns to one.
- The cat_breed for the category of breed, and conf_value were created from the merge of the initial columns.
- The previous columns *(**'p1','p2', 'p3', 'p1_conf', 'p2_conf', 'p3_conf', 'p1_dog', 'p2_dog', 'p3_dog')*** were dropped including **'jpg_url', 'img_num'**

The API/Json Data: I explored this dataset and noticed mainly tidiness issues. There were unnecessary columns.

- I dropped all these columns leaving behind **id, retweet_counts, favorite_counts.**
- I renamed the **id** column to **tweet_id** to enable me to merge the data with the other files.

TIDINESS:

- Separate columns of dog stage (*puppo, pupper, floofer and doggo*) were collapsed into the **dog_category** column and the former were dropped.
- All cleaned files for each datasets were merged.

The cleaned files were saved in separate CSV files as ***df_archive_master, img_files_master,*** *and* ***api_json_master.***

The final file from merging the cleaned copies of each dataset **(final_archive_data**) was used in exploratory analysis.