Wrangle Report WeRateDogs Project

by Sara Vicente

By using Python and its libraries, in this project, I will:

- **Gather** data from three sources.
- **Assess** its quality and tidiness.
- **Clean** the data.

Gathering Data

The datasets I will use are the ones described below:

1. First, I will gather data from a **Twitter Archive** of the famous Twitter account **@dog_rates (WeRateDogs)**, in which users leave comments rating their dogs adding a funny comment to the rating. It is noteworthy that usually, the ratings have a denominator of 10 with numerators greater than this number, for example: 12/10.

2. Then, we will collect the **number of retweets** and the **number of favorites per tweet**, since this information is missing in our first dataset. I will obtain this information using **Tweepy** to query **Twitter's API.ç**

3. The last dataset is contributed by Udacity, since the course instructor collected **images from the WeRateDogs archive** and **classified** them **by breed through neural networks.** Thus, I obtain a table with the predictions of each dog image, the tweet ID, the image URL and the image number that corresponds to the safest prediction.

Assessing Data

Data Inclusion Criteria

In order to analyze the data, I followed the criteria found below:

- Retweets must not be included
- Replies must not be included

QUALITY ISSUES

The four main data quality dimensions are:

- Completeness: missing data
- Validity: if the data make sense
- Accuracy: inaccurate data (wrong data can still show up as valid)
- Consistency: standardization

DataFrame: df_twitter

- **Q1.** Text column contains truncated text instead of displayable text.
- **Q2.** The data contains retweets and replies and I only want to investigate original tweet's ratings.
- **Q3.** There are some columns in this dataframe that I won't be using for my analisys.
- **Q4.** There are some invalid names such as 'a', 'actually', 'all', 'an', 'by', etc.
- **Q5.** Erroneous datatypes: tweet_id, timestamp.
- **Q6.** Some records have more than one dog stage, due to detection errors and the fact that some tweets rate 2 dogs.
- **Q7.** For several columns, nulls are marked as None.
- **Q8.** There are several tweets with a denominator different to 10, might be related to more than just 1 dog per tweet.
- **Q9.** There are several tweets with really high values for the numerators (e.g.: from 100 to 1776) and numerators with decimals.

DataFrame: df_img_pred

- **Q5.** Erroneous datatypes: tweet_id.
- **Q10.** There are some tweet_ids with the same jpg_url.
- Missing values from images dataset (there are 2075 records in this DataFrame compared to the 2356 records in the df_twitter archive DataFrame)

DataFrame: df_twitter_counts

- There are some missing values (there are 2354 records in this DataFrame compared to the 2356 records in the df_twitter archive DataFrame)

TIDINESS ISSUES

DataFrame: df_twitter

- **T1.** The Dog Stage variable is splitted into 4 different columns (doggo, floofer, pupper, puppo). Therefore, it is necessary to melt all the four stages of dogs into only one column.

DataFrame: df_img_pred

- **T2.** The dog breed prediction and the prediction confidence columns in the Image Predictions DataFrame (df_img_pred) should be packed into two unique different columns: breed_pred and confidence_pred.

DataFrame: df_twitter_counts

- **T3.** It is necesary to merge 'df_twitter_counts' into 'df_twitter'.

CLEANING DATA

DEFINE, CODE & TEST

Q1: In order to be able to analyze the text to get names, breed and rating information, I need to display the full text. I will do it by applying the code line that can be found in the code cell below back where I gathered the Twitter Archive.

Q2: Delete retweets and replies in df_twitter table filtering the NaN of retweeted_status_user_id in order to keep only the original tweets.

Q3: I am droping the columns that I will not be using in my analysis from df_twitter dataframe: 'in_reply_to_status_id', 'in_reply_to_user_id', 'source', 'retweeted_status_id','retweeted_status_user_id', 'retweeted_status_timestamp', and 'expanded_urls'.

Q4: To correct invalid names (having filtered by names starting with lowercase), in some tweet's, the real name appears after phrases such as: 'named', 'That is' and 'Name is'. I am going to use the if .. elif .. else statement to try extracting the dog's names from the text column and in case it is not possible, return NaN.

Q5: Convert tweet_id to string and timestamp to datetime data type. (I found more datatype errors but they have been corrected in other cleaning steps).

Q6: While assesing I detected tweets that were rating two dogs, I am going to delete those as I only want to investigate tweets rating one dog. I am also going to correct the dog stages for those tweets with two stages but only rating one dog.

Q7: The missings for name column in archive table (twitter_clean) are represented as "None" instead of Null, therefore I will replace this missing data from None to NaN using .replace function. The missings for doggo, floofer, pupper, and puppo columns in archive table are represented as "None". I will replace those "None" for '' to be able to concatenate the four columns (in further cleaning steps) getting one with the dog's stage.

Q8: By visually inspecting tweets with a rating denominator other than the standard metric (10), I can determine, using the text and the urls, the actual denominator (as well as the numerator). As mentioned above, I just want to analyze tweets that rate one dog, therefore I will remove rows for which the tweet rates more than one dog.

Q9: In WeRateDogs it is accepted that the numerators can be greater than 10, however I will drop those numerators greater than 20 since they do not represent the norm and can be considered as spikes that would skew the sample. I will also correct the values misinterpreted by the algorithm as they are a decimal number.

Q10: Delete duplicated urls from img_pred_clean

T1: Tansform the 4 columns into one by concatenating them and obtaining the dog_stage column with the stege of each dog.

T2: Create a new DataFrame that contains each stub name as a variable, with new index (i, j) using the pandas.wide_to_long function. Each row of these wide variables are assumed to be uniquely identified by i ('tweet_id', 'jpg_url', 'img_num')

T3: Merge twitter_clean with twitter_counts_clean using only keys from twitter_clean with 'tweet_id' as the column level names to join on

STORING DATA

Once the wrangling process is finished and I have my data ready to be analyzed, I storage it into csv files, these two new dataframes are:

- twitter_complete.csv
- img_pred_clean.csv

DATA ANALYSIS & CONCLUSIONS

More information in act_report.pdf

SOURCES

The sources for the analysis are:

- https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.reset_index.html
- https://pandas.pydata.org/pandas-docs/stable/user_guide/merging.html
- https://www.w3schools.com/python/python_lambda.asp
- https://www.geeksforgeeks.org/filter-in-python/
- https://www.dataquest.io/blog/python-datetime-tutorial/
- https://realpython.com/python-matplotlib-guide/
- https://matplotlib.org/3.1.0/gallery/lines_bars_and_markers/categorical_variables.html#sphx-glr-gallery-lines-bars-and-markers-categorical-variables-py
- https://medium.com/towards-artificial-intelligence/matplotlib-complete-beginners-guide-to-line-plots-a436e18d69e4
- https://seaborn.pydata.org/generated/seaborn.jointplot.html#seaborn.jointplot
- https://stackoverflow.com/questions/7391945/how-do-i-read-image-data-from-a-url-in-python