WeRateDogs Dataset

Data Wrangling Report

The dataset for this project 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."

This report documents the wrangle efforts put into preparing the dataset for analysis. The data wrangling procedure includes the following steps:

1. Data Gathering
2. Data Assessment
3. Data Cleaning

1. Data Gathering

In this project, data was gathered from three sources in three different formats. The data gathered are as follows:

a. WeRateDogs Twitter Archive: This file was provided in CSV format, which was downloaded manually from the Udacity classroom.

b. Image Predictions: This file is a TSV file that is hosted on Udacity server. The file was downloaded programmatically using *requests* package and stored in my working directory from where the file was loaded into the notebook.

c. Tweets: To retrieve tweets from Twitter API, the Tweepy package was used to get the status of each tweet by passing the *tweet_id* as a parameter to get_status method. The *json* package was used to read the response, which was in JSON format and the response was writing to a text file tweet_json.txt.

   To access the Twitter API, I created a developer account on Twitter and requested for access which was approved in less than 30 mins. I also timed how long it took to retrieve all the tweets, which was about 36 mins including the waiting time due rate limiting.

   Finally, the text file was read and the retweet_count and favourite_count were retrieve for each available tweet. Note some tweet were not available.

2. Data Assessment

In this step, I assessed the data visually and programmatically. I used Microsoft Office Excel application for the visual assessment while pandas DataFrame methods were used to assess the data programmatically, including head(), info(), describe(). Furthermore, I used value_counts() and duplicated() pandas Series methods for further programmatically assessment. The following data quality and tidiness issues were identified:

Quality Issues

twitter_archive Table:¶

i. Only original rating with images are needed but the table includes retweets and replies ratings.
ii. The following columns: in_reply_to_status_id, in_reply_to_user_id, retweeted_status_id, retweeted_status_user_id, and retweeted_status_timestamp are not needed for the analysis.
iii. timestamp column datatype is string instead of datetime.
iv. The datatype for the rating_numerator and the rating_denominator is int instead of float.
v. Ratings with decimal points are presented wrongly e.g. 11.6/10 => 6/10 was recorded
vi. Wrong rating for rows with multiple rating like text e.g. 24/7 ... 12/10 => 24/7 was recorded instead of 12/10
vii. The source column contains html markup

image_predictions Table:¶

viii. Only interested in the highest confidence level p1 that are dogs
ix. Inconsistent captilization in the p1 => breed column

Tidiness Issues

x. The doggo, floofer, pupper, puppo columns in the twitter_archive represent different dog stages and should be in a single column instead of in four different columns
xi. The three tables should be combined into on table

3. Data Cleaning

In this step, I created a copy of the three data frames and all identify quality and tidiness issues were resolved using the Define-Code-Test framework. Describe briefly below is the action taken to resolve each issue.

Quality Issues

twitter_archive Table:¶

i. Dropped retweets, replies and rows without an image.
ii. Dropped unwanted columns.
iii. Converted the datatype of timestamp column to datetime.
iv. Converted the datatype of the rating_numerator and the rating_denominator to float.
v. Corrected ratings for rows with fractional rating numerator.
vi. Corrected rating for rows with multiple rating like text
vii. Extracted only the content of the anchor tag in the source column.

image_predictions Table:¶

viii. Filtered for only rows which are dogs in the p1 predictions.
ix. Updated capitalisation of the p1 column to first letter capital using the capitalize method of the pandas Series.

Tidiness Issues

x. Combined the doggo, floofer, pupper, puppo columns in one column using the melt method. Afterwards the variable column was dropped and the value column sorted and duplicate removed. (Note dogs with multiple dog stages, only one was selected as the dog stage).
xi. The three tables were merged into one table using the DataFrame merge method.

In conclusion, the cleaned DataFrame was stored in a CSV file as twitter_archive_master.csv, which was now ready for analysis.