We Rate Dogs Project: Data Wrangling

Table of Contents

Data Gathering

The data used for analysis was gathered from 3 different sources using different techniques.

- The Enhanced Twitter data was read directly using the pandas read_csv function. The data was provided and downloaded manually.
- The Image Prediction Tab Separated file. This file was downloaded programmatically using the requests library. It was also read using the read_csv function with the delimited specified as tab.
- The third dataset, tweet_json.txt was collected from Twitter using Tweepy. The data collected using the tweet id in the enhanced twitter data into the text file. The text data was read, line by line into a json file which was then converted to a dataframe, json_df

The Enhanced Twitter data was used as the main source of data. In addition, 2 columns: Retweet Counts and Favorite Count were added from Twitter data, json_df. From the Image Prediction data, the predicted dog breed was also added after wrangling.

Accessing Data

The data was explored visually and programmatically for possible issues/inconsistencies in the data. Upon accessing the data, some quality issues and tidiness issues were found and they are:

1. The timestamp column is not in the correct format
2. There are some invalid names in the dataset
3. The source column is enclosed in html tags
4. The data should only contain tweets not retweets/replies
5. All Retweet and Reply related columns were dropped because they had null values all-through
6. The expanded urls containing the links to the urls had 3 missing values
7. The numerator and denominator ratings should be in one column – rating
8. The doggo, flopper etc. dog stages should not be in separate columns.
9. The retweeted and favorite columns should be added to the dataset
10. The image prediction data had some predictions that were not classified as dogs.

All the issues identified above (Quality and Tidiness) were cleaned in the data cleaning phase.

Data Cleaning

The data cleaning stage consists of 3 sub-stages: Define, Code and Test. For each of the issues identified above, the Define-Code-Test method was used as a procedure for cleaning the dataset.

Quality Issue 1: The timestamp column is not in the correct format.

- Define: Extract Year, Month Day from Time Stamp and Convert it to Datetime from object.
- Code: The extraction can be done using regular expressions and conversion using python datetime method.
- Test: Upon cleansing, the first few rows of the column is checked.

Quality Issue 2: There are some invalid names in the dataset.

- Define: The 'A' and 'None' names should be converted to N/A
- Code: The column is explored and then replaced from None, A to N/A
- Test: The unique values are explored to see if changes are made.

Quality Issue 3: The source column is enclosed in html tags.

- Define: The source of the tweet is enclosed with html tags.
- Code: String Operations are done to extract the text.
- Test: The first few rows of the edited data is checked.

Quality Issue 4: The data should not contain retweets/replies.

- Define: Remove the rows containing retweets or replies from the archive dataset
- Code: Removal can be done using the drop and notna methods
- Test: The size of the dataframe is checked.

Quality Issue 5: All Retweet and Reply related columns are dropped.

- Define: Remove the columns relating to retweet and reply
- Code: Using dropna() function in the column axis, all retweet and reply related columns are dropped.
- Test: Check the first few rows of the dataframe.

Quality Issue 6: The Expanded urls contains 3 missing values.

- Define: The missing values in the expanded urls column should be removed
- Code: The dropna method is used to remove the missing values
- Test: The isnull() method is used to check if there are missing values in the column.

Quality Issue 7: The Retweeted and Favourite columns should be added to the dataset.

- Define: The favorite and retweeted columns would be combined to twitter_archive data on tweet id
- Code: We could combine the two dataframes using the pd.merge function
- Test: The first few rows of the dataset are explored.

Quality Issue 8: Some images were not correctly classified as dogs.

- Define: The breedPredict is created to contain the predictions of dog breeds in the imagePrediction Data. Since some dog breed predictions are not correct, p1, p2 and p3 are checked for a better prediction.

- Code: By applying conditional statements in the imagePrediction dataframe, the correct dog predictions are selected and a missing value(nan) returned when none of the predictions are classified as dogs.
- Test: The first five rows of the breedPredict series is explored.

Tidiness Issue 1: Combine Numerator Rating and Denominator Rating to one Rating column

- Define: Create a column rating to contain the rating for each dog
- Code: Combine the numerator and denominator rating a single column over 10
- Test: Check the first few rows of the rating column.

Tidiness Issue 2: The dog stages are spread across several columns.

- Define: The dog stages should be combined to one column
- Code: A function is created that replaces 'None' strings with an empty string in each of the columns. The strings are then concatenated. For stages with more than one dog stages, one is selected. The empty strings are replaced with missing values.
- Test: The code is tested by exploring the stage column.

Data Storing

The data collected from the 3 sources specified above were combined into one dataframe and saved as a CSV file named: *“twitter_archive_master.csv”*