We Rate Dogs: wrangle report

Natalia Bobkova

January 2022

1 Overview

In this project we worked with data coming from 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.

The goal of this project is to analyze WeRateDogs data and provide interesting insights and visualizations.

2 Data wrangling

In this section we will briefly describe wrangling efforts in gathering, assessing and cleaning data.

2.1 Gathering

There are 3 data sources used for this project:

1. On hand Twitter archive

   It is a Twitter archive with an extensive use of twitter text to extract rating, dog name, and dog "stage" (i.e. doggo, floofer, pupper, and puppo).

2. Additional data via Twitter API

   We can complete Twitter archive information with retweet count and favorite count. We successfully gathered this data from Twitter's API, based on tweets ID available in the Twitter archive.

3. A file with image predictions available online

   The last data set was collected programmatically from Udacity and contains predictions on dogs breeds. Every image in the WeRateDogs Twitter archive was run through a neural network that can classify breeds of dogs. This third data set contains the top three predictions.

2.2 Assessing

All the data sets were assessed both visually and programmatically for quality and tidiness issues.

Programmatic assessment concerned summary statistics, overview of data types and missing values, the presence of duplicates in data (with `info()`, `describe()` and `duplicated()` methods). The following 10 issues were detected across the 3 data sets, 8 for quality and 2 for tidiness:

Quality

Twitter archive

- "None" is not a `NaN` data type in `doggo`, `floofer`, `pupper`, `puppo` columns
- `in_reply` and `retweeted` (with all variations) data are not useful since we are only interested in original tweets
- `timestamp` column is not a `datetime` data type
- invalid values in nominator and denominator columns (0)

Twitter data from API

- `id` column name is not valid to link this dataframe to 2 others (it's `tweet_id` in other tables)
- `in_reply` and `is_quote` (with all other variations) data are not useful since we are only interested in original tweets

Predictions data

- some predicted dog breeds are capitalized, others not
- non explicit column names for predictions

Tidiness

- `doggo`, `floofer`, `pupper`, `puppo` are in separate columns for no particular reason in Twitter archive
- all the three dataframes can be merged on `id_tweet`, after dropping non relevant columns

2.3 Cleaning

First, all the three data sets were copied in order to avoid any inference on original data.

We proceeded with data cleaning. We dropped data based on slicing (for example, to remove rows with 0 in numerator and denominator columns). The `drop()` method was used to remove all data rows that concern retweet and reply data, as well as the columns where this information is specified.

We performed some replacements (`replace()` method), either on column names, or on values (replace a string with `np.nan`). To modify an entire column data type (for `timestamp`) we used `to_datetime()` method.

Slicing also allowed as to isolate data on different dog types and create a separate `dog_class` column.

In order to have only lowercased dogs breeds we used `str.lower()` method.

To test the results of each step we used either `info()` method (renaming columns, dropping rows and/or columns), or `describe()` method (to assess the modification in statistics after certain values were dropped).

Finally, `merge()` was used to have a final data frame, all the three data sets were merged on `tweet_id` column values.

3 Conclusion

The issues identified above are non exhaustive. In fact, during data analysis and visualization process we found out that several issues were left unhandled: for instance, not all the tweets have dogs pictures. Another issue concerns the extraction of dog rates from twitter text: for some reasons, only decimals were extracted but not the integer part (9.74 turned into 74).

Data assessment and cleaning is an iterative process, several iterations are needed in order to identify and fix potentially major issues.