Data Wrangling with WeRateDogs

This report details the data wrangling process of various sources of data concerning the contents of WeRateDogs tweets. The sources included a local csv file, a hosted tsv file, and data retrieved via API.

The wrangling of the data was comprised of the removal of rows and columns that did not serve the purpose of exploring original dog ratings. For example, some ratings were not original tweets but just retweets; and, as found by the image classifier, it seems that not all images were images of dogs.

Gathering Data

The data gathering process for the archive file and the image predictions file were relatively straight-forward, given that mainly the files just needed to be retrieved and written to a DataFrame. As the image predictions file was hosted on a Udacity server, the file was retrieved via HTTP request.

Gathering via the Twitter API was more manually intensive, as a separate API call was needed for every tweet, the JSON object needed to be written to its own line in a text file, and then the text file needed to be parsed for the desired fields to be stored in a DataFrame.

Assessing Data

The messiness and tidiness issues identified were mainly found with the twitter archive file. The image predictions DataFrame did seem to include rows that did not identify any dogs and some duplicate URLs.

Twitter Archive

The main issues concerned the presence of replies and retweets, with replies and retweets for rows and several columns specifically concerning reply and retweet metadata.

Additionally, there were invalid rating scores, invalid names, and untidy dog stages.

Image Predictions

The image prediction file contained duplicate URLs.

Additionally, the file included rows that the classifier deemed to not contain dogs.

Also, depending on which column was or columns were used to filter out non-dog rows, then perhaps some of the other columns would be not be needed.

Twitter API

This DataFrame was generally clean and tidy.

Cleaning Data

Cleaning the data mainly involved addressing each issue identified in the assessment.

Twitter Archive

The reply and retweet columns were used to filter for drop replies and retweets, before the columns themselves were dropped.

For the ratings scores, I re-extracted the ratings to pull floats, accounting for periods and ellipses so they did not invalidate the scores.

For the names, I found that uppercase values were valid names and lowercase values were not (e.g. “a”, “an”). I re-extracted the names, looking for capitalized words following “is” or “named”.

I melted the dog stages columns to reflect a single categorical variable.

I also converted IDs to strings and timestamps to timestamp objects.

Image Predictions

I first dropped duplicate URLs.

Then, given that I wanted to be confident in the classifications, I set the p1_dog column as a filter for rows where I’d trust a dog classification. I then dropped all columns, except for the ID, jpg_url, and p1 column (renamed to “breed), as I felt that filtering on p1 rendered the image classifier fields irrelevant.

Twitter API

All I did here was rename the “id” column to “tweet_id” and converting to string for consistency among the DataFrames.

Joining and Storing the Data

With the irrelevant columns dropped in the twitter archive and image predictions tables, the remaining columns were pertinent to a potential exploration of original dog ratings. Given the potential multivariate explorations involving fields from multiple tables, and that overall, not many columns remained in each of the tables, I joined each of the tables into a master DataFrame, as I felt that was the setup most amenable for exploration.