WeRateDogs Wrangling

1 Introduction

Data wrangling for this project consisted of:

- Gathering data
- Assessing data
- Cleaning data

2 Gathering

Data was gathered from the three sources outlined below:

1. The WeRateDogs Twitter archive. This file was considered “on hand” and was loaded directly into a pandas DataFrame. You can download this file manually by clicking the following link: `twitter_archive_enhanced.csv`
2. The tweet image predictions, i.e., what breed of dog (or other object, animal, etc.) is present in each tweet according to a neural network. This file (`image_predictions.tsv`) was hosted on Udacity's servers and was downloaded programmatically using the Requests library and the following URL: https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-predictions/image-predictions.tsv
3. Using the tweet IDs in the WeRateDogs Twitter archive, I queried the Twitter API for each tweet's JSON data using Python's Tweepy library and stored each tweet's JSON data in a file called `tweet_json.txt` file. Each tweet's JSON data was written to its own line on the .txt file. The .txt file was then read line by line into a pandas DataFrame with each tweet's retweet count, favorite count, and tweet length.

3 Assessing

Visual & programmatic assessments of the data’s quality & tidiness were performed prior to cleaning. The data was first assessed visually to see what types of formatting issues might exist. Then it was assessed programmatically for other quality issues such as **completeness, validity, accuracy, and consistency**. While performing these assessments, notes on the data’s tidiness were taken as well. For data to be considered tidy, **each variable must form a column, each observation must form a row, and each type of observational unit must form a table**. The following issues were found in the dataset:

3.1 Quality

3.1.1 *df_twitter_archive*

- 183 of the tweets are retweets.
- `rating_numerator` and `rating_denominator` are sometimes wrong.
- Remove ratings for non-dog tweets and correct numerators that have errors.
- `in_reply_to_status_id`, `in_reply_to_user_id`, `retweeted_status_id` & `retweeted_status_user_id` should be formatted as strings to fix scientific notation.
- `timestamp` is formatted as strings.
- 109 misidentified names & 745 "None" string values in `name` column. "O" should be "O'Malley". "None" strings should be None type.
- `source` column contains HTML tags.
- Only 2075 `tweet_id`'s had records in `df_image_predictions`. The 281 missing from `df_image_predictions` should be dropped.
- Make `timestamp` display in Eastern Time since that's where the account owner lives.

3.1.2 *df_image_predictions*

- Images with 'False' values for all 3 of `p1_dog`, `p2_dog` & `p3_dog` are unlikely to contain images of dogs.
- Multi-word predictions in columns `p1`, `p2` & `p3` use '_' instead of spaces.

3.2 Tidiness

- All DataFrames should be inner joined on `tweet_id`.
- `doggo`, `floofer`, `pupper` & `puppo` columns should be a single `dogscription` column.
- Retweet related columns can be removed after retweet records are removed.

4 Cleaning

Prior to cleaning, copies were made of all DataFrames so that any mistakes made while cleaning could be reversed. Each issue was cleaned in 3 parts; **Define, Code, and Test**. In the **Define** section, a brief description of the issue was given and how it would be solved. The **Code** section is where code was written to solve the issue. After running the code, the data was then tested in the **Test** section to make sure the code did what it was supposed to do.

Tidiness issues were tackled first to make working with the data easier. Since all data was related to the same tweets, an inner join was performed to form a single DataFrame. After handling tidiness issues, quality issues were tackled next. The assessing & cleaning steps were revisited multiple times while working with the data. Once everything was clean, the cleaned data was saved to both .csv & database files.