

Data Wrangling Report

1. Gathering Data

The dataset that will be used wrangling are tweet archive of Twitter user @dog_rates (https://twitter.com/dog_rates), also known as WeRateDogs. This archive/dataset consists of 2356 basic tweet data from November, 2015 to August, 2017. WeRateDogs is a Twitter account that rates people's dogs with a humorous comment about the dog.

Based on the images in the above dataset (i.e. WeRateDogs Twitter archive), another dataset is created which consists of image predictions (the top three only) alongside each tweet ID, image URL, and the image number that corresponded to the most confident prediction (numbered 1 to 4 since tweets can have up to four images). Though no wrangling will be done directly on this image predictions dataset, it will definitely provide some additional data for our main tweet archive dataset.

Gather Twitter archive CSV file

Using the link provided by Udacity, I downloaded the WeRateDogs Twitter archive manually as `twitter_archive_enhanced.csv` (https://d17h27t6h515a5.cloudfront.net/topher/2017/August/59a4e958_twitter-archiveenhanced/twitter-archive-enhanced.csv) file and imported this file into a dataframe (`tac_df`).

Gather tweet image predictions

I downloaded the tweet image predictions file hosted on Udacity's servers programmatically using Python's Requests library and saved it locally to `image_predictions.tsv` file. Then, I imported this file into a Python Pandas dataframe (`img_df`).

Gather data from Twitter API

Using the tweet IDs in the Twitter archive, I accessed the entire data for every tweet from Twitter API and stored every tweet's entire set of JSON data in a file called `tweet_json.txt` file. Created a dataframe `status_df` from this JSON including only `tweet_id`, `retweet_count`, `favorite_count` and `display_text_range` data.

2. Assessing Data

First of all, I was able to identify 2 quality issues just by going through the Key Points in the Project Motivation page.

Visual Assessment

I opened the `twitter_archive_enhanced.csv` and `image_predictions.tsv` in Excel and scrolled through them, looking for quality and tidiness issues. I was able to spot the following **2 quality** and **2 tidiness** issues:

- a. Quality: unnecessary html tags in source column of twitter archive in place of utility name
e.g. [Twitter for iPhone](#)
- b. Quality: text column of twitter archive contains un-truncated text instead of displayable text
- c. Tidiness: doggo, floofer, pupper and puppo columns in tac_df table should be merged into one column named "dog_stage" .
- d. Tidiness: Twitter archive data without any duplicates (i.e. retweets) will have empty retweeted_status_id, retweeted_status_user_id and retweeted_status_timestamp columns, which can be dropped

Programmatic Assessment

I used pandas' info method on tac_df to spot erroneous datatypes and other quality issues. Then I used value_counts method on *rating_numerator*, *rating_denominator* and *name* columns to look up the range of their values and its distribution. Also to verify 1 tidiness issue that I found during the visual assessment, I queried the archive dataframe to see if any of its tweets has more than one dog-stage mentioned. This entire activity helped me to identify the following 7 quality issues.

- a. Dataset contains retweets therefore duplicates
- b. tweet_id(s) of tac_df table are missing in img_df (image predictions)
- c. Erroneous datatypes (timestamp, source, dog stages, tweet_id, in_reply_to_status_id, in_reply_to_user_id)
- d. Incorrect dog name in column "name" instead few names are in "text" columns
- e. Remove extra characters after '&' from "text" column
- f. Rating numerators with decimals not showing full float

The info method on the other 2 dataframes (img_df and status_df) didn't reveal any quality issues. However, after taking a look at the sample of each of these dataframes, I was able to identify the following tidiness issues:

- a. Join 'tweet_info' and 'image_predictions' to 'twitter_archive'

3. Cleaning Data

I created a copy of all the tables and named them twitter_archive_clean for tac_df, image_predictions_clean for img_df and tweet_info_clean for status_df. For each quality/tidiness issue, I performed the programmatic data cleaning process in 3 stages - Define, Code & Test.

4. Storing Data

After the completion of the cleaning process, I stored the twitter_archive_clean DataFrame in twitter_archive_master.csv file.