Wrangle Report

This document describes my wrangling efforts in wrangling WeRateDogs Twitter data.

The Wrangling consists of three steps:

1. Gathering
2. Assessing
3. Cleaning

Gathering

I gathered data from three different sources:

- A file at hand containing basic Twitter data from the archive of WeRateDogs. The data has originally been downloaded programmatically by someone else.
- Twitter API* (Application Programming Interface) where we can access additional data, such are retweet count and favorite count ("likes").
- A file that we programmatically download** from the internet that contains image prediction data (Every image in the WeRateDogs Twitter archive has been run through a neural network that can classify breeds of dogs).

Each of the three was stored in a pandas DataFrame.

*In order to set up the Twitter API connection some prerequisites were needed:

- Set up a Developers Account on Twitter. Needed to get hold of keys and tokens for accessing the API (get authorized).

- Import tweepy (needed to connect to the API from python)

- Import json needed to store the data in a text file and read it back into a pandas DataFrame.

** In order to download programmatically further prerequisites were needed:

-Import os

-Import requests

Assessing

I went on to assess the data for tidiness and quality issues, ie. whether I had messy and/or dirty data. I assessed both visually and programmatically. Among the more glaring issues were:

- The four columns 'doggo', 'floofer', 'pupper' and 'puppo' do not satisfy the first requirement for tidy data, because it is just one variable indicating some stage of being a dog.
- Lots of values in the 'name' column which are not names ('such', 'quite', 'a', 'an', 'the')
- More tables than observational units. My judgement was that both one and two tables would make sense (could have kept image_predictions separate), but decided that it would be best to merge all three into a single pandas DataFrame.

- Some of the ‘rating_numerators’ were very high. Turns out some of the can be explained by the ‘text’ column. Extracting such ratings as 9.75/10, which had been wrongly cropped to 75/10, would be good.

Cleaning

I created copies of each of the three DataFrames with pandas’ .copy() function, then went on to the (programmatic) cleaning process:

1. Define
2. Code
3. Test

I opted to do all three steps for each issue I resolved. (Because there were many issues, this would save me some scrolling).

To clean the glaring issues mentioned earlier, here are some notes on how I did it:

- Doggo/floofer/pupper/puppo. I used the pandas groupby function to create a new column ‘stage’. Then I deleted the four old columns.
- In order to get rid of names that were not names, only the true names had capital letters, so I used df[‘name’].str[0].str.islower() to access the non-names and set them all to NaN.
- In order to merge the three tables, I had to make sure the ‘tweet_id’ were all in the same type (I chose string), and then I merged two of them, and finally the third with the resulting DataFrame.
- For the numerators hidden in the ‘text’ column, I used the str method ‘extract’ to utilize regex syntax and get the nominator values from the text strings. I also changed type from int to float, for the column to hold decimal values.

I concluded the cleaning efforts by saving it as ‘twitter_archive_master.csv’ using pandas function .to_csv.