Wrangle Report on We Rate Dogs Project

In this report, I will elaborate on the steps I took in wrangling the data for the We Rate Dogs Twitter project. In the main, I am mainly commenting on the process used in ‘wrangle\_act.ipynb’, but here I’m describing the general approach used, especially in assessing the data outside of the Jupyter Notebook.

**Gathering Data**

Three DataFrames were used as the data for the project. The first, a .csv archive of tweets from the WeRateDogs Twitter account, was downloaded manually and opened straightfowardly into a Pandas DataFrame. The second was a .tsv file containing information on predicting the content of the first image of the tweets, and that had to be downloaded programatically and then opened with a tab separated modifier into a Pandas DataFrame. The third was additional information scraped from the Twitter API. Unfortunately, 28 tweets were no longer available, but the ones that were still available were first of all put into a txt file in .json format - ‘tweet\_json.txt’, and then read into a Pandas DataFrame line-by-line.

**Assessing Data**

I worked through the three DataFrames I had in turn. For the first, I did a quick look using .sample() and info(), which enabled me to quickly identify two datatype issues. I then opened the ‘twitter-archive-enhanced.csv’ file in Excel, which allowed me to see that there were some issues with dog names, with dogs being called ‘a’, some called ‘None’, and then there were some odd values for the dog ratings, both in the numerator and the denominator. Having seen this, I returned to Jupyter Notebooks and followed things up there more systematically. I determined that the dog name issue could be resolved simply by removing lowercase ‘names’, and that the validity of the ratings could be fixed by making the numerator values stick between 1 and 14, including decimal values, and only including posts where the denominator was 10. Finally, two tidiness issues in this DataFrame became apparent – firstly, the use of 4 columns for whether the tweet described the dog in the photo as a ‘doggo’, ‘floofer’, ‘puppo’, or ‘pupper’, when this should be one categorical variable.

For the image predictions file, a visual inspection in Excel revealed that there was a consistency issue in terms of some results being reported (in the p1, p2, and p3 columns) using titlecase, and others using lowercase. When inspected programatically, it became apparent that pandas DataFrame was treating p1, p2, and p3 as strings, when really these ought to be category data since the neural network algorithm best matches the image to a pre-determined set of possible categories.

Then, because we wish to analyse this data, it also needed to be merged with the ‘twitter-archive-enhanced.csv’ file. This is a tidiness issue because all of the data belongs in the same container, i.e. information about that tweet. A similar argument goes for merging the information from the Twitter API. Information on retweets and favourites is just information on the tweets made, and belongs in the same table.

**Cleaning Data**

It’s best to look at the code directly here to see how I cleaned the data, but in general I found that tidying the data resolved many issues. I resolved most issues by removing rows that proved troublesome for the eventual analysis, reducing the initial twitter archive of 2356 rows down to 1943.