Wrangling & Analysing Data From WeRateDogs on Twitter

To gather information for this project, the analysis of dog ratings on a Twitter account, I used several different methods. I began by using the pandas package in Python to read a downloaded csv that contained the archive data from WeRateDogs. This includes information such as tweet id, timestamp and the contents of the tweet, which was then saved into my first dataset.

Next, I gathered the data from tweet image prediction using the Requests library. Once I imported Requests in my notebook, I used the url to create a tsv file containing the relevant details about the images in the tweets as well as the systems predictions of what was contained in the picture, three separate guesses for each tweet, the percentage of confidence it had for the guess and whether the prediction was true or false. This file, 'dog_predictions.tsv', became the second dataset for my analysis.

Finally, as a third method for gathering data into a JSON text file, I utilised both the Tweepy library and the Twitter API. (Please note, due to verification issues that I encountered, I did have to use the code provided by Udacity to complete this step of the wrangling process. But, I ensured that I thoroughly read and understood each step of the code.) From all the information gathered during this process, I only retained three columns for my third dataset; tweet id, favourites count and retweets count.

Once I had all the necessary data sorted into three separate tables, I began by visually assessing them. This included looking at the datasets to familiarise myself with the contents and seeing if there were any obvious issues that needed to be addressed throughout the process going forward. Most notably, this step was very useful when it came to finding instances of messy data, tidiness issues within the datasets. Next, to ensure that there were no further potential issues that could cause problems later on in the analysis, I moved on to programmatically analysing the data using pandas methods and queries. This helped me to identify several quality issues that could potentially cause difficulties down the road, a total of 8. I noted both the dirty and messy data so that I could come back and clean them once I was happy with my assessment.

One by one, I cleaned all the issues that I had documented earlier using the code, define and test method. This process included merging all the tables into one easier to analyse dataset, fixing datatypes of columns that were being read wrong by the system, neatening information within the table to make it easier to read and removing things that weren't relevant or necessary to my analysis like retweets or tweets that weren't dog ratings among other things.

To finish off my project, I saved the new master dataset and then began reporting on the insights I discovered throughout the analysis with the use of coding and generating visualisations.