Wrangle Report - We Rate Dogs

Henry Owens
Udacity DAND

This report summarizes my work wrangling the data from We Rate Dogs Twitter account for the DAND project.

Gathering:
According to the instructions, I downloaded the archive csv and loaded it into jupyter notebooks. Then, I used the Requests library to download the image predictions tsv and read that into the notebook. The most interesting part of this stage was using Tweepy to query the twitter API and get more data on the tweets. Per the instructions, I used the Tweet IDs from the archive csv to pull from the API. I used retweet and favorite info.

Assessment:
The assessment stage revealed multiple problems with the archive data: there were quality issues stemming from the extraction process of getting the ratings and the dog stage. Some of the ratings were true to the tweets, but they rated a group of dogs. Seven dogs were rated out of seventy for example. Some of the ratings failed to pull out ratings that were scored as decimals: 11.5/10. I was able to find a few instances of the extraction pulling out what looked like a rating but was not: a dog “robbed a 7/11” according to the tweet. I fixed four ratings manually, after checking some actual tweets. Multiple columns were the wrong type.

The archive data also included We rate dogs retweets and replies. The instructions said to omit the retweets, and after much deliberation I decided to do the same with replies on the basis that the unit of analysis was unsolicited dog ratings. Other twitter users tag We rate dogs in posts of dogs, and they reply with a rating. In some cases though, it was in reply to Snoop Dogg or Linn Manuel Miranda or Jenna Marbles, and it seemed like those were inconsistent with the analysis of simple dog ratings.

There were some tidiness issues with the dog stages being in four separate columns. There were duplicate columns for tweet id in the tweets data frame. The data was also spread across three data frames when it could all be put into one.

Cleaning:

Correcting the data types was the easiest part. To improve the dog stage and rating data values I had to figure out how to extract these from the tweet text. I relied on some posts in the udacity forum for help with regex and str.extract. It picked up a few more accurate rating numerators with decimals and picked up quite a few more dog stage classifications.

It was actually during the cleaning process that I realized some of the ratings were for multiple dogs, that some ratings were actually derived from what looked like a rating but was not, and multiple other issues. Manually checking some of the tweets was helpful to get more familiarity with the content. This was all part of an iterative gather, clean, assess process.