Wrangle_report

April 7, 2019

0.1 Wrangle Report: WeRateDogs Tweet Analysis

The goal of this project was to perform an analysis of WeRateDogs' twitter account by first gathering and cleaning the necessary data. First things first: gathering.

The first part was easy—I just loaded the file that was already on the computer (twitter_archive_enhanced.csv) using pd.read_csv().

For the second dataset (dog predictions), I needed to download a file programmatically from a web server. To do this, I used Python's Requests library. I sent a request (requests.get()) to Udacity's server asking for the desired info, and they sent back a response (the data), which I then saved as a text file to my computer. Then, to open it, I just used pd.read_csv() again.

The last dataset (tweet stats) was the toughest since it required using an API to do a web scrape. Once I requested authorization and got my codes, I set up the request. Then, I made a list of specific tweet ids that I wanted stats for, by subsetting the tweet ids from the twitter_archive_enhanced dataset. Next, I made a data frame to store the tweets in. Then I set up a for loop that ran through the list of tweet ids and pulled only the id, favorite count, and retweet count, then appended them to the data frame I'd made. Tweet ids for any erroneous extractions were printed while the loop was running so I could track the progress/validity of the scrape.

After gathering came assessment and cleaning, which took way longer. I tackled tidiness issues first. For homogeny's sake, I merged the three data frames together. Then, for succinctness, I melted (combined) the dog_type columns into one column, but not before checking out if any dogs were listed in more than one category. 14 were, so I changed their categories to "multiple" AFTER having melted them (it was easier this way). The melting produced a ton of duplicated rows which I got ride of with drop() and a little fiddling. And that was it for tidiness-- not bad at all. Now, quality (colloquially, messiness). First, I removed any tweets that weren't relevant—retweets, replies, or any other originals that weren't rating dogs. Next, I removed irrelevant columns. Then I addressed the unreadable source tags by replacing any extraneous text in the URL strings with nothing. There: easily readable source info. On to non-10 rating denominators. I examined each one that didn't equal 10, found a few that were entered erroneously, and fixed them. The rest I just deleted (it wasn't that many). I did the same thing with numerators, but this time I left the weird looking ones instead of deleting, as ratings outside the 0:10 range are the norm for this account. Next, timestamps. I removed excess info from the strings and converted the column to a datetime variable. While I was at it, I changed the data types for any other columns that weren't what they needed to be. Then, names. I knew from assessment that non-capitalized names were actually just words, so I replaced all of them with "None", for no name. After this I created more descriptive column names. Lastly, I prepped for the questions I wanted answered. I made dummy variables out of any categorical columns I wanted examined with a regression model, and engineered a couple of new ones. Lastly, I used df() to extract individual pieces of time info from the timestamp variable.