Wrangling Efforts

Introduction

This project was designed to help Udacity students master the art of data wrangling. We had to gather, assess, and clean three different types of datasets. We eventually merged all the data together.

This data came from a variety of sources, but the main constant was that the data was from or about the Twitter account WeRateDogs (@dog_rates). This account is a little goofy as it rates dogs on a scale where the numerator is often more than 10 but the denominator is usually 10. There is also a little blurb about the dog as part of the tweet.

Gathering Data

As I mentioned above, there were three datasets which we needed to gather for this project. Here they are in detail below:

- Twitter Archive File: Udacity gave me this file as a csv. I manually downloaded it and uploaded the file to the Jupyter Notebook. This file contained tweet ids and some other information which was useful.
- Image Prediction Files: This file gave us a prediction of what images the twitter account tweeted out. I gathered this file programmatically by using request. This file was hosted on a server by Udacity and I saved it to the Jupyter Notebook directory.
- Tweet Data File: This file was gathered using an API. I had some trouble getting set up with the Twitter Developer process, so I used some code that was given to me by Udacity. This code used tweepy, the Twitter API, and Twitter API credentials to get the JSON data from the tweets. Some of this data was used for my analysis, such as, favorite and retweet counts.

One I gather the three datasets, I read them into pandas DataFrames. I had 3 different types of files. CSV, TSV, and JSON.

Assessing Data

After I read my files into the DataFrames I went through and examined each one to find some messy data and tidy data issues. I used a few different methods to do this.

- Programmatically – I used DataFrame.info() to look at the data types in each column. I had to change some columns from float to string in order to join tables. Another issue I found programmatically was that some denominators less than 10. They should be all be 10 or greater. I used .value_counts() to find this.
- Visually- I used Excel and the .head() function of pandas in the Jupyter Notebook to look at column heads and saw there were some tidiness issue with the ‘dog stage’ indictors. (‘doggo’, floofer, etc…)

While I was visually and programmatically assessing I had the idea of analyzing this data to find the breed, type of rating, and what part of the year the tweet was posted in to maximize engagement with WeRateDog's followers.

Cleaning Data

Before I started cleaning each DataFrame, I created a copy of each so I wouldn't have to regather the data if I messed it all up.

While cleaning my data, I took three steps to ensure I was properly cleaning it. I defined the problem or issue with the data. I then programmatically cleaned the data using regex or pandas. Once I believed I cleaned the data I would test what I just cleaned. This often was looking at value_counts() or .info() to ensure I did what I intended to do. Below are some of the more difficult cleaning methods I used:

- The source column was html. I used regex to remove the tags/link and just leave the text. In general, regex was difficult but Udacity provided us with some documentation and videos in the lesson which helped a lot.
- I had to take out the names which were lower case because if the names were correct, they would be uppercase. I had to use str.isupper().fillna(False) to fill the rows that had the wrong/lower case name.
- I used added the columns doggo, floofer, pupper, and puppo together after I set 'None' to ''. This allowed me to then add a comma in between the different dog stages (if there was one present) so I had one column with readable dog stages.
- Joining all the tables was very enjoyable and satisfying for me. This shows all the hard work payed off and I did it correct.