Wrangle Report

Introduction

The main purpose of this project is to use real world data to wrangle (gather, assess, clean) and then apply analysis with visualizations. The data used was from the Twitter account 'WeRateDogs' (@dog_rates) which "rates people's dogs with a humorous comments about the dog. These ratings almost always have a denominator of 10. The numerators, though? Almost always greater than 10. 11/10, 12/10, 13/10, etc." (Project Overview, Udacity). There are three pieces of data that will be gathered, cleaned, and then merged together to make one final DataFrame to analyze.

Gathering

There are 3 parts to gather for this project. The first is an archived csv ('*twitter-archive-enhanced-2.csv*') file that was provided by Udacity that will need to be downloaded manually and programmatically uploaded into a pandas DataFrame. The next part is to gather tweet count data, which includes, 'retweet count' and the 'favorite count.' This will be done by programmatically downloading from twitter's API using tweepy. This information will be stored in a JSON file, then appended to a dictionary, then loaded into a pandas DataFrame. A twitter developer's profile is needed to access this information. The third and final step is to download programmatically from Udacity's servers, an image prediction's table, created by a neural network, that predicts dog breeds, based on images. A request will be made to get the data from the internet and then saved locally, where it is then loaded into a pandas DataFrame.

Assessing

Initial visual assessment was used to peruse through the data using the **.head()** method, and there were several observations. There were missing values in several of the columns in the **archive table** and the column "floofer" should have been labeled "floof" based on the "Dogtionary." The **image prediction table** had several uniformity issues, such as upper and lowercase letters for dog breeds. The

Next, programmatic assessment was used for the tables. Methods such as the **info()** method showed data types and missing information in each table. The **value_counts()** method was used to show data categories for certain columns to show any glaring issues. In the **archive table** the column *timestamp* needed to be a datetime64 dtype instead of a string. Also, the *tweet_id* was int64 instead of object dtype, which should be a string because you cannot do any mathematical applications on the tweet ids. The *tweet_id* column had the same issue in the **image prediction table**, and the **tweet count** table needed to change the column name *id_str* to *tweet_id* in order for merging properly later on. As far as tidiness issues, all three tables would eventually need to be merged into one in order to properly analyze and the source column in the **archive table** did not reveal anything insightful and cluttered the table when viewed visually.

Cleaning

In order to begin cleaning, each table had a copy made in case the code broke and a fresh clean slate could be started over again. Aslo, the original data is helpful to view incase modifications were made that altered the way the data was interpreted. The **archive table** needed the most help and several methods were used to drop the excess columns that lacked data, and all the retweets and replies were removed as well. Methods such as **dropna()**, **isnotnull()**, **rename(), astype(),** etc were used to accomplish these goals. There are details in the code report. Also each piece that was cleaned in the project was broken up into stages. The first was defining the problem and the way to solve it. Next, was the coding part, then finally the testing part which made sure the codes job performed the intended task.

Finally all the tables were merged into one DataFrame titled the **master** table. These three tables were joined on the *tweet_id* column because that was the common datapiece in each table. After merger, a few extra rows of data were removed to have a complete uniform set. Then the last thing was the file was saved to a csv file (‘master_csv’) and stored locally on the drive.