WeRateDogs Twitter Data Wrangle Process

By Ryan Gee

The necessary datasets for the WeRateDogs analysis were spread across 3 different sources:

1. Archived twitter data was given in the form of a csv file
2. Image prediction data was stored as a tsv on Udacity's server
3. Retweet and Favorite counts resided on Twitter and was extracted via an API

For the Data Wrangle portion of this project, I wanted to practice writing object-oriented code. I started by creating a class called WeRateDogsDataPuller whose main function was to extract the data from the 3 different sources and combine them into one Pandas data frame. The "Archived twitter" data was very straight forward and just utilized the Pandas's read_csv() function to convert the given csv into a Pandas data frame. The "Image Prediction" data was downloaded from Udacity's servers by using Python's built-in requests library. Once downloaded, I used the read_csv() function to convert the tsv file into a data frame. The last dataset, "Retweet and Favorite Counts", was the most involved wrangling task. The first step was to create a developer's account with Twitter, so that I could access their API. Once I gained access to the API, I used the get_status() function in the Tweepy library to pull a JSON for each of the tweet_ids in the "Archived Twitter" dataset. This JSON included a ton of data, but the most important data was the retweet_count and favorite_count data. I created a new class called Tweet that parses and extracts the important data from these JSONs and then writes the data to a new line in a specified text file. Once the text file was fully populated, I created a Pandas data frame by reading each line of the text file. Finally, I used Pandas's merge function to "left join" each of the datasets together. I used a "left join" to keep all of the data even if there was some missing data in the "Image Prediction" or "Retweet and Favorite counts" data.

Overall, this was a comprehensive task that requires knowledge of the Pandas library, Request Library, and API's in order to complete.