Wrangling Report

The Data Wrangling Process

1. Gather

Given that we have the archived Twitter dataset on hand (`twitter_archive_enhanced.csv`), as well as the image predictions dataset (`image_predictions.tsv`), I needed to use this information to access Twitter's data via the Tweepy library. After registering my Twitter application, I was able to use my public and private API keys to programmatically access data using Python. By utilizing the `tweet_id` field, I was able retrieve any piece of information from the tweet in JSON format, namely the `retweets` and `favorites` fields. I placed this information locally in a Python dictionary, and then saved the data into my `tweet_json.txt` file.

2. Assess

Once all the data was gathered, I previewed basic information, using the `.info` method for each dataframe. I looked for wrong data types and formats, missing values, and untidy data. To discover outliers, I also used the `.value_counts` method on predictable fields, such as the `rating` fields, where we expected denominator to be 10 and numerators to be around 10.

3. Clean

After discovering a several fields that had unclean data, I used the Pandas library to make some simple corrections. This included deleting unnecessary data (retweets, replies, and tweets with no photos) and converting data formats for `timestamp` to a DateTime type and `tweet_id` to a String. The largest amount of time and data cleaning analysis was for the rating. There were a couple dozen of outliers for either or both the `rating_numerator` and `rating_denominator`. In some situations, the algorithm picked the number combination, which was not always meant for the score (for example, the first numbers in the tweet referred to a date); this required manually intervention. And in other common situations, the tweet contained a photo of multiple dogs in which case the score was multiplied by a factory of 10, depending on how many dogs were in the photo; this required a simple algorithm to scale the `rating_numerator` by a factor of 10 depending on what the `rating_denominator` was.