Wrangling Efforts Report

In this report I will walk you through my wrangling efforts for the project Wrangle_Act. The process of wrangling the data for my analysis consisted of three parts: gather, assess, and clean.

Included in the assignment were three different resources from which I was to gather data. The first resource was a .csv file given to me that contained Twitter archived data from the WeRateDogs website. Gathering the data from the csv file was straight-forward. I used the Pandas library to read in the file to my Jupyter Notebook using the "_readcsv" method.

The second resource was a .tsv file that contained image data gathered from running the tweet images through a neural network. The data collected was the information created from the neural network including estimated dog breed and accuracy of prediction rates. I obtained this .tsv file programmatically by performing a "get_request" on the Udacity server using the "requests" and "os" libraries. The Udacity server returns the requested files to the assigned variable and from this variable I could write the content to a file that I created.

Finally, the last resource was to be collected by querying the Twitter API using the provided tweet ids from the .csv file given to us. This resource required a little more work to gather the information. For this resource, I had to import the library "Tweepy" to connect to the Twitter API. I went on to Twitter's Developer website and requested special access to their API. After completing a form explaining to Twitter how I would use the information collected from the API, I was granted access. Twitter provided me with consumer_key, consumer_secret, access_token, access_secret numbers to authenticate who I was and to grant me access to the information on the API. I included this information in my code when connecting to the API and was able to gather information for the tweet ids that I was given in the .csv file mentioned earlier. This information included retweet and favorite counts. The information was in json format so I wrote the information being collected for each tweet into a text file. Once all the information was collected, I was able to read the text file with its json format and create a list of dictionaries for each tweet id. Once I had read and copied the information for every tweet id, I created a dataframe from the list of dictionaries. This was the dataframe I was able to use for my analysis.

After the data was gathered, I needed to assess my data. Assessing includes learning more about the data by looking at what is included or not included and by looking for potential issues with quality and tidiness of the data. I looked at my data by opening it up in google sheets to first see if I could find any issues by looking at the data. I then used programmatic methods on the three dataframes (ie. head, tail, info, describe, etc.) to gather more information on the data. In assessing my data, I noticed that all three resources gave me the tweet ids. This was good in case I wanted to merge the information at a later time and keep track of which tweet id the information went with. The three resources also gave me information on where the tweets came from, image information, retweet and favorite counts, dog ratings, timestamps, etc. There

were various quality and cleanliness issues for each resource. In an effort to stay organized, I split them into lists below each resource after it was added to my notebook.

My final step involved looking over my quality and tidiness lists to determine a logical order to be taken to clean and tidy up my data. Once I determined a logical order, I dealt with the issues programmatically.