Wrangling and Analyze Data Project Report.

This report will be divided into 3 parts. With each part briefly describing the steps that I took to gather, assess and finally clean and store that data used for this project.

Gathering data.

The first piece of data to be gathered was the twitter_archive_enhanced.csv file. This contained the Twitter archive data from the WeRateDogs twitter account. This csv file was already provided for this project. So, I just downloaded it from the Udacity classroom and then imported it into the wrangle_act notebook file as tweet_archive dataframe.

The second piece of data was the image_predictions.tsv file containing image predictions. The link to the file hosted on Udacity servers was provided. I used the requests library to download the file programmatically and save it locally. I then imported and read the file into the image_preds dataframe.

The third piece of data was the additional data that was to be queried from the twitter API. Unfortunately, my application for elevated access was denied, so I used the provided tweet_json.txt file. I then opened the tweet_json.txt file within a context manager and looped through each line in the file, used json.loads() to read the file to a python dictionary, accessed the tweet_id, retweet_count, and favorite_count into a list which I then used to make a dataframe called tweet_json.

Assessing data.

I first used visual assessment for each of the above mentioned dataframe to identify some issues. To do this, I used the pandas sample() method. I used other pandas methods to investigate further the issues that I had observed.

I then used programmatic assessment to identify quality issues in the dataframes.The methods I used here are info(), duplicated(), sum(), describe(), value_counts(), loc, head() and many others.. All the quality and tidiness issues found were then noted down in the notebook.

Cleaning data.

Before cleaning any of the dataframes, I made copies of each dataframe. The copies are named tweet_archive_copy, image_preds_copy, tweet_json_copy. I also created a define, code and test cells for each quality and tidiness issue noted.
I then dealt with the quality issues as follows:

I removed the ratings that were from retweets and not tweets.

Then I dropped rows that had more than 50% of their values missing which I could not fill.

Then I dropped all rows that had tweets which had no dog pictures. Some of the rows had erroneous names like ‘the’, ‘actually’, ‘a’ which helped identify them and drop them.

Then, for tweets whose names were erroneously named ‘a’,‘an’, ‘the’ , I listed the indexes for those with no names and those whose names were wrongly captured. I renamed to ‘None’ the pictures with no names and named the rest appropriately.

I then replaced the invalid rating_numerators and rating_denominators with the right values.
Changed the datatype of the timestamp and rating_numerator columns to datetime and float respectively.

Some of the rating_numerators with decimals were not extracted correctly. I filtered out the affected rows and replaced them with the right values.

Created a new dog_stage column and dropped and filled it appropriately then dropped the columns separately indicating the dog stages.

The p1,p2,p3 columns in image_preds dataframe had values starting with lowercase and some starting with uppercase, changed them all to be lowercase.

Created a new 'dog_stage' column and filled it appropriately then dropped the doggo, floofer, pupper, puppo columns

Merged the tweet_json dataframe with the tweet_archive_copy dataframe, then merged the resulting dataframe with the image_preds_copy dataframe. The new dataframe was called tweet_archive_full which was then saved to a csv file called twitter_archive_master.csv.