Shariq Shaikh's Data Wrangling Documentation

The following describes my efforts to collect and ultimately clean a series of dataset containing data on a vast number of tweets. The tweets in question are in relation to a twitter account called "WeRateDogs" which rate peoples' dogs through tweets they post. All data that has been collected and wrangled had been done using python in jupyter notebooks, with the primary module being pandas.

I first started by downloading the primary source of tweet data, provided by Udacity and placing it in an appropriate directory on my machine. From that directory I used pandas 'read_csv' function to load into the notebook using pandas. From there I proceeded to download the tweet image prediction dataset which was also provided by Udacity however here I downloaded it programmatically using a request. I then generated csv programmatically which was placed in the jupyter notebooks corresponding directory. Last I downloaded retweet and favorite counts which I did by using the twitter api called tweepy. The process involved signing up for a twitter developer account, generating a series of access and authorization keys, and using them to programmatically generate status objects which contain large json data. The twitter IDs from the primary dataset were ran through a loop using the api code to gather this json data. The json data for each tweet id was placed in a txt file. The txt file was then used to create a data frame of retweet and favorite counts. The unmodified data frame was then used to generate an csv file for the retweet and favorite counts, a file call 'tweepy_json_df.csv'.

At this point I was ready to start assessing the data for its cleanliness and clean it. I first start by loading and reloading the image predictions and json tweet data and create a copy of all three datasets to perform the cleaning. I then proceeded to assess the twitter archive data. I first asserted that the 78 tweets were retweets which needs to be removed since the only tweets the should be accounted for are original tweets. I then noticed that retweeted_status_id, retweeted_status_user_id, and retweeted_status_timestamp didn't make much sense because based on the json data far more than what was in archive data was re-tweeted and not only that, the retweet info that was shown was only for one retweet. I thought it would be best to remove these along with the in_reply_to_status_id and in_reply_to_user_id since they were both empty and served no purpose. Next, I identified that tweet_ids across all three tables were floats. Since these numerical values are simple labels used to reference to the specific tweet, I believed it would be best to convert these values to strings instead. Following that the source column contained unnecessary html along with the string that indicated what the source of the tweet was from. I used beautiful soup to strip values of the html only leaving the source string. I then realized with so few sources for each tweet I thought I would be best to convert the data type for this column to category. From there I identified the text column also contained the tweet url short hand, so I used regex to separate it out and create its own column for it. In the img_prediction table, the data type for the img_num column was int and should be a string instead so I converted it. I also did the same thing for the archives timestamp column making all the object in it of datetime.

For the tidiness issues with the datasets, in the archive data set I merged the doggo, floofer, pupper, puppo and no dog status into one column and discarded the others. Last I merged all three table together using the tweet_id as the key to join on.