Wrangle Report

GATHER

The gathering phase consisted of reading an archive of tweets in CSV form along with along with the dog breed predictions resulting from applying a neural network algorithm to the images from those tweets. The final and most important data had to be gathered using Twitter's developer API to query for the retweet counts and favorite counts for those tweets. However, I was not able to obtain a developer account due to a malfunction of the Twitter application process. Therefore a copy of the JSON data was obtain in file form. This was read into a dictionary that was keyed on tweet_id to simplify the process of adding the those counts to the original data later in the cleansing phase.

ASSESS

In the assessment phase I observed a couple of columns that contained only 78 values and one date time column that had the incorrect data type. There were also 181 retweets based on the retweeted_status_id. Using the describe method I noticed there were 23 records that did not have valid ratings. Specifically, there were min values of 0 and max values of 170 for the rating denominator. This value was reported to always be 10. Looking at the unique values of the name field I found quite a few, at least 840, invalid names. The only quality issue with image predictions that I found was the inconsistency in the breed names. Some used capitals and different separators were used. There were also quite a few images that were not dogs at all.

CLEAN

To clean the data and produce the single data file named twitter_archive_master.csv I first created copies of the data frames. Then I converted the timestamp column to datetime using the Pandas method. Then removed the 23 records with invalid ratings along with the 181 retweets. After removing those rows, I dropped the retweet as well as reply columns. I fixed the inconsistency in breed names by converting all to lower case and replacing both separators with spaces. To add the retweet and favorite counts I first created the columns preloaded with 0 because there were more tweets in the archive file. That way the missing records would have valid counts of 0. Then I iterated over the tweet_id's in the archive to fetch each count. The final step was to determine the most likely dog breed from the three possible predictions. Do do that I split up the three predictions into separate data frames and then concatenated them after renaming them to have matching columns. Then I simply sorted on the confidence from highest to lowest and eliminated the remaining duplicates while filtering out the images that were not of dogs. Once I had the most likely dog breed, I merged it onto the original archive data frame.