Wrangling the Twitter Archive

By Patrick Maloney

This dataset was very messy and dirty. The enhanced archive was given to me, and then I collected the additional info from the twitter API to get info on favorites and retweets. Image predictions of breeds were downloaded programatically from the udacity servers, utilizing a neural network created in one of the machine learning courses.

An assessment of the dataset revealed many issues. The extraction of information from the text in the enhanced archive had numerous errors, including erroneous scores in both the rating numerators and denominators. This was what I considered to be the most important issue, as these data would be essential to my analysis and would act has my dependent variable in any regression analysis. Thus, I thought it was important to clean these issues by slicing the dataframe to the values that fall within the normal range of ratings and remove all the outliers that could upset the analysis. Other quality issues included needing to convert the datatypes of certain columns to the correct types. Another issue was that the number of observations in the image prediction dataframe, and the number of observations in the tweet archive were not the same. This had to be corrected by making sure to only include tweets that had images, and condensing the archive to the rows common in each dataframe. I also to eliminate any retweets that were not original ratings by the account. This was done by slicing the dataframe to only include tweets where the retweet values were null. Decimal values in the ratings also had to be accounted for by finding decimals in the text and manually correcting the corresponding tweets.

There were many tidiness issues as well. The three dataframes were all observing tweets from the account, so I decided to consolidate everything into a single dataframe. This was done by joining the dataframes on tweet ids. Also, I converted the dog stage columns into a single dog_stage column, rather than have each value listed as a variable, which doesn't conform with tidy data rules. Even though I ultimately undid this by using dummy variables to run a regression in the analysis phase, it is best to have tidy data whenever possible In the image prediction dataframe, I knew I would not use most of the columns, and all I really wanted was the best breed prediction, so I wrote a loop to extract the best prediction into a new column, and then joined that new column on the tweet id to the archive.