We Rate Dogs Wrangle

Description: A look into the tweet data of @dog_rates on Twitter, up until July of 2017. Also included are the results of a neural network created to attempt to identify dog breeds in the photos that accompany the tweets, and an individual tweet's likes and retweet counts.

Data: The data was scrapped from twitter using an API, and a neural network attempting to guess dog breed based on photos attached.

The Data

Twitter_archive:

Tweet_id: an individual ID used to keep track of tweets.
In_reply_status and in_reply_user: Used to keep track of if a tweet was a reply and to who.
Timestamp: time a tweet was sent
Source: how a tweet was sent. I.e "Twitter for iPhone."
Text: The tweet text itself.
Retweeted_status_user_id and retweeted_status_timestamp: when a retweet occurred and by whom.
expanded _urls: the long form url to the tweets.
Rating_numerator and rating_denominator: The bread and butter of the account, the ratings themselves.
Name: The dog's name
Doggo, floofer, pupper, and puppo: The four stages a dog can be, according to @dog_rates.

Image_pred

Tweet_id: the same identification number as the tweets in twitter_archive.
Jpg_url: the urls for the photos
Img_num: the number of images associated with that tweet id.
P1, p2, p3: The first, second and third, respectively, predictions of dog breeds from the photos.
P1_conf, p2_conf, p3_conf: the confidence intervals for the first, second and third guesses of dog breed.
P1_dog, p2_dog, p_3 dog: Whether or not the prediction was true, and thus correct.

Tweet_json

Tweet_id: The same as the previous dataframes.
Retweet_count: the number of retweets for that tweet
Favorite_count: the number of favorites (or likes) of the tweet.

The Wrangle

In a jupyter notebook I read in the .csv files necessary to begin the project. I created the dataframes and looked over them all to visually assess what needed to be fixed in each dataframe and what could possibly be dropped or combined later on down the line. I found incorrect data types in both twitter_archive and image_preds. I also noted that there were a lot of redundant and unnecessary columns. I kept track of all these things in a separate word document and via inline comments.

After this I programmatically assessed the data. I checked for duplicates, misread tweets, and various other issues that could be found in all three dataframes. I also kept track with both comments and a separate document. I made sure to note which tweets I would have to fix manually, and which ones could be fixed with code.

After the assessment was finished I began the wrangling itself. I did this by using the define, code, test method. I did each dataframe separately, keeping track of comments and making sure to document in a separate word file as well. I then created copies of each dataframe so as to not lose the original data.

Starting with twitter_archive I made my way down my list ensuring I kept to the define, code, test parameters. I then moved to image_preds, and finally to twitter_json. I kept image_preds as a separate table as it was easier to read and tidier in my opinion to do so.

After the wrangling was complete I saved masters of the (now two) dataframes and moved on to the actual assessment and visualization portion of the project.

The analysis itself was completed in a separate jupyter notebook file for tidiness. The file is named We_Rate_Dogs_Analysis