Data wrangling WeRateDogs

1. Introduction

The requirement for this assignment was to gather data from three different sources relating to the WeRateDogs Twitter account, assess the data, then clean it, and finally produce visualizations and insights from the resultant product.

As a precursor to this exercise I installed the following modules in my notebook: pandas, numpy, requests, json, and os.

2. Gathering the data

The first set of data to be gathered was The WeRateDogs Twitter archive provided by Udacity as a .csv file twitter_archive_enhanced.csv. I downloaded this and then read this in to create a dataframe called df_csv.

The second set of data to be gathered was that of tweet image predictions of the dog breed produced by running the twitter archive through a neural network. This was contained in a tab delineated file (.tsv) hosted on Udacity's servers that needed to be downloaded programmatically using the Requests library and the following URL: https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-predictions/image-predictions.tsv (https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-predictions/image-predictions.tsv) I requested the file from the provided url, and then read it into a dataframe called df_image_predict.

The third set of data to be collected intended Tweepy to be used to query Twitter's API to source further information relating to the tweets. Despite repeated attempts to be granted access to the data and lengthy response times from Twitter (over two weeks), this was not possible and I had to resort to downloading the tweet-json.txt file provided by Udacity. To create a dataframe for this data, I read in the data and then iterated over it with a for loop to extract the data fields I wanted to include in the dataframe (viz. the tweet_id, favorites_count, retweet_count, and full_text.) I then read into the third dataframe called df_twit.

3. Assessing the data

I then worked through the three different dataframes to assess issues of quality and cleanliness, and detailed problems identified. Visual (see attached xls file twitter_archive_enhanced.xlsx as one example) and programmatic assessment tools were used to do so, with the following list of issues being identified (not all of which were cleaned):

Quality issues:

1. Some records in the expanded urls had two urls.
2. in_reply_to_status, in_reply_to_user_id fields appears floats, and look like they should be integers or possibly strings
3. The numerator_data field has at least one error - line 51 of excel spreadsheet records a score of 5 when the 'text' column shows that the score was 13.5
4. There were some large numerators in the numerator_data column and these would need to reviewed and a decision taken as to how these should be dealt with.
5. The 'source' column had the url beginning <ahref= '
6. Timestamp's data type was an object not date/time
7. There was missing data for in_repy_to_status_id, in_reply_to_user_d, retweeted_status_id, retweeted_status_user_id, retweeted_status_timestamp
8. 'None' was incorrectly classified for the columns doggo, floofer, pupper, and puppo
9. There was missing and incorrect data in the name' column
10. tweet_id was of an integer type but might better be served as a string
11. Some records in the expanded urls have two urls - these need to be separated.
12. The rating_denominator had 23 records that didn't conform to the standard of 10.
13. There were retweets included in the dataset that needed to be excluded

Tidiness issues:

1. Timestamp has date and timestamp in one column – it was decided to split these out into two columns for analytical purposes even though they are of the same variable time.
2. 'Doggo', 'floofer', 'pupper', and 'puppo' all have a separate column, but they are all one variable corresponding to a dogtionary defintion (tidiness issue)
3. Redundant columns needed to be dropped from the image predictions dataframe
4. The three datasets were independent and needed to be merged together

4. Cleaning the data

Finally I iterated through the different quality and tidiness using the define, code, test approach: defining in words what changes needed to be mode, implementing those changes though code syntax, and then testing the code to see that the data had been cleaned.

I concluded the assignment by saving the cleaned data into a master dataframe, which I then wrote to a .csv file.