Data wrangling report

This report briefly describes the data wrangling efforts I performed on WeRateDogs Twitter account data.

DATA GATHERING

In this step, we were required to gather three pieces of data with three different methods, which was a little bit challenging for me but so much rewarding as I gained a strong deep understanding of the importance of the data gathering step and how it can be done in three different ways

1- Directly downloading the WeRateDogs Twitter archive data (twitter_archive_enhanced.csv)

First, we were given a link to download a CSV file containing some basic information about different tweets, this was the easiest and most straightforward gathering method since it was done with one simple line of code using the pd.read_csv() function which basically loads directly any flat file content into a pandas dataframe.

2- Use the Requests library to download the tweet image prediction (image_predictions.tsv)

The second piece of data we were required to gather was hosted on Udacity's servers. We had to download it programmatically using the Requests library, this data contains some cool predictions of different breeds of dogs, and again this was a great opportunity for me to use the Request library and learn how to download files programmatically rather than manually.

3- Using the Tweepy library to query additional data via the Twitter API (tweet_json.txt)

The CSV file we were provided with, contains some very basic information about rating tweets where most of them were extracted from the tweet text, Hence we were required to gather additional data, at least the number of favorites and retweets for each tweet we have in the Twitter archive, the only way to gather this additional data was to deal with twitter's API, this method required much more efforts and time for me than the other methods because I had to understand twitter's JSON object and to spend some time looking over Twitter's official documentation. This was a great opportunity for me to widen my knowledge and get used to interacting with API's.

ASSESSING DATA

After finishing the gathering process, I assessed the data to check for both tidiness and quality issues. To do so I used both visual and programmatic assessment, some of the issues that I discovered with the visual assessment were: the presence of junk tweets that are not ratings and that need to be removed, irrelevant HTML code in the source column, missing values encoded as string 'None' in the name column. Some tidiness issues were also easy to spot with the visual assessment like the dog stage being expressed over four columns instead of one and having pieces of data relating to the same observational unit spread over three tables.

I was able to identify more subtle problems using programmatic assessment: including incorrect data type for id columns and date columns as well as some erroneously extracted ratings and names.

Below is a list of all the issues that were found and resolved:

Quality issues

- Tweets that are not original tweets in `twitter_archive`
- Irrelevant HTML code in source column in `twitter_archive`
- name column has values that are clearly not names in `twitter_archive`
- Dogs having the string 'None' as name in `twitter_archive`
- Invalid ratings in `twitter_archive`
- Wrong dtypes:
  - timestamp in `twitter_archive` should be converted to datetime
  - tweet_id column in `twitter_archive` should be converted to string
  - tweet_id column in `image_predictions` should be converted to string
  - id column in `data_json` should be converted to string
- Some columns' names in `image_predictions` are vague
- Dog breeds in `image_predictions` are not standardized

Tidiness issues

- Dog stages variable in `twitter_archive` should be expressed on one stage column, instead of four
- `image_predictions` dataframe should be combined with `twitter_archive` dataframe
- Retweet_count and favorite_count in `data_json` should be part of `twitter_archive` dataframe

Even though these weren't the only problems in the data, they were sufficient for me to showcase my data wrangling skills and validate the project.

CLEANING DATA

The cleaning process was simple for most of the issues discovered during the assessment, however, some issues required a little more effort to resolve, for instance correcting the tweets with invalid ratings. Rating, as many other variables, was extracted from the tweet text, not having a clear unique pattern to follow during the extraction process caused some dates and other slash-separated numbers to be mistakenly extracted as ratings. To solve this issue, I had to take some time in reading through the regex syntax documentation and learning how to implement it in python, some tweets, however, required further assessment and manual rating extraction.

While performing several cleaning tasks, I learned how to fix multiple data quality and tidiness issues and I've become more acquainted with several python functions.

Following the define-code-test process while cleaning was very helpful and helped me to be more organized and efficient in addressing each problem.