Wrangling on a Dataset Report

Data Analyst Nanondegree, Project 4: Wrangle and Analyze Data

In this report, I will be explaining the steps I took and the decisions I have taken to clean the dataset I was given.
I will divide it into three parts, Gathering, Assessing, then Cleaning.

Tools Used:

- Python programming language
- Jupyter Notebook (IPython)
- Pandas, Requests, and Tweepy Libraries

Gathering:

At first, I was given a .csv file which I seamlessly loaded with pandas into a dataframe called *tw_archive*.
Then I downloaded the second required .tsv file from the given URL using the requests library, to load it again into another dataframe called *tweet_info*.
Lastly, I used the Twitter API through the Tweepy library to get the missing required info. I got the retweet count and favorite count for each tweet, alongside the image URL of the tweet (Where appropriate)
Downloading them took around 15-20 minutes. I stored the data on a new text file (the text itself was formatted in JSON), then I retrieved them into another dataframe called *image_pred*

Assessing:

> At this stage, I'm looking for data quality issues, which could be missing data or incorrect data, and tidiness issues, which are poorly designed tables and datasets.

I started with visual assessment, by exploring each dataframe individually, looking for any unusual or noticeable quality or tidiness issues. I used functions such as, `.head()`, `.tail()`, `.sample(n)`, and whenever needed I examined each row or couple of rows closely using indexing: `.iloc[columns, rows]`

I found some issues quickly such as a redundant '+0000' string appended to each timestamp, and that the 4 columns that define the dog stage better be collapsed into one column - a tidy data issue. Also, I noted that I should merge the original Twitter archive data with the data I gathered using the Twitter API.

While other issues weren't easy to notice like values other than 10 for rating denominator, or and that some names were weird: a, an, the, very.
Those issues required programmatic assessment to confirm, for which I used functions like `.info()`, `column.value_counts()`.
So for example, I used `name.value_counts()` to check how many names were invalid.

Not to forget, to check for datatype issues, I had to use .info() function. I found that most columns had incorrect, or at least inappropriate, datatypes. Like the *id* stored as an **int** instead of a **string**.

In the end, I documented all the issues I noted. Using the following hierarchy:
Quality Issues:
Table 1:
1)..
2)..
Table 2:
1)..
2)..
Tidiness Issues:
1)
2)

Cleaning:

Fixing all the issues I have documented.

I fixed some of the datatypes issues at first, converting every id (like tweet_id) to string datatype. This is important for the next step.
Next is tidiness issues. I prefer to do them as early as possible so that cleaning other quality issues becomes easier.
I merged the tw_archive df with the tweet_info df on the tweet_id, which is why I needed to convert the datatypes as appropriate. Now I had one dataframe that had all the required info of every tweet.

After tidiness issues, I went to fix the other quality issues. Most of them only required few actions to fix: Converting data types, finding and replacing strings, extracting matches with regex expressions, finding and filling NaN values and some simple math to fix rating issues.

Some issues had only one reasonable solution: Dropping the problematic rows. That's because I either couldn't find the missing or correct values or because it was required to remove them.

During the cleaning process, I have discovered new issues and went up to document them. Furthermore, as I start to clean an issue, sometimes I change its definition because I would have edited the data and the data frames. And other times I find that the issue has been resolved on itself. To illustrate, I have documented that I should remove a specific invalid tweet. But then when I arrived at it, I found that the tweet has already been removed by a previous drop of data.