WRANGLING EFFORTS REPORT

In this document, I'll take you through my wrangling efforts. The document will be divided into four headings (gathering, assessing, cleaning, and storing).

Now, let's get started.

GATHERING DATA

The project requires me to gather data from 3 sources, create data frames from each piece of data I gather and merge all the data after they've been assessed and cleaned.

Here are the three data sets I gathered and how I gathered them:

1. ***twitter_archive_enhanced.csv****:* This data was handed to me in the classroom, and I just had to download it manually
2. ***image_predictions.tsv:*** This data set is hosted on Udacity's servers, and I programmatically downloaded it using the ***requests*** library and a file opening context manager.
3. ***tweet_json.txt:*** This data set was gotten from the Twitter API using the ***tweepy*** library. After that, I had to read the text file line by line and extract other relevant data, like the ***retweet_counts*** and ***favorite_count***.

ASSESSING DATA

Now comes the assessment stage.

I opened the files in a spreadsheet package (Excel for the twitter_archive_enhanced.csv file) and a text editor (Notepad for the other datasets). Then, I noted some data quality and tidiness issues.

Next, I use some pandas' methods like ***columns, info, head, describe, dtypes, value_counts, loc***, and other functions to programmatically access the data for issues.

After the programmatic assessment, I made a detailed list (in the .ipynb file) of the data quality and tidiness issues I'll need to clean in the next wrangling phase.

DATA CLEANING

Now, the final data wrangling stage is **cleaning**.

Here's how the cleaning went:

1. First, I made copies of the three data frames I wanted to clean.
2. Then, I dropped the rows with values in the ***retweet_status_id*** (I only need the tweets) column using the ***drop*** function.
3. After that, I dropped the ***retweeted_status_timestamp, retweeted_status_id***, and ***retweeted_status_user_id*** columns using the ***drop*** function, as I don't need any retweet data.
4. Then, I created a new column (***stage***) and set the default value to ***None*** to place all the dog stages in one column.
5. After that, I concatenated all the values in the four dog stage columns (doggo, floofer, pupper, and puppo) and removed the None values.
6. I realized some rows had two dog stages (because there were two dog stages in the text). So, I delimited the values with a comma for better readability.
7. Then, I replaced the empty spaces I created in step 5 with ***nan*** values
8. Next, I investigated further, noted that some stages weren't two, and then cleaned them accordingly.
9. After that, I extracted the post sources from the links.
10. Then, on to another series of dropping operations.
11. After that, I changed the datatype of the ***timestamp*** column from object to **DateTime**.
12. Then, I dropped rows with ***rating_numerator*** as 0.
13. Next, I created a column that states the breed the neural network determined.
14. I then created a column that states the calculated rating instead of 2 columns having the numerator and denominator.
15. After that, I replaced the ***name*** column with "a", "an", and other improper name values with nan.
16. Then I merged the three data frames to get a master data frame
17. After that, I replaced all **"None"** values with NaN
18. Last, I changed the data type of the ***tweet_id*** and other id columns to object.

STORING DATA

This stage is probably the most boring. Here, I saved the gathered, assessed, and cleaned master dataset to a CSV file named ***"twitter_master.csv".***

CONCLUSION

Although the usual convention (at least, as was taught at Udacity) is to handle the tidiness issues before the data quality issues, I ordered the cleaning tasks as I thought they were essential and would make the data clean.