DATA WRANGLING : WeRateDogs TWITTER ARCHIVE

The dataset analysed in this project was the twitter archive of the twitter account @dog_rates. The User name of the account iis WeRateDogs. It is a twitter account that posts dogs and rates them according to a unique rating system where the denominator is 10 and numerators are generally greater than 10 e.g. 13/10.

The dataset used was gathered from three different sources.

- The twitter archive which was sent to udacity and provided for use in this project.
- The image predictions tsv which contains breed predictions for the images of dogs in the tweets.
- The tweet_json which was data gathered by querying the Twitter API.

GATHERING DATA

The first order of business was gathering the pieces of data which i would need for my analysis.

- The twitter archive dataset had been provided as a file named ‘twitter_archive_enhanced.csv’. I simply uploaded this file to my remote directory and read it into a pandas dataframe using pd.read_csv.
- The link to the image predictions file was provided. I downloaded this programmatically using the requests library and wrote the contents into a file named ‘image_predictions.tsv’.
- The remaining tweet data was gathered from the twitter API. Using the tweet IDs in the WeRateDogs Twitter archive, I queried the Twitter API for each tweet's JSON data using Python's Tweepy library and stored each tweet's entire set of JSON data in a file called ‘tweet_json.txt’. Each tweet's JSON was written to its own line. Then I read this .txt file line by line into a pandas DataFrame containing the columns tweet id, favorite count and retweet count. I also saved this dataframe to a csv file so I could easily assess it whenever needed.

ASSESSING AND CLEANING DATA

I assessed all three datasets both visually and programmatically and made a note of all the quality and tidiness issues I observed. The data was rife with issues and was quite messy.
I proceeded to clean the issues I had noted that would affect the quality or even the functionality of my analysis.
These issues and their solutions are summarized in the table below

QUALITY ISSUES

ISSUE	SOLUTION
The source column in the twitter archive dataset had html tags in the entries which made the text harder to read interpret	I extracted the relevant text from the string using regular expressions and pandas .str.extract()
There were erroneous entries for dog names which were wrongly extracted from the tweet text. These were in lowercase as opposed to actual names which were in title case	I extracted the names that were in lower case and replaced them with None (to match entries with no recorded name) using regular expressions and pandas .str.replace()
Invalid values in the rating denominator column. All denominators ought to be 10	I located the rows with invalid values and manually corrected what I could with ratings from the tweet text. I also dropped rows that featured multiple dogs with added ratings and rows with no valid rating in the text
Invalid values in the rating numerator column.	I located these rows and manually updated what I could with correct data from the text. I also converted this column to a float to enable addition of decimal values. I dropped the rows that had invalid values that could not be corrected.
Irrelevant columns in the dataset	I used the pandas drop function to drop all columns that were not required for my analysis.
Non-descriptive column names in the image predictions table (p1,p2,p3,p1_conf,p2_conf,p3_conf,p1_dog ,p2_dog,p3_dog)	I renamed this columns so that they would be more descriptive using the pandas .rename() function. For example, p1 became top_dog, p1_conf, top_confidence_level and p1_dog, top_dog and so on.
The Predicted dog breeds in the image predictions table were in an inconsistent string format with lower and uppercase entries and underscores in the names	I replaced the underscore with white spaces using pandas .str.replace() and changed all entries to title case.

The datatypes were not correct for some columns. These were tweet id which should have been a string but was an integer, the created dog_stage column (to be explained in tidiness issues) and the source which needed to be categorical and the timestamp which was not in datetime format .	I converted these columns to the appropriate datatype using .astype() and pd.to_datetime()

TIDINESS ISSUES

ISSUE	SOLUTION
Retweets were present in the dataset. These are essentially duplicated data because the refer to the same dog entries and ratings as the original tweet.	I filtered for only rows that did not have a retweeted status id which means that they are not retweets
Dog rating should be in one column for ratings instead of in two for the numerator and denominator	I created a rating column by dividing the rating numerator column by the rating denominator column and then dropped the numerator and denominator columns
Floofer, Puppo, Pupper and Doggo are all dog stages and should be in one column for dog stage rather than four different columns for each dog stage	I extracted the dog stages from the tweet text and assigned these values to a new column called dog_stage. I then dropped the floofer, puppo,doggo and pupper columns.
Dataset is in three different tables. This should all be in one table because they all refer to the same unique tweets as evidenced by the tweet ids	I merged the twitter archive, image predictions and tweet json data into one master dataset called twitter_master on their tweet ids .

I saved this master dataframe in a csv file named ‘twitter_archive_master.csv’. And then I proceeded to analyze my clean data to gain valuable insights.