INTERNAL REPORT

TO	UDACITY
FROM	Edun Joshua
DATE	August 24, 2022
RE	WeRateDogs Tweet Analysis

Procedure:

1: Gathering data

2: Assessing data

3: Cleaning data

4: Storing data

5: Analyzing, and visualizing data

6: Reporting

Step 1: **Data Gathering**

This project is supposed to demonstrate proficiency in gathering data in 3 different ways.

1. Importing a flat file
2. Downloading programmatically
3. Making API queries.

However, due to Twitter not validating my Twitter Developer account, I could not gather data via API query and instead resorted to reading the JSON file supplied by Udacity. I loaded these data sources into the variables *df_tweets*, *df_neural*, and *df_join*

INTERNAL REPORT

Step 2: **Data Assessing**

I systematically checked for data quality and tidiness issues both visually and programmatically.

Here is a summary of my assessment:

1. Tweet archive (*df_tweets*):

- Columns with nulls:
- There are 'None's in more than 90% of these rows overall
- Invalid names such as 'a', 'actually', 'all', 'an', ..., 'very' are present in the name column
- There are 1976 dogs without classification
- Incorrect extraction of ratings due to the *text* field containing multiple ratings
- Extraction errors reflected in both *ratings_numerator* and *ratings_denominato*r columns
- *source* column contains information within HTML tags
- Variable names as column names
- *timestamp* is in object data-type
- *expanded_urls* is 2.5% NaNs

2. Image Predictions File (*df_neural*):

- Duplicates in the *jpg_url* column.
- ʼhe algorithm made some predictions that are not dogs.
- 543 rows out of 2075 of the entire dataframe, are not dog predictions at the p1 level; 522 at the p2 level and 576 at the p3 level.
- The prediction columns should be collapsed.

3. Additional Data via the Twitter API (JSON file) (*df_json*):

- No issues observed

Step 3: **Data Cleaning**

Before cleaning, I:

- I created copies of the dataframes
- Then inner merged on the common column tweet_id

The resulting dataframe has a shape (2067, 29).

I took care of the issues identified in the assessment phase in the following steps

1. Visualize and drop null-infested columns
2. Identify the retweets and remove them
3. Convert timestamp column to a datetime datatype
4. Use a regex pattern to extract the content between the tags
5. Extract fractions from the text column and split by "/". Then treat on a case-by-case basis for exceptions
6. Convert the newly extracted columns, *rating_numerator* and *rating_denominator*, to the float datatype
7. Identify culprit rows by filtering for names with low character counts and input correct names if names really exist in the text column. If no name exists, impute generic name 'Nameless'.
8. Extract Dogtionary classifications using regex and drop original columns
9. Write a conditional statement to form a single predicted_breed column depending on 'p1_dog', 'p2_dog', 'p3_dog' taking into account their prediction confidence levels and then dropping the redundant columns
10. Use the .astype method to convert *source*, *dog_class*, and *predicted_breed* into the category datatype.
11. Split the dataframe before storing: *tweets_df* to hold tweet data and *dogs_df* to hold dog data

INTERNAL REPORT

Step 4: **Storing Data**

I saved the whole cleaned dataframe and also the derived dataframes into .csv files

Step 5: **Analyzing and Visualizing Data**

First, I performed Exploratory Data Analysis on the df_join_clean, visualizing the distribution of each column to better understand the data.

I split the EDA into two sections:

- EDA for categorical data
- EDA for continuous data.

Then, I proceeded unto analysis, answering the following questions:

- What dog breed was rated the highest?
- What dog class was retweeted the most?
- What trends are there in tweet activity over time?

Using barplots, , stripplots, boxplots and heatmaps, I properly communicated the findings

Step 6: **Conclusion**

I summarized my findings succinctly.