@WeRateDogs Tweets Analysis

- Gathering the Data

In order to perform this analysis, 3 sets of data were needed.

First, the ‘twitter-archive-enhanced.csv’ file was needed. This file contains the tweets archive of @WeRateDogs, and it was provided by udacity.

Second, ‘image_predictions.tsv’ was downloaded programmatically.

Finally, using twitter api and tweepy library, json data for each tweet as was collected via the tweet id provided in the tweets archives.

- Assessing the Data

- First exploring all provided datasets showed that they should all be merged into one single dataframe.
 -Only original tweets should be included in the analysis. Through programmatic inspection, few columns related to retweets and replies were found.
- Few columns had improper data types that needed to be changed for effective analysis.
-Visual inspection showed that dog names extracted from the tweet's text were wrong on several occasions.
- Dog type extracted from tweets had some faults in it. Dog type itself was misrepresented, as four columns were used to describe a single variable.
-The timestamp column had an unnecessary string at the end of the date and time. -
- Source tweets were represented in a long and confusing string,which could be simplified.
- Numerator and denominator ratings had some wrong values.
- Dog names provided in the twitter archive had some faults in it. Some tweets had no names stated in them, or the wrong string was extracted from tweet text and labeled as name.
- Visual inspection also showed not all tweets rated dogs.
- Dog breeds predicted by the neural network were represented in many forms.
- Neural network predicted more than one dog breed, which causes an issue during analysis as more than one breed is assigned to each dog.

- Cleaning the Data

First step in cleaning the data was to create copies of all dataframes to be analyzed. Then, in order to perform accurate analysis all three datasets were merged together to create a master data set. Then rows containing data of replies and retweets were filtered out of the master dataframe. In order to rate a dog an image must be uploaded with the tweet so tweets with no images were filtered out and dropped. After filtering retweets and replies columns representing data related to them were dropped using pandas drop method. Next, columns classifying the dog type as doggo, floofer, pupper or puppo were merged into a single column using pandas melt method, this created duplicates. The solution was to sort the dataframe and drop duplicates through tweet id. It was stated in the project guideline that dog classification might be wrong so we reiterate to assess programmatically and visually to check where the wrong values are by counting values of each type and visually inspecting suspected tweets and correcting the wrong values manually.

We need to extract the dog breed predicted by the neural network. So we loop through the columns containing image classification and determine which breed to keep in the dataset by adding it to the dataframe with its confidence . Other columns relating to image prediction are dropped as they are no longer needed.

Using the string split method we remove the ‘+0000’ extra characters in the timestamp column. Then using regular expression tweets source is extracted from a long complicated string.

Running the name columns through a loop to get lower case names, which was found to be the wrong names during visual inspection. It was found some of the names in tweets which have wrong names followed a different pattern than the rest of the dataset. Those names were extracted from said patterns using regular expression. Rest of the names were inspected manually and replaced programmatically if no name was found in the tweet.

Several columns had wrong data types, those were fixed using astype method. Some column names were changed to be more descriptive. Finally, dog_breed string values were represented in many forms, so they were standardized.

Sources:

Stackoverflow.com
Geeksforgeeks
Python documentation
Pandas documentation
Previous lessons examples
regex.com