WeRateDogs Data Wrangling Project

We analyzed data from the tweet archive of the user @dog_rates who is popularly know as WeRateDogs. They have a unique rating system where the rating denominator is usually 10 but the numerators are almost always greater than 10.

The data wrangling process consisted of three steps:

Gathering Data:

Data was gathered from three different sources for this project:

1. The tweet archive for WeRateDogs stored in twitter-archive-enhanced.csv file was made available to us by Udacity. We simply had to download the file and use the read_csv function in pandas to read the data into a new dataframe twitter_archive_df

2. The image-predictions.tsv file which contains data related to the image predictions for different dog breeds. This file has been created by Udacity and is available on Udacity's server. We used Requests library to download the file programmatically to the current folder we are working in. Once the file has been downloaded we read into a dataframe called image_predictions_df.

3. We used tweet_ids to query Twitter API to obtain each tweet's JSON data. If we did not find the tweet_id, then we write the tweet id in JSON format to the file. We read each line from the JSON text file and extract the tweet_id, retweet_count and favorite_count from the JSON and populated it in a new dataframe.

Assessing Data:

There are two ways of assessing data:

1. Visual Assessment
   We observed each dataframe by printing it out in the Jupyter notebook and we physically observed the data provided in the csv files.

2. Programmatic Assessment

We use functions and methods like info(), head(), value_counts() and specific code to assess data issues which are not visible. For several discrepancies in data, we had to open each tweet post individually and evaluate.

The issues found during assessment were classified into Tidiness Issues and Quality Issues based on if there was an issues with structure or the actual data itself. We focused on original tweets in this project which had an image. Identifier were found using Twitter website to identify retweets and replies.

Cleaning Data:

Before we begin cleaning, we created copies of each of the dataframe, so if any errors occur, we can restore our dataframe and the original one does not get affected.

Cleaning data issues is usually conducted in three steps: Define, Code and Test. We first define what cleaning action will be performed, then write the code to clean and then test if the cleaning was successful.

We had to use the merge function() to combine rows from the other dataframes with the twitter_archive_df to create a master dataframe.

We had to define a custom function with multiple if statements to combine the doggo, puppo, floofer and pupper columns values and store a value ‘Not Classified’ for null values.

For removing HTML tags from the Source column, a custom regular expression had to be devised to extract the source values.

For extracting the dog breed prediction and confidence level that matched, a custom function had to be written using nested if statements.

Based on what kind of analysis we wanted to perform, the columns that did not provide necessary data will be deleted.

The cleaned data had to finally be exported to a csv file using the to_csv function.