Wrangling Report

In this report I am going to describe my wrangling and cleaning effort in python.

Importing Required Library's

First before doing anything we have to import the required library's for this project which are:

NumpPy: A library that executes code in low-level for better execution speed.

Pandas: To store the data in dataframes which are very fast and easy to manipulate.

requests: To request files from the internet.

json: To parse json string as dictonary.

matplotlib: a versatile library for plotting data.

Gathering Data

After importing the required library's we now need to gather the dog ratings from WeRateDogs

Twitter account using the API but unfortunately the free plan doesn't provide the required permissions to lookup for tweets anymore so I had to download the twitter file provided from udacity.

Now we have to gather 3 files to continue the analysis:

twitter-archive-enhanced.csv: the archive of tweets provided from WeRateDogs

image-predictions.tsv: this file includes the data generated from the nerual network to predict the dog breed from the image in the tweet.

tweet-json.txt: this file included the gathered tweets using twitter API.

Then we download the files programmatically using the requests library.

After gathering the required data using the requests library and storing the files into the machine we are going to load **tweet-json.txt** then parsing it to python dictionary we will get the data we are going to use which is the **id retweet_count** and **favorite_count** then adding it to a dataframe called **df_tweets_dump** then adding **image-predictions.tsv data to df_image_predictions and loading twitter-archive-enhanced.csv** to **df_twitter_archive.**

Accessing Data

Now after gathering the required data we are going to access the data visually and programmatically to find the quality and tidiness issues.

Quality Issues:

- In **df_twitter_archive** some tweets **expanded_urls** are missing.
- In **df_twitter_archive** not all the tweets are about rating dogs so instead of making **rating_numerator** and **rating_denominator** NaN it include random numbers from the tweet.
- In **df_twitter_archive** some tweets doesn't include the dogs name.
- In **df_twitter_archive** some tweets doesn't add the correct name of the dog.
- In **df_twitter_archive** not all the tweets includes the stage of the dog
- In **df_twitter_archive** it includes a source column for the download link of twitter for iphone which doesn't serve any purpose.
- In **df_image_predictions** it is not required to predict all the images in tweet the includes multiple images because all the images represent the same dog.
- In **df_twitter_archive** we can remove the tweets that are not about dogs ratings by using top 1 prediction from df_image_predictions.
- In **df_twitter_archive** Some dog names are not valid such as: 'a' 'such' 'the' 'just'.
- In **df_twitter_archive** Empty names are defined with 'None' instead of NaN.
- **df_image_predictions** we only need the top 1 prediction which is the most reliable prediction.

Tidiness Issues:

- **df_tweet_dump** data should be merged with **df_twitter_archive**.
- **df_twitter_archive** includes retweets which is not required (we can check for retweets by gathering all the rows with **retweeted_status_id** or **retweeted_status_user_id** that is not equal to null).
- We can add the dog breed to the tweet in the archive using the images predictions provided from udacity.
- **df_twitter_archive** includes replies which is not required (we can check for replies by gathering all the rows with **in_reply_to_status_id** or **in_reply_to_user_id** that is not equal to null).
- Remove all the tweets in **df_twitter_archive** which is not included in **df_image_predictions** to avoid unreliable data.
- In df_twitter_archive uses 4 columns for each dog stage instead of using one column for the dog stage.

IN TOTAL WE HAVE 11 QUALITY ISSUES AND 6 TIDINESS ISSUES THAT NEEDS TO BE CLEANED, ALL OF THE CLEANING PROCESS IS DONE IN JUPYTER NOTEBOOK USING DEFINE-CODE-TEST FRAMEWORK.

Storing Data

After completing the cleaning process we have a cleaned twitter archive, now we are going to store the data in our local machine for future use using pandas to_csv() function.