DATA WRANGLING REPORT

Ebenezer Mayowa, PEACE

Project Purpose: To practice what I have learned from the Data Wrangling section in Udacity Data Analyst Nanaodegree program. The dataset that is wrangled is the tweet archive @DogRates, also known as @WeRateDogs. We rate dogs is a Twitter account that rates people's dogs with a humorous comment about the dogs.

Project Goal: to effectively wrangle data related to dog ratings.

Project Steps Overview

Step 1: Gathering data

Step 2: Assessing data

Step 3: Cleaning data

Step 4: Storing data

Step 5: Analyzing, and visualizing data

Step 6: Reporting

Gathering Data;

The data used for this project were from three different datasets, and they were obtained differently.

The first Data, Twitter_archive was provided by Udacity, I manually downloaded it to my project workspace. Guidelines on how to were already provided in the project guideline.

The second data, Tweet Image Prediction was downloaded programmatically, using the python requests library and the url provided by Udacity.

With the **with open** function, I wrote the response to a tsv file, then read the downloaded file into a dataframe.

```
# URL
url = 'https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-predictions/image-predictions.tsv'
response = requests.get(url)

# Code to open a tsv file and save the response content
with open('image-predictions.tsv', mode='wb') as file:
    file.write(response.content)

# Read the TSV file
image_predict = pd.read_csv('image-predictions.tsv', sep='\t')
```

The third data needed me to create a developer account and create a twitter Api for the project, I could not do such as I was being flagged multiple times so I decided to use the data provided by Udacity in the accessing data column of the project.

Supporting Materials

- twitter_api.py
- tweet_json.txt

With the **with open** function, I read the file into the project workspace and saved them to a dataframe called tweet_json.

Accessing Data;

I accessed the data from the three files provided both visually using the jupyter notebook and spreadsheets applications and in the case of the tweet_json file made use of notepad, and statistically (using functions like .head, .info, .describe) to ensure correctness and tidiness.

Cleaning Data;

This section was where I made sure to use the Define, Code and Test Framework.

After making a duplicate of my datasets, I removed the columns I didn't need for my analysis, I combined the dog stages seeing as they were in various columns and I would still need to combine the three data sets into one master file, I also converted columns into their appropriate data types, foe example, the timestamp column was converted from an integer value to a date time.

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2175 entries, 0 to 2355
Data columns (total 11 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   tweet_id            2175 non-null   int64
 1   timestamp           2175 non-null   object
 2   source              2175 non-null   object
 3   text                2175 non-null   object
 4   rating_numerator    2175 non-null   int64
 5   rating_denominator  2175 non-null   int64
 6   name                2175 non-null   object
 7   doggo               2175 non-null   object
 8   floofer             2175 non-null   object
 9   pupper              2175 non-null   object
 10  puppo               2175 non-null   object
dtypes: int64(3), object(8)
memory usage: 203.9+ KB
```

```
df1cleaned['timestamp'] = pd.to_datetime(df1cleaned['ti
```

```
df1cleaned.dtypes
```

```
tweet_id                            int64
timestamp             datetime64[ns, UTC]
source                             object
text                               object
rating_numerator                    int64
rating_denominator                  int64
name                               object
doggo                              object
floofer                            object
pupper                             object
puppo                              object
dtype: object
```

After doing all these, I merged the three dataset into one data called twitter_archive_master.csv.

Storing the data;

Here, I stored the three data sets that have been merged.

SAVING OUR DATA

```
#Let us save to a folder named twitter_archive_master.csv
data.to_csv("twitter_archive_master.csv", index=False)
```

```
#Let us check if our code worked
data = pd.read_csv("twitter_archive_master.csv")
data.head(20)
```

Conclusion

Despite several setbacks, I was able to complete project with the help of Google, stackoverflow and peers who helped out when I reached out for help. Data wrangling takes 80% of the time during a data analysis project and shouldn't be belittled..