wrangle_report

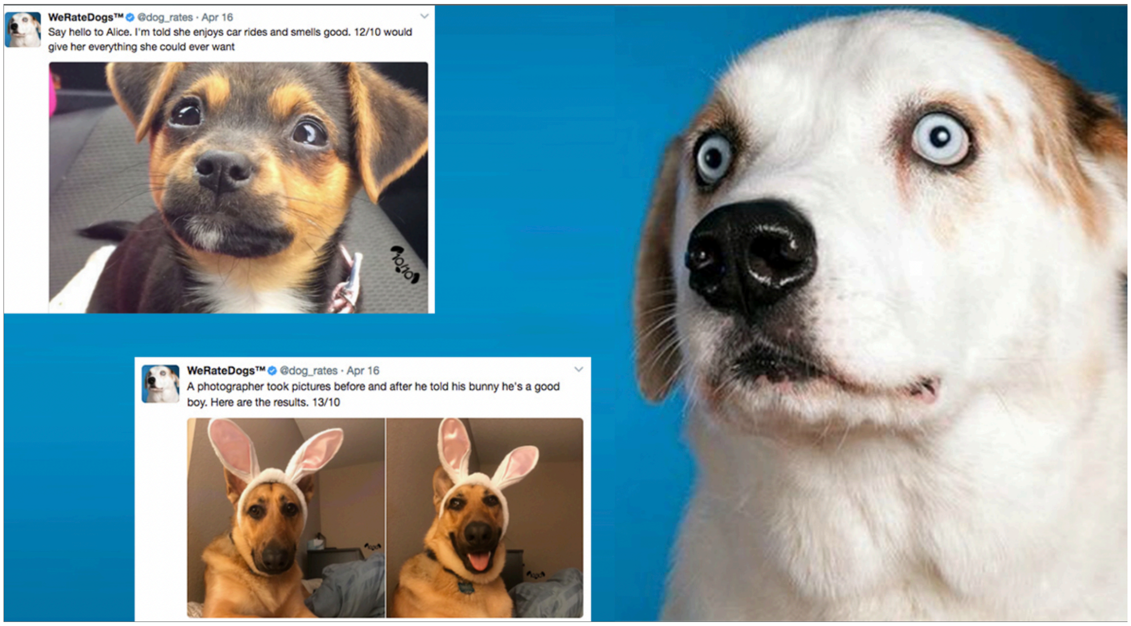
September 28, 2022

0.1 Reporting: wragle_report

0.1.1 Introduction

This is a project done in fulfilment of the requirements of the Udacity nanodegree. In this project, I will be using Python and its libraries to gather data from variety of sources and in variety of formats, access the quality and tidiness of the gathered data, and then proceed to clean it. This project aims to bring to fore the utilisation of all that was learnt in the Data wrangling module of the nanodegree.

The dataset I will be wrangling, analyzing, visualising and drawing insights from, is the tweet archive of a Twitter account @dog_rates that rates people's dogs with a humorous comment about the dog. The ratings of these dogs as seen in the Twitter account almost always have a denominator of 10, while the nemurator has rating mostly above 10. At the end of this project I hope to pose and answer some questions regarding the datasets worked with and draw some insights from such answers.



WeRateDogs™ @dog_rates · Apr 16
Say hello to Alice. I'm told she enjoys car rides and smells good. 12/10 would give her everything she could ever want

WeRateDogs™ @dog_rates · Apr 16
A photographer took pictures before and after he told his bunny he's a good boy. Here are the results. 13/10

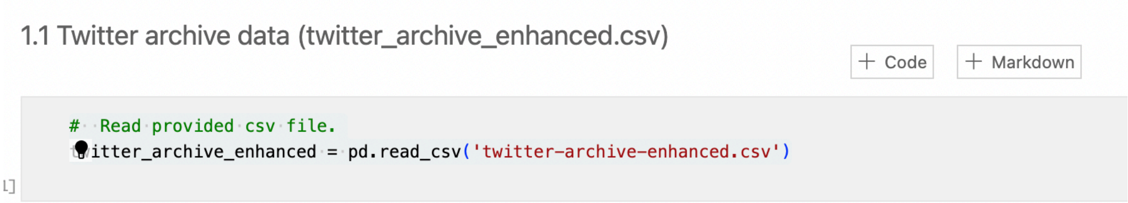
0.1.2 Data Wrangling Process

Importing libraries. To get a job done, one has to be properly equiped with neccesary tools. Therefore since I will be using Python for this project, it was only fitting that I import its libraries and packages that will be needed for every stage of this project. These include but not limited to pandas, requests, tweepy and seaborn.

0.1.3 Data Gathering

Udacity in collaboration with WeRateDogs (the twitter account whose tweets I will be analysing), this archive contains basic tweet data (tweet ID, timestamp, text, etc.) for all 5000+ of their tweets as they stood on August 1, 2017. For the purpose of this project, three (3) datsets will be needed; the WeRateDogs Twitter archive, the tweet image predictions and additional data from the Twittwer API.

WeRateDogs Twitter Archive This dataset was provided by Udacity in a csv file format which I downloaded. Gathering this dataset was staright forward as it had already been provided. Using the pandas function *read_csv* I uploaded and read the data into a pandas Dataframe I named **twitter_archive_enhanced**.

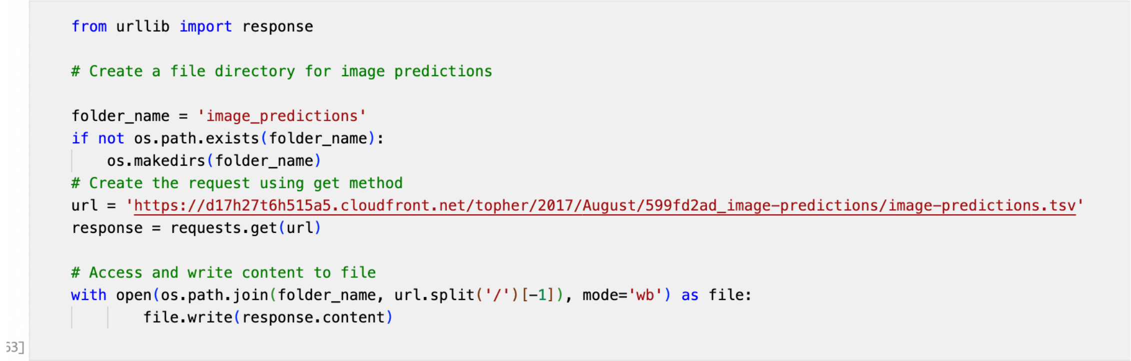


1.1 Twitter archive data (twitter_archive_enhanced.csv)

```
# Read provided csv file.
twitter_archive_enhanced = pd.read_csv('twitter-archive-enhanced.csv')
```

Code

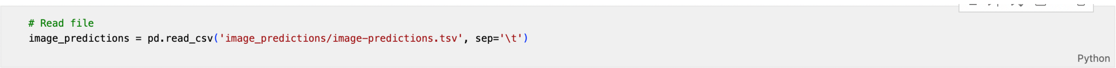
Tweet Image Predictions The dataset for image predictions was programatically downloaded from Udacity's servers using the Requests library. This came as a tsv file which I also read into a pandas Dataframe name **image_predictions** using *read_csv* and ”̂ as seperator.



```
from urllib import response

# Create a file directory for image predictions

folder_name = 'image_predictions'
if not os.path.exists(folder_name):
    os.makedirs(folder_name)
# Create the request using get method
url = 'https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-predictions/image-predictions.tsv'
response = requests.get(url)

# Access and write content to file
with open(os.path.join(folder_name, url.split('/')[-1]), mode='wb') as file:
        file.write(response.content)
```

Code

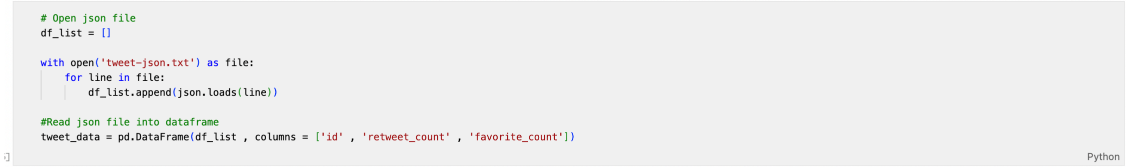
,



```
# Read file
image_predictions = pd.read_csv('image_predictions/image-predictions.tsv', sep='\t')
```

Additional Data From Twitter API The WeRateDogs Twitter archive contains basic tweet data but not everything. Therefore the Twitter API needed to be querried to get favorite/like counts as well as retweets count. Two ptions were given by Udacy to obtain this data; creating a Twitter developer account and using Tweepy to querry the Twittter API, or downloading the

resulting json file from the querry provided by Udacity. I opted for the later option as I was already behind time at the time I started the project.
I therefore read the json file line by line into a pandas Dataframe I named **tweet_data**.



```python
# Open json file
df_list = []

with open('tweet-json.txt') as file:
    for line in file:
        df_list.append(json.loads(line))

#Read json file into dataframe
tweet_data = pd.DataFrame(df_list , columns = ['id' , 'retweet_count' , 'favorite_count'])
```

Code

0.1.4 Accessing Data

After successfully gathering the three (3) datasets needed for my analysis, the next step is to visually and programatically access them for quality and tidiness issues. Pandas functions like *head*, *info*, *describe*, and *sample* were employed in determining some of the datasets issues, these are; 1. Twitter Archive

```
* tweet_id is **int** instead of **string**
* Nulls represented as "None" in the dataset.
* Wrong dog names like "a", "an", "my", "unacceptable" etc in the **name** column.
* **timestamp** is string instead of datetime type.
* For the purpose of this project we are only interested in original tweets, therefore retweet
* Columns like **expanded_urls** and **source** not required for my analysis.
```

2. Image Predictions Table

   - tweet_id is **int** instead of **string**.

   - Columns **p1**, **p2**, and **p3** has values starting with both upper and lower cases.

   - Missing **tweet_id**s values.

   - Some columns not properly named, that is not descriptive.

3. Additional Resources (tweet_data) Table

   - **id** column is **int** instead of **string**.

   - **id** column should be “tweet_id”

0.1.5 Data Cleaning

After highlighting the quality and tidiness issues in the access phase of the project, I proceeded to cleaning the issues documented using the define-code-test method. But not without first making copies of the original datasets, merging individual pieces datasets into a single master pandas Dataframe named **clean_twitter_archive** I also made sure to first deal with the tidiness issues as it is best practice.