WRANGLE REPORT

- ## Gathering data

➔ By reading the twitter archive provided by the udacity.

.

```
#reading the "twitter-archive-enhanced.csv" file in t
tae_df=pd.read_csv('twitter-archive-enhanced.csv')
tae_df.head()
```

➔ Downloading the Prediction data and reading it in pandas.

```
#programmably downloading the file
url = 'https://d17h27t6h515a5.cloudfront.net/topher/2017/A
nn = requests.get(url)
with open(url.split('/')[-1], mode='wb') as file:
    file.write(nn.content)
```

➔ Scraping Data using Tweepy

Each tweet's retweet count and favorite ("like") count at minimum were scraped using the tweet IDs in the WeRateDogs Twitter archive, queried the Twitter API for each tweet's JSON data using Python's Tweepy library and store each tweet's entire set of JSON data in a file

```
consumer_key = '#CPrAP36MamWx'
consumer_secret = '#B5DBc15IjeY1kWa9lVPy22yLlBN'
access_token = '#ZIKGjpdMOYr2f0FT7zyiafVpn0JF'
access_secret = '#fmPlDiJha5qbWLTGKhzXDU'

auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_secret)

try:
    api = tweepy.API(auth,
                     parser = tweepy.parsers.JSONParser(),
                     wait_on_rate_limit=True,
                     wait_on_rate_limit_notify=True)
except:
    print("ERROR")
```

called tweet_json.txt file. Each tweet's JSON data should be written to its own line. Then read this .txt file line by line into a pandas DataFrame with (at minimum) tweet ID, retweet count, and favorite count.

➔ Saved the dataframes for further wrangling

```
In [263]: #reading the scraped data
          scraped_df=pd.read_csv('tweet_json.txt', encoding = 'utf-8')
```

```
In [145]: #reading the prediction file
          ip_df = pd.read_csv('image-predictions.tsv', sep='\t')
```

```
In [146]: #reading the twitter archive
          tae_df=pd.read_csv('twitter-archive-enhanced.csv')
```

- ## Assessing data

➔ After gathering each of the above pieces of data, I assessed them visually and programmatically for quality and tidiness issues. I Detected and document quality issues and tidiness issues.

> **Quality Issues**

Twitter Archive :-

- 1) The data type of tweet_id should be string since we do not need to perform any calculations on it.
- 2) Timestamp datatype is incorrect
- 3) Denominator of some of the ratings is incorrect,i.e., not on a fixed scale 10.
- 4) Some of the tweets are retweeted with respect to the first.
- 5) Some dogs names are missing or unknown and marked as none,a,the.
- 6) There are unnecessary columns in the dataset.

Prediction file :-

- The data type of tweet_id should be string since we do not need to perform any calculations on it.

Scraped data :-

- The data type of tweet_id should be string since we do not need to perform any calculations on it.
- 7) There are 84 duplicate values in our scraped data.
- 8) There are 14 tweets which we were not able to read with the following tweet id's:-

Tidiness Issues ¶

- 1) There are certain columns defining the breed of the dog, these should be melted to a single column.
- 2) 'Unnamed: 0' column should be removed , as it does not possess any useful info.
- 3) Scraped data and the twitter archive should be merged into one dataset and also the prediction file.

• Cleaning data

➔ Before I started cleaning the datasets I created copies of each dataset. Afterwards, I tried to fix every problem programmatically. First, I fixed the quality issues .Some issues were fixed in one cleaning step as they required only a single line of code execution.. In other cases I had to plan the logic manually. After this, I fixed the tidiness issues, merged the datasets, and fixed the remaining quality issues.

- **Storing data**

➔ After cleaning the data, I stored the final cleaned dataset as a csv-File:

master_data.to_csv ('master_data.csv', index=False)

```
#storing the masterdata into a csv file for further analysis
master_data.to_csv('master_data.csv', index=False)
```