WRANGLE AND ANALYZE DATA

WRANGLE REPORT

INTRODUCTION

In this Project, I had to extract the data from the twitter using API. I had to analysis the tweets from WeRateDogs twitter account. WeRateDogs is a Twitter account that rates people's dogs with a humorous comment about the dog. These ratings almost always have a denominator of 10. The numerators, though? Almost always greater than 10. 11/10, 12/10, 13/10, etc. Why? Because "they're good dogs Brent." WeRateDogs has over 4 million followers and has received international media coverage.

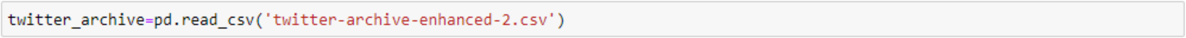
Steps for Data Wrangling:

1. Gather the data
2. Access the data
3. Clean the data

GATHERING:

This project consists of three datasets:

1. **Twitter-archive:** This dataset was provided by udacity. The WeRateDogs Twitter archive contains basic tweet data for all 5000+ of their tweets .



```
twitter_archive=pd.read_csv('twitter-archive-enhanced-2.csv')
```

2. **Image-predictions:** This dataset was to be downloaded programmatically. The results: a table full of image predictions (the top three only) alongside each tweet ID, image URL, and the image number that corresponded to the most confident prediction (numbered 1 to 4 since tweets can have up to four images).
3. **Twitter API and JSON:** By using tweet_id, I'm extracting favourites, retweets and date and time. After extracting the data I will be appending the data into a list of dictionaries. Then storing the data into **tweet_json.txt** file.

ACCESS:

After Gathering the data. I have accessed the three files through python.

Checking for any duplicates urls or empty columns in the datasets, describing the datasets and checking the datatypes of the data.

QUALITY:

- There were several empty columns.
- The timestamp column is an object.
- In some numerator columns weren't matched to rating numerator.
- Missing values from images dataset in image_predictions.
- There were several duplicated urls in the image_predictions.

CLEANING

After accessing the data I have cleaned the data

- Merge the entire data
- Melting the 'doggo', 'floofer', 'pupper' and 'puppo' columns into one column 'dog_state'
- Drop the columns we will not need – in_reply_to_status_id, in_reply_to_user_id, retweeted_status_id, retweeted_status_user_id, retweeted_status_timestamp, expanded_urls, date_time
- Clean the rows- duplicated tweet_id and tweets with no pictures.
- Condensing dog breed predictions
- Drop 66 jpg_url duplicated
- Convert the datatypes.
- Extract Dog Rates and Dog Counts.

STORING THE DATA

After the cleaning, the data I have stored the cleaned data into **twitter_archive_master.csv**.