# WRANGLE AND ANALYZE DATA

Wrangle Report

JUNE 19, 2019

5TH PROJECT

Udacity Data Analysis Nanodegree

# Contents



# Figures

# 1. Abstract

A lot of data around the word some of those data aren't cleared , structured, or organized, based on **DOMO** site *"Over 2.5 quintillion bytes of data are created every single day, and it's only going to grow from there. By 2020, it's estimated that 1.7MB of data will be created every second for every person on earth."*[1].

In this project I will practice what I learned in data wrangling Udacity's Data Analysis Nanodegree program. The dataset that I will be wrangling is the tweet archive of Twitter user **@dog_rates**, also known as **WeRateDogs**.

# 2. Introduction

**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."*[2]. So, I will use those tweets (dataset) to wrangling data.

In this project the data wrangling, which consists of:

- Gathering data, download resources from deferent references.
- Assessing data, for quality and tidiness issues.
- Cleaning data, clean the quality and tidiness issues that identified in previous step.

The most tools, libraries and programming language used in this project are:

- Python
- Pandas Library
- Numpy Library
- Requests Library
- Tweepy Library
- Json Library
- Matplotlib Library
- Jupyter Notebook
- Twitter's API

# 3. Gathering Data

In this section I will talk about gathering data in detail.

Data was gathered are from three deferent sources:

1- The WeRateDogs Twitter archive file (.csv) was provided and downloadable file in the Resources. This file include attributes for each tweet which are tweet id, timestamp, text, rating numerator, rating denominator, name, dog stage etc (See Figure 1). Then save it in dataframe named archive_df

```
In [88]: archive_df.columns

Out[88]: Index(['tweet_id', 'in_reply_to_status_id', 'in_reply_to_user_id', 'timestamp',
                'source', 'text', 'retweeted_status_id', 'retweeted_status_user_id',
                'retweeted_status_timestamp', 'expanded_urls', 'rating_numerator',
                'rating_denominator', 'name', 'doggo', 'floofer', 'pupper', 'puppo'],
               dtype='object')
```

*Figure 1: Tweets Attributes*

2- The tweet image predictions file was downloaded programmatically by using Requests library. What breed of dog is present in each tweet according to a neural network. The file sored in dataframe named img_df. (See Figure 2)

| tweet_id | jpg_url | img_num | p1 | p1_conf | p1_dog | p2 | p2_conf | p2_dog | p3 | p3_conf | p3_dog |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 892177421306343426 | https://pbs.twimg.cor | 1 | Chihuahua | 0.323581 | TRUE | Pekinese | 0.0906465 | TRUE | papillon | 0.0689569 | TRUE |
| 891815181378084864 | https://pbs.twimg.cor | 1 | Chihuahua | 0.716012 | TRUE | malamute | 0.078253 | TRUE | kelpie | 0.0313789 | TRUE |
| 891689557279858688 | https://pbs.twimg.cor | 1 | paper_towel | 0.170278 | FALSE | Labrador_retriever | 0.168086 | TRUE | spatula | 0.0408359 | FALSE |
| 891327558926688256 | https://pbs.twimg.cor | 2 | basset | 0.555712 | TRUE | English_springer | 0.22577 | TRUE | German_short-haired_pointer | 0.175219 | TRUE |
| 891087950875897856 | https://pbs.twimg.cor | 1 | Chesapeake_Bay_retriever | 0.425595 | TRUE | Irish_terrier | 0.116317 | TRUE | Indian_elephant | 0.0769022 | FALSE |
| 890971913173991426 | https://pbs.twimg.cor | 1 | Appenzeller | 0.341703 | TRUE | Border_collie | 0.199287 | TRUE | ice_lolly | 0.193548 | FALSE |
| 890729181411237888 | https://pbs.twimg.cor | 2 | Pomeranian | 0.566142 | TRUE | Eskimo_dog | 0.178406 | TRUE | Pembroke | 0.0765069 | TRUE |
| 890609185150312448 | https://pbs.twimg.cor | 1 | Irish_terrier | 0.487574 | TRUE | Irish_setter | 0.193054 | TRUE | Chesapeake_Bay_retriever | 0.118184 | TRUE |
| 890240255349198849 | https://pbs.twimg.cor | 1 | Pembroke | 0.511319 | TRUE | Cardigan | 0.451038 | TRUE | Chihuahua | 0.0292482 | TRUE |
| 890006608113172480 | https://pbs.twimg.cor | 1 | Samoyed | 0.957979 | TRUE | Pomeranian | 0.0138835 | TRUE | chow | 0.00816748 | TRUE |
| 889880896479866881 | https://pbs.twimg.cor | 1 | French_bulldog | 0.377417 | TRUE | Labrador_retriever | 0.151317 | TRUE | muzzle | 0.0829811 | FALSE |
| 889665388333682689 | https://pbs.twimg.cor | 1 | Pembroke | 0.966327 | TRUE | Cardigan | 0.0273557 | TRUE | basenji | 0.00463323 | TRUE |
| 889638837579907072 | https://pbs.twimg.cor | 1 | French_bulldog | 0.99165 | TRUE | boxer | 0.00212864 | TRUE | Staffordshire_bullterrier | 0.00149818 | TRUE |
| 889531135344209921 | https://pbs.twimg.cor | 1 | golden_retriever | 0.953442 | TRUE | Labrador_retriever | 0.0138341 | TRUE | redbone | 0.00795775 | TRUE |

*Figure 2: Tweet Image Prediction Data*

3- Twitters's API, to get tweets that in Twitter archive by using Python's Tweepy library, and adding favorite count and retweet count. Then save the file as JSON format named tweet_json.txt and read this file and store it in dataframe named tweets_df.

# 4. Assessing Data

Assess data visually and programmatically for quality and tidiness issues using pandas.

## 4.1 Visual Assessment

Once I opened `twitter_archive_enhanced.csv` I assessed the data as following:

- Quality: html tags in source column of twitter archive. Like <a href=""http://twitter.com/download/iphone"" rel=""nofollow"">Twitter for iPhone</a>
- Quality: None values instead NaNs valus.

## 4.2 Programmatic Assessment

Assessment was performed using the following methods by panadas library:

- head()
- taill()
- info()
- value_counts()
- duplicated()
- count()
- isnull()

## 4.3 Quality issues

- archive_df
  - timestamp and retweeted_status_timestamp are object type instead of datetime.
  - source is HTML format.
  - There are records have more than one dog stage.
  - name has missing values with "None" instead of NaN.
  - doggo, floofer, pupper, and puppo have missing values with "None" instead of NaN
  - There are 181 retweeted tweets
  - There are many columns in this dataframe making it hard to read, and some will not be needed for analysis.
  - There are tweets has denominator grater than 10 and has numerators less than 10.

- tweets_df
  - There are 166 retweeted. Keep only original tweets.

- img_df
  - There are 2075 images, but archive_df contain 2335 tweets. there are 260 missing (maybe some of tweets doesn't contain image).
  - There are 66 images are duplicated.

### 4.4 Tidiness issues

- doggo, floofer, pupper, and puppo are unique columns instead one column "dog_stage"
- Merge all the 3 dataframe to one dataframe based on tweet_id.

## 5. Cleaning Data

This part of the data wrangling was performed in three stages: Define, Code and Test. In the first I made a copy for all the dataframes.

During the cleaning process I used the following methods and tools:

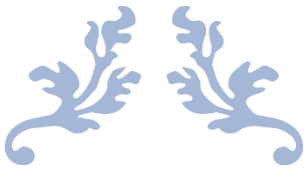
- merge()
- copy()
- info()
- drop()
- np.isnan
- astype()
- to_datetime()
- count()
- replace()
- Queries
- np.nan
- columns
- value_counts()
- apply()
- head()
- Loop

### 5.1 Storing Data

After I clean the data, now we have cleaned and structure data. I stored it into dataframe named f_df (final dataframe) then I save as csv by panada's to_csv() function and named as twitter_archive_master.

## 6. References


1- https://www.domo.com/solution/data-never-sleeps-6
2- Udacity: Data Wrangling section > Wrangle and Analyze Data Project > Project Overview

# WRANGLE AND ANALYZE DATA

Wrangle Report

JUNE 19, 2019

5TH PROJECT

Udacity Data Analysis Nanodegree