

# Data Wrangling Report

## Introduction

Data wrangling is a core skill that everyone who works with data should be familiar with since so much of the world's data is not clean. It is a process divided into 3 main steps:

- Gathering.
- Assessing.
- Cleaning.

## Gathering

Data was gathered from 3 different sources:

### **1. From WeRateDogs Twitter archive given by Udacity in csv format:**

Using panda's method 'read\_csv', I managed to read the data stored in the file 'twitter-archive-enhanced.csv'. I stored it in a DataFrame called 'twitter\_archive'. The data has many issues that will be cleaned and resolved later.

### **2. Image prediction file downloaded programmatically using Requests library and the URL provided by Udacity in tsv format:**

Using Requests library and 'get' method, data was downloaded in a file 'image\_predictions.tsv'. Then, the content was stored in a DataFrame called 'image\_predictions' using pandas' method 'read\_csv'.

### **3. Data retrieved by querying Twitter's APIs and using Tweepy library.**

Using the list of tweet\_id's in dataframe 'twitter\_archive', I made a loop through each tweet and query Twitter's APIs with the tweet ID to get each tweet's JSON data. Then, I retrieved the required data ('favorite\_count', 'retweet\_count', 'followers\_count', 'favourites\_count', 'created\_at') and store it in a list called 'df\_list'. There were some errors, and the tweet\_id of each error was stored in list called 'error\_list'. Finally, I created a DataFrame called 'tweet\_data' using the list.

- We reached the limit of Twitter APIs 2 times.
- We got 2337 tweet\_id correctly and 19 errors.
- The total time was about 1927 seconds = 32 min.

## Assessing

After gathering the data and storing them in DataFrames, the following step was assessing the data for quality and tidiness. Data were assessed programmatically and visually.

### Quality:

Issues with content. Low quality data is also known as dirty data.

Identified quality issues are:

- in\_reply\_to\_status\_id, in\_reply\_to\_user\_id, retweeted\_status\_id, retweeted\_status\_user\_id should be integers/strings instead of float.
- retweeted\_status\_timestamp, timestamp should be datetime.
- The numerator and denominator columns have invalid values.
- In several columns null objects are non-null (None to NaN).
- There are invalid names (a, an and less than 3 characters).- We only want original ratings tweets, not retweets.
- We might change the type of columns: (in\_reply\_to\_status\_id, in\_reply\_to\_user\_id, retweeted\_status\_id, retweeted\_status\_user\_id and tweet\_id) to string since we aren't doing any actions on them.
- Sources are not readable.
- Missing values from images dataset (2075 rows instead of 2356)
- Some tweet\_ids have the same jpg\_url
- Some tweets have 2 different tweet\_id, that are retweets.

### Tidiness

Issues with structure that prevent easy analysis. Untidy data is also known as messy data.

Identified tidiness issues are:

- Dog stage is in 4 columns (doggo, floofer, pupper, puppo), no need for that.
- Merge 'tweet\_info' and 'image\_predictions' into 'twitter\_archive'.

## **Cleaning**

It is the process of fixing and resolving issues identified in the Cleaning process. The (define, code, and test) steps were used in the cleaning process. First, copies of the DataFrames were created before cleaning. Then, the steps of cleaning were applied iteratively on all issues.

## **Storing**

The final DataFrame called 'twitter\_archive\_clean' contains 1990 rows and 15 columns with the correct data types. The dataset is then stored in a csv file called 'twitter\_archive\_master.csv'. At this point, the data was successfully wrangled and therefore ready for analysis and visualization.

## **Analysis & Visualization**

These steps are not part of data wrangling process. However, it cannot reflect correct and accurate insights without performing data wrangling first. Visualizations and insights are provided in 'act\_report.pdf'