**Data Wrangling Report (wrangle\_report)**

Data Wrangling has become an important skills in the data science field. It allows data analyst to extract information from any kind of data provided. This important because real-live data is messy and scattered in different sources, and even if gathered, insights cannot be found since the data is dirty and could be misleading. This report will display the steps performed to clean tweet archive of Twitter user [@dog\_rates](https://twitter.com/dog_rates), starting from gathering the data, then assessing it, and finally cleaning it.

***I. Data Gathering***

The data in this project was gathered from three different sources using three different methods of data gathering. The three methods are the following:

**1. From a saved CSV file. (*twitter-archive-enhanced.csv*)**

This was the easiest source of data since you only need to download it and store it into your project folder. And then reading the file using *pd.read\_csv()*.

**2. By downloading a TSV file hosted on the Udacity server. (*image-predictions.tsv*)**

This was achieved using the *requests* library hosted on the following URL:

https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad\_image-predictions/image-predictions.tsv

Then opening the file programmatically using *with open*.

**3. From Twitter API. (*tweet\_json.txt*)**

This data was supposed to be from a query to Twitter API. Unfourtuntly I do not have a Twitter account, so I decided to go with the second option and download the json file manually and read it using *pd.read\_json()*.

***II. Assessing the Data***

This part of the project was my favorite since you get to play the detective role and try to find underling issues within the data. Visual assessment and programmatic assessment were used effectively throughout the project. The issues found in the data can be summarized as follow:

**1. Quality issues:**

General quality issues to be cleaned:

* Some columns have names that do not represent them or have unclear names.

df\_tweetsdata frame:

* Erroneous datatypes (timestamp column should be datetime object).
* Columns with too many null values or not needed for the analysis (source, retweeted\_status\_timestamp, retweeted\_status\_user\_id, retweeted\_status\_id,in\_reply\_to\_user\_id, in\_reply\_to\_status\_id). Most of them cannot be fixed.
* Missing values in dog name column or has wrong names like 'a' or 'the' ('None' values should be replaced with NaN).
* Denominator for the rating is greater than 10 in some rows.
* Timestamps having an extra ‘+0000’.

df\_image\_pred data frame:

* Predictions are lowercased and have an extra dash (p1, p2, p3).
* Rows where all predictions are not dog predictions should be null since wrong predictions will affect the analysis later (i.e., p1\_dog and p2\_dog and p3\_dog are all false).

df\_tweets\_info data frame:

* Unnecessary columns (4 columns are just needed).
* The rows that contain 'RT' in the text column, which indicate it is a retweet, so it is redundant data and could affect our analysis.

**2. Tidiness issues:**

General structural issues:

* The 3 data frames should be joined on tweet\_id.

df\_tweets data frame:

* The 4 columns of dog breed should be combined into one column called dog\_type.

***III. Cleaning the Data***

Cleaning was the most part that took effort for me, maybe because it is a first time for me. I decided to clean the structural issues first by joining all datasets on the the *tweet\_id*. This made life easier because instead of applying a method to all datasets, you apply it once to the joined one. Then I minimized the dog type columns to one column only, this helped in the analysis part since all types are in one column. I then dropped unwanted columns that has many nulls or are not needed (this could be considered as Quality and Tidiness cleaning).

After that I moved to the quality issues, started with easier tasks like changing the types of some columns, renaming unclear columns. Secondly, changing the denominator to be all 10s, if this was not cleaned, wrong results could happen. I also made sure that ‘None’ is replaced with *np.nan*, because *np.nan* has a certain behavior in some analysis functions. I also nullified some wrong dog names such as ‘the’, ‘a’ and ‘an’. Additionally, I removed the ‘\_’ in the dog breed columns and capitalized the dog breeds to be more readable.

***IV. Storing the Data***

After cleaning was done, the data now is ready to be analyzed, so I saved it to an CSV file as *'twitter\_final.csv'* in the same project folder.