***WeRateDogs Twitter Account Wrangling and analysing project:***

Our goal was to wrangle WeRateDogs Twitter data to create reliable analysis and visualizations.

In this document I will be summarizing my data wrangling steps from gathering data from several different sources to assessing data and finally to clean the datasets and store them in a master database (db format and CSV) format.

1. Gathering data:

There were a number of sources to gather the data we required for this project. Below is a list of those sources:

* 1. Twitter\_archive\_enhanced.csv file. This file was provided by Udacity. This dataset contained data for about 2300 entries which were twitter statuses. Each tweet's text has been used to extract rating, dog name, and dog "stage" (i.e. doggo, floof(er), pupper, and puppo)." Of the 5000+ tweets, we have tweets with ratings only (there are 2356)
  2. Gathering extra data from Twitter API. We used our own consumer key and token to access the tweet statuses and download them. because we have the WeRateDogs Twitter archive and specifically the tweet IDs within it, we can gather this data for all 5000+. We used Tweepy library to query Twitter's API to gather this valuable data. I called this dataset “tweet\_df”.
  3. Downloading programmatically a TSV file containing AI predictions based on the images of the dogs, i.e., what breed of dog (or other object, animal, etc.) is present in each tweet according to a neural network. This file (image\_predictions.tsv) is hosted on Udacity's servers and should be downloaded programmatically using the Requests library.

1. Assessing data:
   1. I first visually assessed all datasets and found some issues within them. I have listed those issues here as well as in the Jupyter Notebook file.
   2. Second, we assessed the data programmatically, using great functions in Pandas.

There are 2 types of issues with the dirty data that we had in this project:

* + 1. Tidiness issues
       - The denominator of “10” and the numerators could all be in one column
       - Dog stages need to be in one single column (doggo, floof(er), pupper, puppo)
       - It is not clear if the tweet is the original status or a retweet of something else, so we need to add a column for Retweet (Yes/No) or to remove the retweets from the dataset to be left with the original statuses.
       - The Columns titles are not descriptive enough (img\_num, p(1-3), p(1-3)\_dog)
       - Number of records for this data frame is not the same as the archive. It has 2075 records while the other has 2356 records
       - Columns are not optimally arranged.
       - Should merge 2 Data Frames (Tweet\_DataFrame and Tweeter\_archive\_enhaced)
    2. Quality issues
       - “source” column is not clear, and we need to extract the exact source like iPhone, etc.
       - For “Dos Stage” and “Name”, for missing values, "None" is used rather than NaN
       - Timestamp is not standard, so we could change it to a standard format of Date-time
       - A lot of Dog Stages are missing! So, not every dog has a “dog\_stage” label.
       - All other ID fields also need to be converted into String like in\_reply\_to\_status\_id
       - It appears that we have inconsistency in extracting the ratings (the nominators). Sometimes the digits after decimal point is extracted
       - Some of the photos are not dog (like the p1 says: hummingbird, p2: peacock, p3: eel)
       - full text contains some URL and also rating which are irrelevant, need to be removed
       - Tweet\_id is in Integer form, needs to be string

After a comprehensive assessment of the data, I created a list of 9 quality issues and 7 tidiness issues in total. I have addressed all these issues in my Notebook following the steps of: (Issue, Define, Code, Test), that is to first clearly mention what the issue is, then define the method we have come up with to deal with the issue, after that we write a series of codes to resolve the problem and finally we use a few functions to test the results.

I have also analysed the data after I thoroughly cleaned the data frames and stored them in new databases (SQLite databases and CSV files). A short report on some of the analysis’s findings is also submitted.