Data Wrangling Report

Data wrangling is the process of getting the data I want from various resources **manually** or **programmatically** in order to answer some questions, use it in making business decisions, etc.

There are 3 steps in data wrangling: **Gathering**, **Assessing** and **Cleaning**.

I have done the entire data wrangling process in the project and I will briefly describe my efforts below.

Gathering Data

Gathering is the process of getting the data manually (e.g. clicking a download button) or programmatically (e.g. scrapping from the internet, downloading programmatically or retrieving from APIs).

In this project I needed 3 types of data:

- The Twitter archive from **WeRateDogs** that contains basic tweet data for Tweets with ratings only. I used the traditional way of downloading files **manually** to download this file.
- A file of predictions about what breed of dog is present in each tweet according to a neural network. For that file I used the *request library* to request a response from an URL provided on **Udacity** platform to **programmatically** download it.
- Finally, the retweet and favorite count for each tweet. I used *tweepy* access library to access the Twitter API and fetch these data for each tweet by tweet id that was originally provided the tweet archive data.

Assessing Data

After gathering, I need to assess our data either **visually**, **programmatically** or both to detect any structure or quality issues in the data, then document them to be fixed later.

Common data quality issues include: *missing data*, *invalid data*, *inaccurate data* or *inconsistent data*.

Some of the issues I have found in our dataset are:

- **Quality** issues:
  - Some “a” values in name column and some need to be dropped (“by”, “n”, etc.).
  - Wrong data types for some of the columns, e.g. “tweet_id”, “timestamp”, etc.
- **Tidiness** issues:
  - When I fetched the retweet count and favorite count, I store it in its own table, but this data can be merged with the tweet archive data as they both are observational unit (tweets features).
  - There are 3 predictions for each record (“p1”, “p2”, “p3”) and each prediction column has extra two columns, “value” and “prediction accuracy”, that means I had 9 columns in total, but I could reduce them to only 4.

Cleaning Data

Finally, we"re ready to clean our data, from the missing values, structure and quality issues. I used a variety of Pandas library methods and function in that phase, e.g. head, info, value_counts, etc.

In cleaning phase I have to follow a 3 steps for each issue observation (define, code and test) to fix it.

Some of the issues I cleaned:

- **Missing** issues:
  - There were some missing values in column "name" in tweet archive data, and after checking the original retweets I found that those dogs have no name from the beginning, thus, I couldn't do anything and I decided to replace the string "None" values with "Null" values for all the records in this table.
- **Structure** issues:
  - I joined the two tables tweet archive and retweet and favorite, using merge function.
  - Also, I found that the 4 dog stages (Doggo, Floofer, Pupper and Puppo) are separated in 4 columns, but they are one variable "stage" using "melt" method.
- **Quality** issues:
  - There were a plenty of quality issues in the dataset, e.g. dropping unwanted columns ("source" and "rating_denominator").
  - Modifying wrong or inaccurate values for name column. And re-fetch the correct name from "text" variable if needed.
  - Wrong data types, e.g. I changed "tweet_id" column data type from "int64" to "object" and "number" column data type from "object" to "category" in image predictions table.