Udacity Project
Wrangling and Analyzing Data
"WeRateDogs Twitter Archive"

By Fagr Ahmed, 2020-12-13

Introduction:

The purpose of this project is to put in practice what I learned in data wrangling data section from Udacity Data Analysis Nanodegree program. The dataset that is wrangled is the tweet archive of Twitter user @dog_rates, also known as WeRateDogs. 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. This report briefly describes my wrangling efforts.

Project details:

The tasks of this project are as follows:

- Gathering data
- Assessing data
- Cleaning data

1-Gathering data:

I gathered data from 3 sources, stored in separate files:

1. **Twitter archive file:** WeRateDogs Twitter Enhanced archive, manually downloaded from the Udacity servers.
2. **The image predictions file**, programmatically downloaded from the Udacity servers.
3. **Twitter API & JSON:** The entire set of each tweets' JSON data, downloaded by querying the Twitter API using the Tweepy library.

The favourite_count and retweet_count were extracted programmatically from this file. I loaded the 3 raw data files into separate tables: archive, predictions and json_data.

2-Assessing data:

Once the three tables were obtained I assessed the data as following:

- **Visually**, I used two tools. One was by printing the three entire dataframes separate in Jupyter Notebook and two by checking the csv files in Excel.
- **Programmatically**, by using different methods (e.g. info, value_counts, sample, duplicated, groupby, etc).

Then I separated the issues encountered in quality issues and tidiness issues. Key points to keep in mind for this process was that original ratings with images were wanted.

3-Cleaning data:

This part of the data wrangling was divided in three parts: Define, code and test the code. These three steps were on each of the issues described in the assess section.

First and very helpful step was to create a copy of the three original dataframes. I wrote the codes to manipulate the copies. If there was an error, I could create a new copy from the original.

Whenever I made a mistake, I could create another copy of the dataframes and continue working on the cleaning part. There were a couple of cleaning steps that were very challenging.

One of them was in the image prediction table. I had to create a ‘nested if’ inside a function in order to capture the first true prediction of the type of dog. The original table had three predictions and confidence levels. I filtered this into one column for dog type and one column for confidence level.

Other interesting cleaning code was to melt the dog stages in one column instead of four columns as original presented in twitter archive.

One very challenging cleaning step was when I had to correct some numerators that were actual decimals. This issue was brought to my attention after the first Udacity review. Using Excel and visual assessment was not sufficient to verify those decimals. Therefore, I had to run a code in order to check those actual tweets (decimals numerators).