Data Wrangling Report

Prepared by: Ahmed Mohamed Hussein Unshur

1. Introduction:

The following report presents a data wrangling project that was completed as part of Udacity’s Data Analyst Nanodegree program.

Data collected from the real-world is mostly dirty and messy, which is why it’s important to acquire a number of skills of handling and cleaning such data.

The dataset that was wrangled (and then analyzed and visualized) is the tweet archive of the Twitter account @dog_rates, also known as WeRateDogs.

2. Wrangling Process:

For this project, we have followed the wrangling process of gathering, assessing, and cleaning data.

2.1. Gathering Data.

We have gathered three datasets using three different methods. The datasets gathered were the following:

a. The WeRateDogs Twitter Archive.

The first dataset was provided by Udacity as a CSV file. The file was downloaded manually from the following link:

https://d17h27t6h515a5.cloudfront.net/topher/2017/August/59a4e958_twitter-archive-enhanced/twitter-archive-enhanced.csv

b. The Tweet Image Predictions.

The second dataset was in a TSV file on Udacity’s servers. We have downloaded the file programmatically using the Requests library and the following URL:

https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-predictions/image-predictions.tsv

c. Additional Data from the Twitter API.

Gathering the third dataset required accessing the Twitter API. But instead of requesting a developer account from Twitter to access the API, we have downloaded the following two files provided by Udacity:

i. **twitter_api.py** which is the Twitter API code to gather the required data. After reading and understanding how the code works, we have copied and pasted it into Jupyter Notebook based on Udacity's instructions.

ii. **tweet_json.txt** which is the resulting data from **twitter_api.py.** The file was read line by line into a Pandas dataframe with the following columns: id, retweet count, and favorite count. Then, the dataframe was saved into a CSV file.

2.2. Assessing Data.

After gathering all three datasets, we have assessed them both visually and programmatically. We have displayed each dataset in Jupyter Notebook to conduct visual assessment. For programmatic assessment, we have used Pandas' functions and methods. Assessment focused on quality and tidiness issues.

Assessing the entire datasets completely was not required in this project. We have only assessed 9 quality and 2 tidiness issues.

a. **Quality Issues.**
   - i. **Twitter archive dataset.**
     1. tweet_id is a float instead of string.
     2. timestamp is a string not a datetime datatype.
     3. Non-names in the name column, e.g. a, an, the, by, his.
     4. Dog stage columns (doggo, floofer, pupper, and puppo) are string instead of categorical datatype.
     5. Duplicate tweets (181 retweets).
   - ii. **Image prediction dataset.**
     6. tweet_id datatype is an integer instead of string.
     7. Multiple predictions (p1, p2, and p3).
   - iii. **Data from Twitter API.**
     8. id datatype should be a string instead of an integer.
     9. id column should be labeled as tweet_id.

b. **Tidiness Issues.**
   1. In Twitter archive dataframe, dog stages are in four separate columns instead of one column.
   2. All datasets are from the same observational unit, but they are in three separate dataframes.

2.3. Cleaning Data.

In the final part of the wrangling process, we have cleaned issues documented in the assessments section. But before cleaning, we created copies of the original datasets. Then, we cleaned the datasets programmatically using the define-code-test framework.

For each issue identified in the assessments, we have:

a. Defined a data cleaning plan in writing.
b. Translated definitions into code and then run it.
c. Tested the dataset using code to make sure that the cleaning code worked properly.

We also documented the steps.

After cleaning the issues, the master dataframe was stored in a CSV file named:
twitter_archive_master.csv.

3. Conclusion.

In this report, we have described how we have wrangled dataset of the Twitter user @dog_rates, also known as WeRateDogs. We have used Anaconda, Python and some of its packages and libraries (NumPy, Pandas, Matplotlib, Seaborn, Requests, Tweepy, and JSON), Jupyter Notebook, Sublime Text, and Microsoft Word.