Wrangle and Analyze Data @WeRateDogs

Process Documentation

By

Qusay Elewy

November, 2021

Introduction

In this report, we are going to document our wrangling efforts for WeRateDogs Twitter account's data. That is, we are going to briefly discuss the work done in the three (3) key data wrangling tasks: *Data Gathering, Assessing Data,* and *Cleaning Data*.

Data Gathering

The data used in this project come from three sources listed as follows:

- **twitter_archive_enhanced.csv:** The WeRateDogs Twitter archive data, which is downloaded directly.
- **image_predictions.tsv:** The tweet image prediction file which is downloaded using Requests library.
- **tweet-json.txt:** Provides the tweets' JSON data. Accessed using json library to extract additional data to what was provided in the Twitter archive data file, i.e., retweet_count and favorite_count.

Assessing Data

Both *visual* and *programmatic* assessments have been performed on the three datasets. For the visual assessment, the three datasets have been loaded into three data frames which we viewed and explored in Jupyter as well as MS Excel. And for the programmatic assessment, functions such as describe(), info(), value_counts(), nunique(), and duplicated() have been used.

These three schemas, extracted using info() are listed below.

Archive Dataset

```
Data columns (total 17 columns):
tweet_id                      2356 non-null int64
in_reply_to_status_id         78 non-null float64
in_reply_to_user_id           78 non-null float64
timestamp                     2356 non-null object
source                        2356 non-null object
text                          2356 non-null object
retweeted_status_id           181 non-null float64
retweeted_status_user_id      181 non-null float64
retweeted_status_timestamp    181 non-null object
expanded_urls                 2297 non-null object
rating_numerator              2356 non-null int64
rating_denominator            2356 non-null int64
name                          2356 non-null object
doggo                         2356 non-null object
floofer                       2356 non-null object
pupper                        2356 non-null object
puppo                         2356 non-null object
dtypes: float64(4), int64(3), object(10)
```

Prediction Dataset

```
Data columns (total 12 columns):
tweet_id    2075 non-null int64
jpg_url     2075 non-null object
img_num     2075 non-null int64
p1          2075 non-null object
p1_conf     2075 non-null float64
p1_dog      2075 non-null bool
p2          2075 non-null object
p2_conf     2075 non-null float64
p2_dog      2075 non-null bool
p3          2075 non-null object
p3_conf     2075 non-null float64
p3_dog      2075 non-null bool
dtypes: bool(3), float64(3), int64(2), object(4)
```

Json Dataset

```
Data columns (total 4 columns):
tweet_id          2354 non-null int64
name              2354 non-null object
favorite_count    2354 non-null int64
retweet_count     2354 non-null int64
dtypes: int64(3), object(1)
```

Issues found during this assessment have been divided into two lists: one for quality issues, which are concerned with the content; and the other for tidiness issues, which are concerned with the data structure.

Cleaning Data

A copy of our datasets has been made before commencing the data cleaning process so that we could check our original data anytime during the process.

Define-Code-Test framework is utilized in our work, where each issue was documented along with its definition, the code used to fix it, and the result of testing the changes made.

Summary

Python libraries allowed us to access various data sources and formats, and although there were some tricky issues to fix, Pandas made it fairly easy to access and manipulate our data. Plotting visualizations using matplotlib is not only visually appealing, but it also helps us to understand our data better and draw some interesting insights.

As we can see below, our actions have improved the overall quality and tidiness of our data. We now have a combined dataset that is compact, with the right data types, and with much fewer null values; null values are have been marked as NaNs (instead of “None” and empty cells.)

Master Dataset

```
Data columns (total 17 columns):
tweet_id                 2175 non-null int64
in_reply_to_status_id    78 non-null float64
in_reply_to_user_id      78 non-null float64
timestamp                2175 non-null datetime64[ns]
source                   2175 non-null object
text                     2175 non-null object
expanded_urls            2175 non-null object
rating_numerator         2175 non-null int64
rating_denominator       2175 non-null int64
name                     1391 non-null object
stage                    344 non-null object
favorite_count           2175 non-null int64
retweet_count            2175 non-null int64
jpg_url                  1994 non-null object
probability              1994 non-null object
probability_conf         1994 non-null float64
is_dog                   1994 non-null object
dtypes: datetime64[ns](1), float64(3), int64(5), object(8)
```