Wrangle and Analyze Data @WeRateDogs

Process Documentation Report by Morayo Egbewumi

September 2022

Introduction

Using the data from the WeRateDogs Twitter account, we will describe our operations in this report. To put it another way, we'll quickly go over the activity that goes into the three (3) main data wrangling activities of data gathering, assessment, and cleaning.

Data Collection

The following three sources are where the information for this project was gathered:

-twitter_archive enhanced.csv: A direct download of the WeRateDogs Twitter archive data.

-image predictions.tsv: was obtained by using the Requests library.

-tweet-json.txt: This file contains the JSON information from the tweets. accessed utilising the json library to extract more data, such as retweet count and favorite count, from the Twitter archive data file.

Analyzing Data

On the three datasets, evaluations using both visual and programmatic methods have been carried out. The three datasets were put into three data frames for the visual evaluation, which we then saw and analysed in Jupyter and MS Excel. Additionally, tools like describe(), info(), value counts(), nunique(), and duplicated() have been employed for the programmatic evaluation.
The following list contains the three schemas that info() extracted.

Archive Dataset

```
<class 'pandas.core.frame.DataFrame'>
 RangeIndex: 2356 entries, 0 to 2355
```

```
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

```
RangeIndex: 2075 entries, 0 to 2074
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
favourite_count       2354 non-null int64
retweet_count         2354 non-null int64
dtypes: int64(3), object(1)
```

The issues identified during this evaluation have been separated into two lists: one for quality problems, which relate to the content, and the other for tidiness problems, which relate to the data structure.

Cleaning Data

Before beginning the data cleaning procedure, a copy of our datasets was generated so we could examine the original data at any moment.

In our work, the Define-Code-Test model was applied, and for each issue, the definition, the code used to correct it, and the outcome of testing the modifications made were all recorded.

Summary

Despite having a few difficult problems to resolve, Python modules allowed us to access a variety of data sources and formats, and Pandas made it quite simple to access and handle our data. Not only are plotting graphics using matplotlib pleasing to the eye, but they also enable us to better analyse our data and derive some intriguing conclusions.

The quality and tidiness of our data have increased generally, as we can see from the graph below. We now have a merged dataset that is small, has the appropriate data types, and has many fewer null values; these null values are now designated as NaNs (rather than "None" and empty cells).

Final Dataset

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2175 entries, 0 to 2174
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)
```