UDACITY

DATA ANALYST NANODEGREE

GABRIEL SALVADOR BARÇANTE BARROS

Wrangle Report

Brazil

2021

1. Introduction

Real-world data rarely comes clean. Using Python and its libraries, we can gather data from a variety of sources and in a variety of formats, assess its quality and tidiness, then clean it. This is called data wrangling.

The dataset that I will be wrangling (and analyzing and visualizing) 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. The numerators, though? Almost always greater than 10. 11/10, 12/10, 13/10, etc. Why? Because "they're good dogs Brent".

This report describes the steps that were necessary to gather, assess, and clean data about the WeRateDogs twitter.

2. Gathering Data

In the gather part of the project, I have to work with three different pieces of data, as follows:

- The WeRateDogs twitter archive was handed as a CSV format, and was downloaded manually. This was imported as the DataFrame **archive**.
- The tweet image predictions, i.e., what breed of dog (or another object, animal, etc.) is present in each tweet according to a neural network. The file "image_predictions.tsv" is hosted on Udacity's servers and was downloaded programmatically using the Requests library and this link. This was imported as the DataFrame **predictions**.
- Using the tweet IDs in the WeRateDogs twitter archive, I queried the Twitter API for each tweet's JSON data using Python's Tweepy library and stored each tweet's entire set of JSON data in a file called "tweet_json.txt", with one tweet's JSON data per line. Then, I read that file line by line into a Pandas DataFrame with the tweet ID, retweet count, and favorite count. This was the DataFrame **tweet**.

After gathering the data, I had three Pandas DataFrames that needed to be assess and clean.

3. Assessing Data

So, I have successfully gathered data. Now, I need to assess the data to determine what is clean and potentially what else to gather if I am missing some data. In assessing the data, I'm inspecting the dataset for two things: data **quality** (i.e., content issues) and lack of **tidiness** (i.e., structural issues).

In order to accomplish that, I used the two styles of assessing the data: visual and programmatic. However, regardless of the type of assessment, after each issue that is **detected**, we have always to **document** that issue.

In the visual assessment, I printed each DataFrame in the Jupyter Notebook and looked at over with my own eyes, trying to catch some issues.

In the programmatic assessment, I used code to view specific parts of the data to help me detect problems in the data, like using functions or methods to summarize the data. Since I was using Python in this project, the Pandas' methods that I used were:

- .sample
- .info
- .describe
- .value_counts
- .duplicated (with sum)
- .sort_values
- Various methods of indexing and selecting data (.loc, .iloc and query)

After I'd detected and documented some issues, I organized all those issues by quality and tidiness, as follows:

3.1. Quality issues:

3.1.1. archive table:

- Missing values in the *in_reply_to_status_id*, *in_reply_to_user_id*, *retweeted_status_id*, *retweeted_status_user_id*, *retweeted_status_timestamp*, and *expanded_urls* columns.

- Dogs name as 'a' (validity issue).

- Nulls represented as 'None' in *name*, *doggo*, *floofer*, *pupper*, and *puppo* columns (validity issue).

- Erroneous data type (*dog_stage* - category; *retweeted_status_id*, *retweeted_status_user_id*, *in_reply_to_status_*id, and *in_reply_to_user_id* - integer; *timestamp* and *retweeted_status_timestamp* - datetime).

- Tweet ID 835246439529840640 with the wrong rating numerator and rating denominator.

3.1.2. predictions table:

- The predictions of dog breed (*p1*, *p2*, and *p3* columns) should be formatted as the name of the breed (" " as "_").

- Duplicate images in the *jpg_url* column.

3.1.3. tweets table:

- Erroneous data type (*favorite_count*, *retweet_count* and *id_str* columns - integer).

- Inconsistency in the name of the column referring to tweet id (*id_str*).

- Duplicate rows.

3.2. Tidiness issues:

- One variable (dog stage) in four columns (*doggo*, *floofer*, *pupper*, and *puppo*) in the **archive** table.
- Favorite count and retweet count (**tweets** table) should be part of the **archive** table.
- One variable (dog breed) in three columns (*p1*, *p2*, and *p3*) in the **predictions** table.
- The *dog_breed* column in the **predictions** table should be part of the **archive** table.

4. Cleaning Data

After assessing the dataset, I was ready to clean each of the issues I documented before. The first thing that I had to do was to make a copy of each one of the DataFrames. All of the cleaning operations were conducted on those copies.

Since there are a lot of issues, the data cleaning process was conducted using the *Define-Code-Test* framework in each one of the issues documented in the assessment part.

I started addressing the missing data, which is a completeness issue (quality). The archive table was the only one with missing data in some of its columns. Since one of the key points in this project was to only use original tweets with ratings and images, i.e., no retweets and replies, most of the null values were dropped to deal with that problem.

After addressing missing data, cleaning for tidiness was the next logical step, because tidy datasets are easy to manipulate, and in the context of data wrangling, tidy datasets with data quality issues are almost always easier to clean than untidy datasets with data quality issues.

The data was messy because it was divided in three tables, but each one of these tables have the same observational unit. Moreover, the tweets and predictions tables had one variable in three columns. First, I handled this last problem in each table, so later I could merge the tweets and predictions tables to the archive table.

Once the missing data and tidiness issues are cleaned, cleaning the remaining data quality issues, i.e., the ones outside of the completeness issues, is all that remains.

5. Conclusion

After cleaning, I reassessed the data and I could have revisited any step of the data wrangling process deemed necessary. Since I didn’t think it was, I stored the clean **archive** DataFrame in a CSV file as “twitter_archive_master.csv”.

With a clean data, I can analyze and visualize the data, but always in mind that I could revisit any step in the process any time.