wrangle_report

July 1, 2022

0.1 Wrangling Report

Data wrangling is very important when it comes to working with data.I was able to wrangle the data successfuly by following four crucial steps. The steps are outlined below.

0.1.1 step 1: Gathering Data

Three peices of data were required for this project: `twitter_archive_enhanced.csv`, `image_predictions.tsv`, and `tweet_json.txt`.

The WeRateDogs Twitter archive

I downloaded the twitter_archive_enhanced.csv manually from https://d17h27t6h515a5.cloudfront.net/topher/2017/August/59a4e958_twitter-archive-enhanced/twitter-archive-enhanced.csv. This url was provided by Udacity. After downloading, I uploaded and read the data into pandas DataFrame

The tweet image prediction

The `image_prediction.tsv` file was accessed from the url: https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-predictions/image-predictions.tsv using the **Requests** library

Additional data from the Twitter API

Using the twitter API to gather each tweet's retweet count and favorite("like") count and writing it to `tweet_json.txt` file was challenging for me as it required twitter `API CONSUMER KEYS` and `ACCESS KEYS`. I was denied access to these keys from my twitter developer account several times and so I read the already provided `tweet_json.txt` containing all the tweet information line by line into pandas DataFrame.

0.1.2 Step 2: Assessing Data

After gathering all the pieces of data, I assessed them visually(using excel) and programatically using pandas functions.

The following pandas functions were useful: - .head() and .tail() - .info() - .value_counts() - .duplicated() - .unique() - .isnull()

After the assessment I was able to detect the following **`quality issues`** and **`tidiness issues`**.

Quality issues

From the `twitter archive table` 1. Retweet and reply observations not needed 2. Erroneous datatypes(tweet_id and timestamp) 3. Source is html tag 4. Duplicated tweet_ids 5. The expanded_url has missing URLs 6. Inconsistencies with dog names(a, an, all) for instance are not dog names 7. `text` column with url

From the `image prediction table`

8. Erroneous datatype(tweet_id)

Tidiness issues

From the `twitter archive table` 9. Dog stages has separated columns(doggo, floofer, pupper, and puppo) instead of one. Its violate the principle of **each variable is a column**.

From the `image prediction table` 10. - Dog prediction also spread in different columns(p1, p2 and p3) - Confidence level of prediction of dog images in different columns(p1_conf, p2_conf, and p3_conf) - Whether the prediction is dog or not is spread across different columns(p1_dog, p2_dog and p3_dog) - img_nums not necessary for the analysis

0.1.3 Step 3: Data Cleaning

Cleaning of the data was a bit challenging for me and took me two days to achieve a success. I used Udacity's three stage programmatic cleaning process- **`Define`**, **`Code`** and **`Test`**. All the quality and tidiness isuess were dealt using some of the functions below.

- .drop_duplicated()
- .drop() and .dropna()
- .isnull()
- str.extract() and str.split()
- regex expressions
- lambda function and .apply()

0.1.4 Step 4: Storing Data

With the cleaning process done, all the pieces of the data were merged to form single master DataFrame and was stored in a csv file named `twitter_archive_master.csv`.