Data Wrangling Report

By Muhammad Ghazal

Feb 2021

Before starting of working on the project, I imported the libraries that I might need to start on the project. The libraries are:

- pandas
- numpy
- tweepy
- requests
- re
- json
- matplotlib.pyplot
- datetime
- os
- seaborn
- scipy.stats
- warnings

then I used `warnings.filterwarnings('ignore')` to hide the warnings in my code finally I used `%matplotlib inline` to display plots in my notebook

Data Gathering

In this phase, I needed three files to start working on the project.

1. Firstly, I downloaded the first file twitter_archive_enhanced.csv manually from the project module in the nanodegree program home in my classroom, then I loaded it into my notebook using `pd.read_csv` method.
2. Secondly, I gathered the file image_predictions.tsv by using `requests.get` method to download it from url, os module and `open()` function to write the contents of the file which is downloaded to a csv file then I used `pd.read_csv` method to load image predictions data.
3. finally to gather the api data, I created twitter developer account then I used my api keys, `OAuthHandler` method, `set_access_token` method and tweepy module, then I used `api.get_status` to extract one tweet's id information to ensure that gathering is successful, then I used `open()` function to write the tweets of the api in the third file tweet_json.txt then I used a list and another `open()` function to append the list by the tweets from tweet_json.txt using `json.loads` method finally I created a dictionary that contains a list of tweet_ids, a list of retweet_count and a list of favorite_count then I used `pd.DataFrame` method to convert the dictionary to an API dataframe.

Output:

- twitter_archive_enhanced.csv
- image_predictions_df.tsv
- tweet_json.txt

Data Assessment

Visual Assessment

I assessed the files visually using the spreadsheet application Microsoft Excel

Programmatic Assessment

I used jupyter notebook and pandas functions for assessing programmatically

Output

Quality issues

All the three tables

- Tweets that are replies and retweets not original tweets

`archive` table

- Duplicated urls in `expanded_urls` column
- Wrong data types(id, timestamp, rating_numerator, rating_denominator)
- Tweets have no images
- Null values are represented as string `None` in columns:(`name` , `doggo` , `puppo` , `pupper` , `floofer`)
- Wrong names like: `a` in `name` column
- In `rating_denominator` column they are values that do not equal 10
- In `rating_numerator` column they are very big values like 1776

`image` table

- wrong data type in tweet_id column

`api_df` table

- wrong data type in tweet_id column
- Tweets have no images

Tideness issues

`archive` , `api_df` tables:

- A single observational unit is stored in multiple tables (tweet_id) column

`archive` table

- column headers are values, not variable names (doggo, puppo, pupper, floofer)

`image` table

- Multiple types of observational units are stored in the same table.(p1, p1_conf, p1_dog, p2, p2_conf, p2_dog, p3, p3_conf, p3_dog)

Data Cleaning

The structure of the process of data cleaning in general is: Define, code, test.

1. Define: in this phase, I define the solution of an issue by defining the steps of solving that issue.
2. Code: in this phase, I apply the definition of the solution of the issue by coding, I use python and pandas functions in the code.
3. Test: in this phase, after running the code which is the solution, I use python and pandas functions to ensure that the issue has solved.

The Solutions of the quality issues and tidiness issues: Solutions of quality issues All the three tables

- Tweets that are replies and retweets not original tweets *I Used* `query` *method and* `index` *method to extract indices of replies and retweets in* `archive` *table then use two lists to append them by indices of replies and retweets of images and api tables then I used* `drop` *method to drop them then use* `reset_index` *method to reset indices of the tables finally I used* `drop` *method to drop columns:* `in_reply_to_status_id` , `in_reply_to_user_id` , `retweeted_status_id` , `retweeted_status_user_id` *and* `retweeted_status_timestamp`

`archive` table

- Duplicated urls in `expanded_urls` column *I Used* `values` *method to assign the urls to the variable* `urls` *then I created an empty list to fill it with the urls that are not duplicated then in a* `for` *loop I used* `rfind` *method to find the highest index of ',' in the url then I used that index to slice the url such that the url after the last ',' is appended to the list then I used another* `for` *loop to substitute the duplicated urls with not duplicated*
- Wrong data types(id, timestamp, rating_numerator, rating_denominator) _I Used `astype` method to change dtype of `tweet_id` column to `str` and I changed dtype of columns:(`rating_numerator` , `rating_denominator`) to `float64` finally I used `pd.to_datetime` method to change dtype of `timestamp` column to `datetime` _
- Tweets have no images *I Used* `merge` *method to join* `archive` *table with* `image` *table by left join then I used* `query` *method and* `index` *method to extract the tweets that have no images then I used* `drop` *method to delete them finally I used* `reset_index` *method to reset indices*
- Null values are represented as string `None` in columns:(`name` , `doggo` , `puppo` , `pupper` , `floofer`) *I Used* `query` *method and* `index` *method to extract indices of* `None` *values of columns:(* `name` , `doggo` , `floofer` , `puppo` , `pupper` *) then I used* `for` *loop to assign* `NaN` *value instead of* `None` *values in* `name` *column and empty strings for columns:(* `doggo` , `floofer` , `puppo` , `pupper` *)*

- Wrong names like: `a` in `name` column *I Used `re.compile` method to determine the pattern to use it for extracting the names of the dogs if they exist then I used a `for` loop and inside it I used `try` and `except` to avoid errors and I used `re.findall` method to extract the name from the text, if the name doesn't exist, I add `np.nan`*
- In `rating_denominator` column they are values that do not equal 10 *I Used methods: `re.compile` and `extract` to extract the denominators and the numerators from the text then I used `query` method to extract denominators that do not equal 10 and are multiplied by 10 then I divided them by 10 then using a `for` loop divide numerators and denominators by the quotients of the previous division finally I used `findall` method to fix manually the numbers that are not 10 and can not be divided by 10*
- In `rating_numerator` there are very big values like 1776 _Use `query` method and `index` method to extract indices of wrong numerators then use `drop` method to drop the rows that have ratings:(`24` , `1776` , `420`) because the texts of them don't include the right numerators of them then use `reset_index` method to reset indices of `archive` table then use `re.findall` method inside a `for` loop to extract the right numerators from the texts finally use another `for` loop to assign the right numerators in place of the wrongs in `archive` table_

`image` table

- wrong data type in tweet_id column _Use `astype` method to change the data type of `tweet_id` column to `str` _

`api_df` table

- wrong data type in tweet_id column _I Used `astype` method to change the data type of `tweet_id` column to `str` _
- Tweets have no images *I Used `merge` method to join `api_df` table with `image` table by left join then I used `query` method and `index` method to extract the tweets that have no images then I used `drop` method to delete them finally I used `reset_index` method to reset indices*

Solutions of tidiness issues

`archive` , `api_df` tables

- A single observational unit is stored in multiple tables (tweet_id) column _I Used `merge` method with inner join to join two dataframes: `archive` and `api_df` on the column: `tweet_id` _

`archive` table

- column headers are values, not variable names (doggo, puppo, pupper, floofer) _I Used `add` method to group the four columns: (`doggo` , `pupper` , `puppo` , `floofer`) into one column then I used `values` method to extract the values of that column then I used `assign` method to add that column to the `archive` table whose name is `dog_stage` then I used methods:(`query` , `index`) to extract the indices of the rows that have empty string in `dog_stage` column, then I used a `for` loop to substitute the empty strings by `np.nan` , then I fixed manually the values that has more than one stage that are merged without space like doggopuppo using `query` method to extract them and using a `for` loop and `values` method in it I added '-' between them like doggo-puppo finally I used `drop` method to drop them_

`image` table

- Multiple types of observational units are stored in the same table.(p1, p1_conf, p1_dog, p2, p2_conf, p2_dog, p3, p3_conf, p3_dog) _Create a list that contains new names for `image` df then assign it to the column headers using `columns` method finally use `pd.wide_to_long` method to reshape the dataframe_

Output

- twitter_archive_master.csv
- image_predictions_master.csv