Project 4: Wrangle Report

Omar Bougacha

April 2020

1 Introduction

In this project, we are supposed to practice the skills learned related to data wrangling process. The application is to wrangle data about the WeRateDogs twitters. As it was covered by the lessons, the data wrangling process consists in gathering data, assessing it's quality and tidiness, and fixing the pointed issues. The objective behind such a process is to obtain a clean and tidy dataset that will allows to answer the analysis questions.

This report presents the effort I made during this project and a summary of the performed actions to fulfill the data wrangling objective. The remaining of this report is organized in the data wrangling process. In section 2, I present the methodology used to gather the data. Thus, I present each used technique of the process and the corresponding tools used. In section 3, I present the data assessment process for each data frame and I point out the detected issues. Section 4 contains the cleaning procedure with its documentation. Finally, I conclude this report with a summary of the results and what have I learned during this project.

2 Gather

This section presents the process of gathering the data. The process could be achieved through several techniques:

- Download data manually then load it.
- Download data programmatically then load it.
- Use API to access Data.
- Load data from JSON Structure.

Each of these techniques is presented below:

2.1 Reading Data From CSV File

In this technique, the project managers (in this case Udacity team) provided us with an already collected and regrouped data in a CSV file. The file is downloaded manually and saved in the project directory. Then, the data in the CSV file is loaded into a dataframe using the **Pandas** module and its **read_csv** function. Thus, we obtain a dataframe named **twitter_archive_df**. All codes and results are presented in the **wrangle_act.html** file.

2.2 Download Data Programmatically

In case we have several files to download and to load to create our data, it might become absurd to download each one manually. To automate this procedure and to help others to reproduce our work, we can download the data files programmatically. In this technique, we used the **requests** module to get the data from the **URL**. We create the file using the write binary **wb** option then save the into the file using the write function. Once the file is saved in the project directory we load it into a pandas dataframe using the same technique as previously.

2.3 Gathering Data with API

There is another way to gather data effectively and allow to reproduce the results and the tasks. This way is by using the application programming interface or API. Most website provides this sort of access to help accessing the data. In this case, we are supposed to use this technique to gather the tweets about WeRateDogs and save them into a text file. However, the API for twitter (Tweepy API) is not an open source API. We need to get approved by the Twitter developers to get this access. Normally, the request takes about few days to get approved. However with the Covid-19 situation it become a bit difficult to get the access. I applied for the access it was more than 15 days ago and I still didn't get approved. So I gave up waiting. Thankfully, the Udacity teachers provided us with another option. It is downloading manually the resulting text file and then use it directly. So I chose this option. However, I took a look to the Tweepy code and I understood what it is supposed to do. In the **wrangle_act.html** file I explained each instruction of the code. Actually, it is quite intuitive.

2.4 Reading Data From JSON Structure

One common used gathering technique is reading JSON structure and extract the data from them. For this technique, I used the **json** module to manipulate the JSON structures and the **StringIO** function from the **io** module. The process is simple, we have a text file in which each line is a json structure. So we read this text file line by line of simply get all the lines in a structure using **readlines()**. Each line is a string. We change its type to an StringIO and load a json table from that io string. Afterwards, we just need to access the attributes we want and store them in a dictionary. Each json structure will be converted to a dictionary. All the dictionaries are gathered in a list that will be used to create a pandas dataframe. The code of this procedure is presented in the **wrangle_act.html** file.

Of course, these are not the only techniques that could be used to gather data. We can access SQL databases, we can collect data from html pages etc.. This conclude my gathering data part. In the end of this process I collected 3 dataframes:

- twitter_archive_df
- image_pred_df
- tweets_json_df

3 Assess

In this step of the wrangling process, we want to assess the issues presented in the collected data. Data could present several issues that are related to the collection process or to the entry process. Issues of data are classed into two categories:

- Data quality issues: "dirty data". These issues could be subdivided into several groups. There is no consensus about the number of groups but most commonly we can find:
  - Data Accuracy: Data entries should present the right values in the right consistent and unambiguous form.
  - Data Consistency: Data entries for the same feature/column should be consistent and have the same representation. This should help also cross-referencing records from different tables.
  - Data Validity: The data values should be conformed to a predefined schema. They should satisfy the constraints.
  - Data Completeness: The data entries do not present missing values or missing records across the tables.
  - Data Uniqueness: The data does not present duplicated records.
  - Data Timelessness: Or Currency, means that the collected data is up-to-date.
- Data tidiness issues: "untidy/messy data". The data tables should verify three conditions to be considered tidy:
  - Each variable is a column,
  - Each record/entry/observation is a row,
  - Each type of observational unit forms a table.

Assessing data consists in detecting this issues. For this purpose, two methods are used:

- Visual assessment: In this method, we will look at the data in the file using generally excel or other methods (we can use the pandas functions). Then we will try to identify the issues using only what we can see. I used this method in my assessment process. It helped my identify the presence of missing data, the presence of some data presentation issues, and some tidiness problems all the details are presented in the **wrangle_act.html** file.

- Programmatic assessment: Not all the issues could be detected using only our eyes. And when the data becomes large it becomes difficult to investigate its quality manually using only our eyes. Therefore, we can use some programming techniques to detect these issues. We can use several function from the pandas library such as **isnull()**, **info()**, **nunique()**, **duplicated()**, and **value_counts()** or accessing directly the attributes of the dataframe using **shape** or **dtypes**. I used several of this techniques to further investigate the issues. All the details are documented in the **wrangle_act.html** file.

To summarize, the combination of these techniques allowed me to detect the following issues. I did not investigate all possibilities since it was only instructed to detect 8 quality issues and 2 tidiness issues. The presented below issues could be exclusive or there could be more issues:

- **Data Quality:**
  - twitter_archive_df:
    - *Missing Values:* 2278 missing value in each the 'in_reply_to_status_id' and 'in_reply_to_reply_id'
    - *Missing Values:* 2175 missing value in each the 'retweeted_status_id', 'retweeted_status_user_id' and 'retweeted_status_timestamp'
    - *Missing Values:* 59 missing value in the 'expanded_urls'
    - *Accuracy:* The 'in_reply_to_status_id' and 'in_reply_to_user_id' should be integers since they hold the id of the original status and the user id of the original status.
    - *Accuracy:* 'timestamp' should be a datetime type.
    - *Accuracy:* 'retweeted_status_id' and 'retweeted_status_user_id' should be integers since they hold ids.
    - *Accuracy:* 'retweeted_status_timestamp' should also be transformed to datetime type.
    - *Accuracy:* Source should be a categorical variable (should be represented with categorical data type). And the values should be ('Twitter for iPhone', 'Twitter Web Client', 'Vine - Make a Scene', and 'TweetDeck').

        - *Accuracy:* Several anomalies in the name column should be fixed. (a, an, and this etc) are not dog names.
    - image_pred_df
        - *Missing Values:* 281 missing record.
        - *Accuracy:* p1, p2, and p3 should be categorical.
    - tweets_json_df
        - *Missing Values:* 2 missing record.
        - *Accuracy:* 'tweet_id' should be of integer type.

- **Data Tidiness:**
    - The stage of the dog values are presented as columns.
    - The twitter_archive_df presents the tweet information and the dog information
    - three tables are used to describe only two observational units.

4 Clean

Once the issues about the gathered data quality and tidiness are detected they are to be fixed. The cleaning process of the data is an important task and should be done carefully. The documentation of this process is also important so that the results could be reproduced and one can understand how the final data is obtained.

The process of cleaning is documented through 3 steps:

- **Define:** Here, one defines the exact operation he is going to perform on the data. The definition should be concise and to the point. Using a verb to describe the action and mentioning the details.

- **Code:** Here, one writes the code used to perform the action described. The code should be clear and well commented.

- **Test:** The done operations should be tested to see the results of the performed operation. Further definition and coding could be required.

The process of cleaning data is not sequential it could require some looping to solve the problem. Details of each operation I've done to clean the data are well documented in the **wrangle_act.html** file.

5 Conclusion

Throughout this project, I learned how to wrangle data. I also discovered by application the importance of this process and its fundamentals. I also discovered that although the process might seem sequential, it requires a lot of looping work to obtain clean tidy data.

In this project, I had the chance to start a basic application on data wrangling process. I worked on several techniques of data gathering, assessment, and cleaning. I had the chance to practice most of the new learned skills. And I'm excited to get to use them in real life applications.