DATA WRANGLING REPORT

By Oluwatosin Durodola

The purpose of this project is to put in practice what I have learned from the Data Wrangling section in Udacity Data Analyst Nanaodegree program. The dataset that is been used is the tweet archive @DogRates, 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." WeRateDogs has over 4 million followers and has received international media coverage.

Project Goal:

The goal of this project is to effectively wrangle data related to dog ratings. The data was provided by WeRateDogs, in which they downloaded their Twitter archive and sent it to Udacity via email exclusively to be use in this project. This archive contains basic tweet data (tweet ID, timestamp, text, etc.) for all 5000+ of their tweets as they stood on August 1, 2017. Once we have effectively gathered, assessed, and cleaned our data in this project, it can be used for our analysis.

This report briefly describes my wrangling effort.

Project Steps Overview

The tasks performed in this project are as follows: >

> Step 1: Gathering data
>
> Step 2: Assessing data
>
> Step 3: Cleaning data
>
> Step 4: Storing data

Gathering Data

The dataset used for this project consisted of three different datasets that were obtained as follows:

`twitter_archive_enchanced.csv` : This data was provided in the project guideline. I downloaded it then uploaded it to my project directory through the jupyter upload function in the jupyter notebook. Then I imported pandas and numpy and load it through pandas' read_csv.

`image_prediction` : The url to get the image_predictions was provided by udacity. I imported the Python `requests` library, then assign the link to url and with the get() function of the requests library, I got the data through its url and saved it in a response variable. I used the Python `with open` function to write the response's content to a `tsv` file in the same working directory, then read it with pandas's read_csv with additional argument which is sep='\t' because it was a tsv file.

`tweet-json.txt` : I make use of the tweet-json provided by udacity because my Twitter APi was not approved and the udacity APi wasn't working for me, it just keeps failing.

I read the txt file with pandas' read_json with argument `inline=True` then I extract and rename the columns I need using the Python `with open` function and a `for loop` , I read the `tweet_json.txt` line by line and loaded each line as `json` file. I saved each tweet_id, retweet_counts and favorite_counts, which I later converted to a dataframe named `tweet_json` .

Assessing Data

After gathering, I moved to assessing which is the next step after gathering, I assessed the data as follows:

Visually: I previewed the three dataframes individually in a jupiter notebook and scrolled through them for assessment.

Programmatically: I also assessed the dataset programmatically with pandas methods and functions such as .info(), .describe(), .isnull(), .head(), .tail(), .sample(), .duplicated(), .value_counts() ets.

At the end I was able to detect the quality issue and tideness of the gathered data which are as follows:

Quality issues

Twitter_archive

1. Missing records in the following columns; 'in_reply_to_status_id', 'in_reply_to_user_id', 'retweeted_status_id', 'retweeted_status_user_id', 'retweeted_status_timestamp', expanded_urls
2. Erroneous data type (`timestamp` column is in obj instead of datetime)
3. Erroneous data type (`tweet id` column is in int instead of string/object)
4. Inappropriate value for `source` . It should be Twitter for Iphone, Twitter Web Client, Vine - Make a Scene, TweetDeck
5. some name are not consistent like some 'a', 'the', 'on' etc, they are mostly in lowercase and None also is expressed as nan

image_predictions

1. Erroneous data type (`tweet id` column is in int instead of string/object)
2. Unify mixed writing of dog breeds in image_predictions, some lower case, some title case etc.

tweet_json

1. Erroneous data type (`tweet id` column is in int instead of string/object)

Tidiness issues

Each variable is a column Each observation is a row Each type of observational unit is a table

1. One variable in four columns (doggo, floofer, pupper, puppo) in `twitter_archive` table (dog_stage)
2. One variable in two columns (rating_numerator, rating_denominator) in `twitter_archive` table (rating)
3. `twitter_achrive` , `image_predictions` and `tweet_json` should be one table because they all have characteristics of tweets.

I also listed the columns' description of each dataset.

Cleaning Data

I followed the steps principle that was given to use which are `Define` , `Code` and `Test` .

Each steps with short brief of the issues stated in the `assessing` section.

Then, I made a copy of the original three datasets using the pandas copy function which were list below:

- **twitter_archive_clean**
- **image_predictions_clean**
- **tweet_json_clean**

Then, I followed the `Define` , `Code` amd `Test` process and made the following cleaning efforts:

- I drop six columns (in_reply_to_status_id', 'in_reply_to_user_id', 'retweeted_status_id', 'retweeted_status_user_id', 'retweeted_status_timestamp', expanded_urls) with missing value which are related to retweet id since it was not needed for our anaylsis.

Then after I drop those with missing values, I moved on to clean those with tidness issues.

- I join the four dog stages spread across four columns into one single column using concatenation, first replacing none with empty string. I dropped the four dog stage columns and also drop rows with more than two dog_stage value.
- Since rating was given to columns I decided to replace them with one column by dividing rating_numerator by rating_denominator to given me rating column. I then round them to one decimal place and drop those greater than 1.5
- I merge the three datasets as one using the pandas merge function assigning it df.
- I converted the timestamp column of the df from object to datetime.
- I converted the tweet_id column of the df from integer to string.
- I replace the hyperlink of the source to iphone, tweetdeck and web using the loc function.
- I replace all values in the name column that started with small letters with nan including those with none because it was confirmed that those names weren't dog names.
- I changed all p1, p2, and p3 values to title case and remove the underscore from those that have it.

Storing the Data

After gathering, assessing and cleaning the data, I saved the merged data 'df' in a csv file named `twitter_archive_master.csv` .

Conclusion

This project was interesting, I wish I would work on it again to do more cleaning. There are so things I would like to clean to due to time constraints. It was challenging but I am glad to be able to do it despite the errors I encountered.

Indeed, the world data is dirty and untidy. I will keep on improving my skill using Python programming language and other available tools to equiped myself.

Thanks for the opportunity