wrangle_report

July 19, 2022

1 Data Wrangling Report

The dataset that we wrangled in this project is the tweet archive of the 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 always almost have a denominator of 10.

1.1 Project Goal

The goal of this project is to effectively wrangle (i.e. gather, assess and clean) the WeRateDogs Twitter data in order to create worthy analyses and visualizations. This report briefly describes my wrangling efforts.

1.2 Project Task

The tasks for this project are as follows:

- Gathering Data
- Assessing Data
- Cleaning Data

1.2.1 Gathering Data

For this project, we gathered three different datasets which were obtained using three different methods.

- The first dataset was the `Twitter archive data` which was provided by Udacity. I downloaded it manually on my local machine through a link `twitter_archive_enhanced.csv`, after downloading, I uploaded it to Jupyter Notebook Workspace and loaded it into a pandas dataframe named `twitter_df` after importing the `pandas` libraries.

- The second dataset was the `image_predictions.tsv` file which was hosted on Udacity's servers and I downloaded it programmaticaly using the `Requests` library. I imported Python `requests` and `os` libraries. With the get() function, I got the data through its url and I got the output as `Response 200` showing that it was successful. Then with Python `with open` function, I wrote the response content and then I read the downloaded tsv file into a pandas dataframe named `imagepre_df`.

- The third dataset was the `tweet-json.txt` which was supposed to be obtained by creating a twitter developer account to query Twitter's API but personally I encountered some issues while doing that so instead I downloaded two files provided `twitter_api.py`(Twitter API code to gather some of the required data for this project which includes favorite_count and retweet_count etc.) and `tweet_json.txt`(the resulting data from the twitter_api.py). Once these files were downloaded, I created an empty list named `tweets_list` then I read the `tweet_json.txt` line by line into a pandas dataframe. Then I appended each line to a list of dictionaries and then converted it to a dataframe.

1.2.2 Assessing Data

For this project, it was required of us to detect 8 quality issues and 2 tidiness issues.

After the three datasets has been downloaded, I assessed the data with the following methods:

- **Visual Assessment:** I visually assessed the datasets in Jupyter notebook using pandas dataframe, scrolled randomly(top, bottom,left,right) to check for issues. Also, I made use of Google sheets in order to view the parts that has been collapsed in pandas.
- **Programmatic Assessment**: I assessed the datasets programmatically with the use of various python and pandas methods. They include .head(), .tail(), .info(), .describe(), .value_counts(), .isnull() and .duplicated().

1.2.3 Cleaning Data

This part of the data wrangling process requires the `Define`, `Code` and `Test` approach. These three processes were carried out on each of the issues detected in the assess section.

First, I made a copy of each of the datasets - twitter_df_clean= twitter_df.copy() - imagepre_df_clean= imagepre_df.copy() - tweets_df_clean= tweets_df.copy()

Following the `Define`, `Test` and `Code` process, I made the following cleaning efforts.

- I dropped the `expanded_urls` columns which had missing urls since we were unable to obtain the missing urls.
- I dropped other columns `in_reply_to_status_id`, `in_reply_to_user_id` with missing data that does not apply to original tweets.
- Also, I dropped columns related to retweets`retweeted_status_id`,`retweeted_status_user_id`,`retweeted_` since we only want original ratings.
- I merged using the python melt() function all the dog stages into one main column named `dog_stages` in the `twitter_df`.
- I merged the `tweets_df` with the `twitter_df`.
- I converted the names in the predictions column (p1, p2, p3) that began with lowercases to uppercases.
- I converted the datatype of `timestamp` column from `object` to `datetime`.
- I replaced the invalid dog names in the `name` column with 'None'.
- I extracted the string text of the tweet's source that was mixed up with html amd url codes in the `source` column.
- I dropped tweets with `rating_denominator` NOT equal to 10.

1.2.4 Storing Data

After gathering, assessing and cleaning, I saved the master dataset to a csv file named `twitter_archive_master.csv`.

1.2.5 Conclusion

This project helped me practice my Data Wrangling skills and Yes, I encountered some errors and issues but it's also part of learning. *Data Wrangling is one of the core skills that are required of a data analyst to be very familiar with.* Also, I was able to hone my Python programming skills and learn more about Python's libraries and packages with which I was able to successfully wrangle, analyze and gain insights.