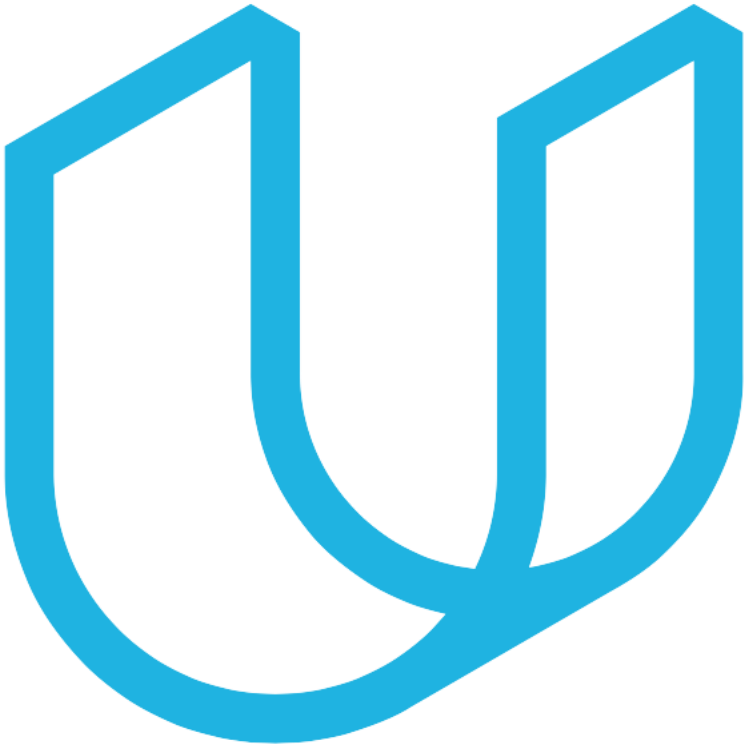
Project 5: Wrangle and Analyze Data (Wrangle Report)

Data Analyst Nanodegree Project



Date 19th August 2020

By Wisatat Komwatcharapong

Wrangle Report (by Wisatat Komwatcharapong)

Introduction

Project Details

Real-world data rarely comes clean. Using Python and its libraries, I had to gather, assess, and clean the data, in order for it to be used for analysis and visualization. Fully assessing and cleaning the entire dataset would require exceptional effort so only a subset of its issues (eight quality issues and two tidiness issues at minimum) needed to be assessed and cleaned. The tasks for this project were:

1. Data wrangling, which consisted of:
    - Gathering
    - Assessing data
    - Cleaning data
2. Storing, analyzing, and visualizing the wrangled data
3. Reporting on my data analyses and visualizations (act_report.pdf)

Gathering

The data for this project was in three different formats and they were obtained as mentioned below: Twitter Archive File – WeRateDogs: This was extracted programmatically by Udacity and provided as twitter_archive_enhanced.csv to use.

Image Predictions File: The tweet image predictions, breed of dog present in each tweet according to a neural network. This file (image_predictions.tsv) was hosted on Udacity's servers and downloaded programmatically using the Requests library

Twitter API & Tweet JSON File: By using the tweet IDs in the WeRateDogs Twitter archive, which casue me all sort of problem as it was tweet_json.txt file using twitter API for @WeRateDogs. I have requested the twitter developer account, but this is not approved yet. I may use the provided tweet_json.txt in this project instead of the twitter api.

Accessing

After gathering the data, the three tables were saved and assessed Visually and Programmatically. With both the assessments I looked for Unclean data i.e Dirty data with content issues and messy data with structural issues. Generally, I looked for Tidiness and quality issues.

Quality Issues

twitter_archive

1. Replace None with Nan
2. Data contains 181 retweets (rows where retweeted_status_id, retweeted_status_user_id and retweeted_status_timestamp have a number instead of NaN)
3. Erroneous datatype (tweet_id, timestamp). in_reply_to_status_id, in_reply_to_user_id, retweeted_status_id and retweeted_status_user_id are erroneous datatype, but we don't use these.
4. source in HTML format is difficult to read
5. Incorrect names or missing names in name column, for instance, a, an, the - all are written with lower case letters
6. The denominator columns have invalid values.
7. The numerator columns have invalid values.
8. The dog names format should be consistent. (First letter capital for name column)

df_image
1. Lowercase breed names in p1, p2, p3 and '_' is used instead of space.
2. Erroneous datatype (p1, p2 and p3 should be categoral datatype).
3. 66 duplicate jpg_url

Tidiness Issues

1. Merging 3 dataframes.
2. doggo, floofer, pupper and puppo columns can marge in 1 column.
3. rating_numerator and rating_denominator merge into 1 column.
4. p1, p2 and p3 columns and confidence columns can marge in 1 column.
5. Remove unwanted columns.

Cleaning

Used basic python function i.e. duplicates , drop, sort , value_count ,describe , info and others to comply with above mentioned point. I struggle with few issues and had to spend a lot of time to get my understand. As little help was provided its first time I used so many websites for checking syntax and possible solutions.

This part of the data wrangling was divided in three parts as per the data sets and was further divided as per the three steps Define, code and test. I have provided a headline and code and test blocks below it to make is easy for understanding.

First and the most important step was to create copies of all the three data frames. So that I can do trial and errors in the copy frames rather than the originals. I replaced the “None” to “NaN” to get dropped easily. I changed the datatype of timestamp, rating_numerator, rating_denominator, p1, p2 and p3. The standard for "rating_denominator" is 10, but on checking we found that it includes some other numbers, which could be the mis parse. So, I check the text corresponding to those ratings and noticed that few of them were analyzed incorrectly due to the presence of another fraction in the text. I corrected the same for both rating denominator and numerator. I melted doggo, floofer, pupper and puppo in dog_stage columns, and droped stage column, because it doesn't need as many columns as the original dataset

I found out that there were 66 duplicate values for jpg_url i.e. same url was added to the data multiple times. So, I dropped the duplicated data. I also changed the column names to make it more descriptive and readable. Again, the dog breeds of the all the three prediction columns involved both upper and lowercases for the first letter. I rectified the same to make it consistent

Conclusion

Data wrangling is one of the key skills in analyzing, debugging, and cleaning up data for easy analysis.
I have practiced this skill with the Python library which is one of the most powerful analytical tools, and this skill can be applied in many fields, not only for data analysis.