The WeRateDogs Project Data Wrangling Report

by Z. Jiang
May, 2020

This report is a brief summary of the wrangling procedures applied in completing the ‘WeRateDogs’ project as part of Udacity’s Datalyst Nanodegree.

The data wrangling procedures applied in this dataset are listed below.

1. **Gathering**
2. **Assessment**
3. **Cleaning**

I will summarize the data wrangling procedures used in this project below follow the above sequence.

1. Gathering

The original dataset does not include the ‘retweets’, ‘favorites’, or the neural network identified dog breeds information. The missing information need to be gathered from different sources.

The original dataset ‘twitter_archive_enhanced.csv’ was directly downloaded from the provided url link which was the file consisting of the largest amount of data.

The second dataset was obtained using a Python library Tweepy through querying Twitter’s API for the additional data on the tweets in the archive file using the tweet id. The data was returned by the Tweepy library in json format, from which it was possible to iterate through and appended to a file as a list of dictionaries and then a pandas data frame. A copy of the file was saved in csv format as the data frame created.

The third dataset was programmatically downloaded via Python Request library. The file contains the image predictions of the dog breed made by a neural network on some of the tweets downloaded in the archive file. The file was in tsv format.

2. Assessment

The three data frames were assessed both visually and programmatically inside a jupyter notebook with pandas. The functions called for the programmatical assessments including df.info(), df.head(), df.sample(), df.value_counts().

Both quality and tidiness issues were identified in the original three data frames. One markdown cell named ‘Assessment’ in the ‘wrange_act.ipynb’ file has been designated to the summary of both quality and tidiness of the datasets.

The quality issues are related to the content of the data. Various validity, accuracy, completeness, and consistency related issues in the three data frames were identified and summarized under ‘Quality issues’ inside the ‘Assessment’ markedown cell.

The tidiness issues related the data structure based on the criteria that each variable should form a column, each observation should form a row and each type of observation should form a table.

The assessment of the original data sets led to the conclusion that all three data sets should be integrated into a single data frame. Besides, the columns which missing too much information and irrelated to the analysis can be excluded.

3. Cleaning

The last step of the data wrangling process is data cleaning to fix the tidiness and quality issues identified in the assessment step. The cleaning of each identified issue was applied by the standard define, code, and test process. The detailed cleaning procedures have been recorded in the ‘wrangle_act.ipynb’ file.

The data cleaning step was mainly completed using programmatic approaches including some of the pandas built-in functions (melt, merge, etc).

Note: The three required changes suggested by the mentor have been made. Please find the brief summaries of the changes below.

1. The ‘retweets’ in the archv_cln datafram has been removed through `archv_cln = archv_cln[archv_cln[‘in_reply_to_status_id’].isnull()]`

2. All the incorrect dog names in the ‘name’ column have been cleaned through identifying the lower case first letter of the incorrect names.
Code: `archv_cln[‘name’] = map(lambda x: x[0].isupper(), archv_cln[‘name’]`

3. Extract ratings properly. The ‘rating_numerator’ column’s datatype was converted to ‘float’ first, then all the rows with the rating_numerator extraction issue have been listed and the rating number was fixed one by one.

Save the cleaned data

The data wrangling resulted in a clean data frame for later data analysis and visualization. The cleaned data was saved to ‘twitter_archive_master.csv’ file. As the mentor suggested, the`‘index=False’` argument in `‘to_csv()’` function has been applied to avoid writing the row number to the final csv file.