Data Wrangling Report

By Mohamed Khoukha
January 2021

As an assignment for the Udacity Data Analyst Nanodegree; This report illustrate the main steps involved in the data-wrangling of Twitter account “WeRateDogs”.

Data Gathering

In this step, collecting data takes place. For this project, there were three main sources for the data to deal with:

1. Twitter_archive_enhanced.csv file, this file was delivered by email and downloaded manually to our working directory and then imported into our working environment using Pandas function “pd.read_csv”.
2. Image_prediction.tsv is the second file that has been hosted on a webpage downloaded from its relevant URL using the requests library get function and pd.read_csv pandas’ function.
3. The final dataset was gathered from REST API via the tweepy library by querying the API to obtain extra information pertinent to the tweets’ ids in the first file, e.g. retweets count and favorite count aspects.

Data assessment

in this step, we investigate our imported datasets both visually and programmatically for quality and tidiness issues.

- The visual assessment done on spreadsheet application such as excel and then the programmatic assessment is conducted in Jupiter notebook.

- Missing data were addressed first then messy structures were addressed to facilitate the tackling of the rest of the quality issues.

-Several issues were detected and listed below:

Quality Issue (issues with content)
1. twitter_archive_df:
1.1 Only want original ratings (Delete the 181 retweets and 78 replies)
1.2 Don't need those columns: 'in_reply_to_status_id', 'in_reply_to_user_id', 'retweeted_status_id', 'retweeted_status_user_id', 'retweeted_status_timestamp', 'img_num', 'expanded_urls' and 'jpg_url'
1.3 All rating_denominator should be "10" and some rating_numerators are extreme values
1.4 Since all the denominator is 10 after last step, we can get rid of rating_denominator column and change rating_numerators to 'rating'
1.5 Many dog names are meesed up, such as "such" "a" "quite"
1.6 timestamp have extra "+0000"
1.7 timestamp's datatype should be converted to "datatime"

2. img_predictions_df:
2.1 Remove "_" and capitalize the image predictions.(p1, p2, p3 column names)

Tidiness Issue (issues with structure)

0. Join 3 DataFrames.
1. twitter_archive_df:
1.1 Dog stage's 4 variables: doggo, floofer, pupper, puppo should be in single column of categorical variable
1.2 Dog stage have 'None' instead of np.nan

2. img_predictions_df: 2.1 Image prediction should be summarized to one column 'dog_breed'

Data Cleaning

It was performed chronologically for the most parts, ensuring that tidiness issues are mostly addressed first.

Issue number	Issue type	Solution
0	tidiness	Inner join twitter_archive_df_clean, img_predictions_df_clean, and api_df_clean on tweet_id
1.1	Tidiness	Create 'dog_stage' variable which is made by extracting the dog stage variables from the text column
1.2	Tidiness	Dog stage have 'None' and replace 'None' to np.nan
2.1	Tidiness	Use the ture prediction to fill in dog_breed column. If no ture prediction, fill in use np.nan
1.1	Quality	Select the rows from twitter_archive_df that retweeted_status_id and in_reply_to_user_id columns that is null
1.2	Quality	Remove columns: 1.in_reply_to_status_id, 2.in_reply_to_user_id, 3.retweeted_status_id, 4.retweeted_status_user_id, 5.retweeted_status_timestamp, 6.img_num
1.3	Quality	Drop rows where denominator of rating != 10 and where numerator rating >> 10
1.4	Quality	Drop rating_denominator column
1.5	Quality	We find all the incorrect names have lowercase first letters. We will change those names to None, then change all the None to np.nan
1.6	Quality	Use *str.strip* to remove "+0000"
1.7	Quality	use pd.to_datetime convert timestamp's datatype
1.8	Quality	Use regular expression and Series.str.extract to find real source between tags > and <
2.1	Quality	Use Series.str.replace to remove '_' and use Series.str.capitalize to convert 'p1' 'p2' 'p3'

After cleaning, I stored the clean df in CSV file with name 'twitter_archive_master.csv'.

The process was enhanced using various sources from the internet, such as from Stack Overflow, Google, and Pandas.