Udacity Project | Wrangle and Analyze Data

Wrangle Report

Junghoon Suk

Overview

The main objective is to gather WeRateDogs Twitter datasets and transform them into clean and tidy data enough to produce a reliable and meaningful analysis. Raw datasets are given internally or to be scraped on the web through an official Twitter API.

Project Workflow

As per project guideline, the project includes:

- Gathering
- Assessing
- Cleaning
- Storing
- Analyzing and visualizing data ***(not part of this report, refer to act_report.pdf)***

Gathering data

There were three sets of data available in different forms:

- ***'twitter_archive_enhanced.csv':*** internally provided as a .csv file that is yet dirty and consists of core information (i.e. twitter url, tweets, etc.) and saved as a dataframe 'tweet_archive.'
- ***'Image_predictions.tsv':*** internally shared with a url. It consists of information like dog images and breeds predicted using neural network. Used ***requests.get*** method (then written as a .tsv file) and saved in a dataframe, 'image_prediction.'
- ***'tweet_json'txt':*** it is recommended to pull information using Python's Tweepy library through the official Twitter API and save in .json format. However, due to errors in accessing the Twitter API, the text file (json.format) internally provided by Udacity was used instead. 'tweet_id','retweet_count' and 'favorite_count' idata are saved in a dictionary and then in 'tweet_count' dataframe.

Udacity Project | Wrangle and Analyze Data

Assessing and Cleaning data

Through both visual and programmatic methods, 9 data quality and 3 tidiness issues were spotted and cleaned as below. Assessing and cleaning procedures can be iterative and sometimes reassessment was made to deepdive or change direction (marked as **reassessed**)

Original data frames were copied for the cleaning process, and named as 'tweet_archive_clean', 'image_prediction_clean' and 'tweet_count_clean'.

Tidiness Issues

T1. `*tweet_archive_clean*` *table:* doggo, floofer, pupper, puppo are one variable (dog_stage), spread over multiple columns.

- changed 'None' to ' ' (blank) first, and concatenated the 4 columns into a new column 'stage'
- **reassessed:** separated double entries (i.e. doggopupper) with ' , ', and replaced blanked data (' ') with np.nan for future data use
- dropped the previous 4 columns after creating 'stage' column

T2. `*tweet_count_clean*` *table:* same observational unit as *`tweet_archive`_clean'*

- inner merge *`tweet_archive_clean`* and *`tweet_count_clean`* tables on *'tweet_id'* and shorten the table name *`tweet_archive_clean`* as ***`tweet_clean`*** for convenience. inner merge to drop as tweet_ids that do not contain the two key variables (retweet_count, favorite_count), but still check if the result is significant after the merge.

T3. `*image_prediction_clean*` *table:* share image and breed information

- left merge *`tweet_clean`* to *`image_prediction_clean`* on *'tweet_id'*. row data that do not have image_predictions information would be dropped as a result

Udacity Project | Wrangle and Analyze Data

Quality Issues

tweet_archive_clean' table

Q1. Missing data in 'expanded_urls' column (2297 vs 2356)

- After merging, there were two rows that have null values. As they don't have information about 'retweet_count' and 'favorite_count' (our key data), dropped the 2 rows.

Q2. in_reply_to_status_id, in_reply_to_user_id columns: 78 rows with non_null data indicates the tweets are actually 'replies'

- Dropped the rows with non-null values as quality may not be as great as the original tweets.[1]
- And then removed 'in_reply_to_status_id', 'in_reply_to_user_id' columns with no data entries.

Q3. retweeted_status_id, retweet4ed_status_user_id, retweeted_status_timestamp : 181 rows with non-null data are 'retweets'

- We are interested in getting original tweets (no retweets) so removed the 79 rows with non-null values 'in retweeted_status_id', 'retweeted_status_user_id', 'retweeted_status_timestamp' column.
- Then, dropped the three columns as they should be filled with unnecessary 0 non-null values - checked before removing them

Q4. erroneous data types

- *'tweet_id'*: used astype method to transform into string
- *'retweet_count', 'favorite count'*: used astype method to transform into int
- *'timestamp'*: used to_datetime method to transform into datetime object. And inspected if all date are not beyond August 1st, 2017 (< '2017-08-01')

---

[1] Inspired by a Udacity Knowledge thread
https://knowledge.udacity.com/questions/79100

Udacity Project | Wrangle and Analyze Data

Q5. ‘source’ column with tag

- **reassess:** before applying the regex to all, tested it with a few text first. Also, compared before and after counts by using value_counts() method to check if all entries are still correctly placed
- Created a regex to pull <a href> tag out of the text, and replace them with blank(' ').

Q6. some values in 'rating_denominator' are not equal to 10

- **reassess1:** inspected again all the rows with 'rating_denominator' != 10, and 'text' column.
- Used a regex expression to pull those with more than one rating value ([\d]+/[\d+]) from the text, then replace.
- **reassess2:** Created a new column with the calculated rating (rating_numerator / rating_denominator) and name it as 'rating_cal' [2]
- Compared a rating shown in the 'text' column, and removed irrelevant data, and removed ‘rating_cal’ column that was used for cleaning purpose

Q7. incorrect dog names (i.e. a, an, the, None)

- **reassess:** Some names that contain 'None' are not related to dogs: 'We only rate dogs', 'don't send' in text indicate that they are jokes so drop them first.
- replaced 'None' in `name` column with np.nan, and 'a', 'an', 'the' in `name` column names with np.nan, and dropped rows that contains 'not a dog'

Q8. multiple or unrelated urls in 'expanded_urls' column

- created a new list to include only one url that contains 'https://twitter.com' (and all those entries with more than one url are to be dumped)
- created a new column 'twitter_url' and drop the 'expaanded_url'

---

[2] Inspired by a Udacity Knowledge thread: https://knowledge.udacity.com/questions/110847

`image_prediction_clean` table

Q9. some rows contains images that are not dogs [3]

- Created two lists (breed, conf) and use for loop to iterate p1 -> p2 -> p3 to append ‘correct’ breed of the dog in 'breed' list and its confidence rate in 'conf' list
- **reassess:** needed using .loc to slice the data through the iteration but the index was messy now. So used .reset_index( ) to re-organize it first.

Storing data

- Stored the cleaned dataframe (‘tweet_clean’) in ‘‘twitter_archive_master.csv’.

Summary

Many of the data quality and tidiness issues were handled as part of this wrangling efforts. However, there are also some limitations identified for the future data collection and analysis as:

- The data are constrained until August 1st, 2017.
- Dog stages and names columns consist of substantial amounts of missing data, as many tweets don’t include them. For the purpose of the following analysis, dog name data are not quite necessary, so it is accepted fine. However, dog stage analysis is thought to be included, so more data with stage information may give better insights.

[3] Inspired by a Udacity Knowledge blog: https://knowledge.udacity.com/questions/219635