Wrangle Report

By YIJIE ZHANG

Introduction

The dataset that I wrangled is the tweet archive of Twitter user @dog_rates, also known as WeRateDogs. It is a Twitter account that rates people's dogs with a humorous comment about the dog. These ratings almost always have a denominator of 10. The numerators are almost always greater than 10.

Gathering data

The data for this project come from three different dataset:
1. **Twitter archive file.** WeRateDogs downloaded their Twitter archive and sent it to Udacity via email exclusively for us to use in this project. This archive contains basic tweet data (tweet ID, timestamp, text, etc.) for all 5000+ of their tweets as they stood on August 1, 2017. I downloaded twitter-archive-enhanced.csv manually.

2. **Twitter API & JSON.** Using the tweet IDs in the WeRateDogs Twitter archive, I queried the Twitter API for each tweet's JSON data using Python's Tweepy library and store each tweet's entire set of JSON data in a file called tweet_json.txt file. Then read this .txt file line by line into a pandas Data Frame with tweet ID, retweet count, favorite count and other information.

3. **Image Predictions File**. I used URL information and request data programmatically from Udacity's servers and save it as a tsv file(image_predictions.tsv). This is a table full of image predictions (the top three only) alongside each tweet ID, image URL, and the image number that corresponded to the most confident prediction.

Assessing data

Visual Assessment. I print the three tables in Jupyter Notebook in order to notice the issues of the data. If the table size is too large, I opened them in Excel to check the full content of the tables.

Programmatic Assessment. I used python code to evaluate the data quality. Then I wrote down these quality issues and tidiness issues for further data cleaning.

Quality issues:
1. Data completeness.

There are missing values in some columns in tweet_archive table. For example, in_reply_to_status_id and in_reply_to_user_id have only 78 non-null values. retweeted_status_id, retweeted_status_user_id and retweeted_status_timestamp have 181 non-null values. Some values are missing in columns like "doggo","floofer" and "name". These missing data is represented by None instead of NaN. Some original ratings have no image data.
2. Data validity issue. In twitter_archive table, the last few rows show some dog names are 'None', 'a', or 'an.'
3. Data accuracy issue. In twitter_archive table, timestamp is an object and retweeted_status_timestamp is also an object (the other retweeted statuses are floats).
4.Rating data consistency issue. There are three columns including information about rating: text, rating_numerators and rating_denominators which are sometimes not consistent. It produce confusement for data users
5. Rating number validity. There are some strange outliers in columns. From coding(twitter_archive.rating_denominator.value_counts()),there are 23 rating_denominators that are not equal to 10. Similarly, there are rating_numerators that are even bigger than 100.
6. Erroneous data. pupper, puppo, floofer and doggo column: There are some IDs with more than one dog stage information (two dog stages are rated).
7.Duplicated values. From coding (sum(image_prediction.jpg_url.duplicated())),we know there are some duplicated values in this column.
8. Data types are not consistent. In tweet_json table, tweet_id is object type. This is not consistent with data type in other 2 tables (int64 data type),which is not suitable for further data analysis.
9. Table size do not match. There are 2356 rows in twitter_archive, while 2075 rows in images_prediction.The size of the table do not match.
10. Timestamp accuracy. Timestamp column is object instead data time. The values in this column include year, month, day, time and number like "+0000". These numbers are mixed together. "+0000" does not make sense.
11. Dog breeds prediction consistency. The dog breeds prediction information can be found in columns p1, p1_dog, p2, p2_dog, p3 and p3_dog in image_prediction table. These value are not consistent and convinient for further analysis.

Tidiness Issues
1. Columns issue. Each variable forms a column. In twitter_archive table,the last four columns all relate to the stage of dogs (dogoo, floofer, pupper, puppo).
2. Untidy data. In twitter_archive table, column source data have <a> and /a> tag in HTML format surrounding the text. These are not common internet address for ordinary users.

3. Uppercase and lowercase mixture. In images prediction table, there is no consistency in p1, p2 and p3 columns. Sometimes the dog breed prediction are lowercase, sometimes they are uppercase.
4. Many tables VS one table. Tidiness requires each type of observational unit forms a table. These three tables can be merged into one master dataset to improve data tidiness.

Cleaning Data

Solving the data issues found above would facilitate further data analysis.
First of all, I created copies of the three original data frames. If there was an error during the cleaning process, I could come back and create a new copy from the original files.

Then I deleted unnecessary data from the tables in order to simplify the further data analysis. These deletion include retweets and some columns like "in_reply_to_user_id" and "expanded_urls" etc.

A difficulty part of cleaning is change the original columns and values. For example, the dog stage information were scattered in four columns which was not convenient for further process. I used code to extract related dog stage information from tweet texts and put the data into a new column.

Another challenging cleaning step is to correct rating information. Due to some typos or errors, the rating numerators and rating denominators have lots of confusing numbers. By using Excel and Python code, I examined the texts and compare them with numbers, correcting number as much as I can.

Furthermore, I adjusted timestamp information, corrected dog names and clarified the dog breeds data.

Finally, by merging these into one data frame, I got the suitable data table for storing, analyzing, and visualizing Data.