We Rate Dogs: Udacity Data Wrangling project - Twitter Data.

by Yukti Sareen, Apr 2019.

As part of the data wrangling section of the Udacity Data Analyst Nanodegree program, I worked on the WeRateDogs twitter data. WeRateDogs has a culture of unique rating system where the numerator is generally more than 10, for example, 14/10 - which means to say that the dog in question is cuter than a 10/10.

The data wrangling efforts involved 3 stages,

1. Gathering of data
2. Assessing of the data
3. Cleaning of the data

We will look at each of the stages in more detail below.

1. Data gathering

This stage involved collecting data from three different sources. The first was a comma separated file given as part of the project's instructions. This dataset acted as the the twitter master data for the tweets in question with attributes like tweet_id, text, url, etc. The name of the file is twitter_archive_enhanced.csv.

The second set of data used in this project was available at a web server address provided as part of the project instructions. We could visit the location programmatically and download the data into a local file to be used in our project. I used the requests python library to read the contents in chunks to avoid clogging the memory. I downloaded the content to a tab separated file, and saved as image-predictions.tsv. This dataset had analytical data of images that were classified as dogs or not in 3 different attempts with confidence ratings of each prediction.

The third set of data involved scrapping the twitter api using the python tweepy library for additional information on the tweet ids that we had as part of the other two data sets. The additional information is attributes like favorites, and retweets. This could be used to gauge the popularity of the said tweets in the community. This data was saved as tweet_data.txt.

2. Assessing the data

After gathering the datasets as mentioned above, I imported them into data frames to be used in assessment of quality and tidiness issues in the data. I used data frame methods like info(), head(), value_counts() to view a summary or sample of the data at every stage. I was able to find a few quality issues in the 3 sets of data and also some tidiness issues which I would upon resolution could better help in the analysis.

Quality issues refer to issues or inconsistencies in the actual content of the data. We view items like outliers, inconsistent values, null values, missing data when looking for quality issues, as these problems could skew our analysis when used in our aggregations or models.

Tidiness issues on the other hand refer to the structural problems (or room for improvement) of the data which when resolved could help in approaching the data better in advanced stages of the analysis. Here is a summary of our assessment of the data sets.

Quality

1. Data was gathered from 3 different sources - file, file server, api. The no. of rows in each set are different. Not all data for the tweet_ids in the main set was retrieved from other sources.
2. The tweet data has retweets.
3. Underscore char found in many dog categories (dog names).
4. Rating denominator value has many issues. Numerator values also have outliers and values that don't fit the system.
5. 5 columns in the main data set have no values at all.
6. The dog predictions data set has many rows where one of the predictions is true, but the image is not of a dog.
7. Cannot use the url fields.
8. Dog stage has no value for many dogs.
9. Some rows have more than one dog stage. This will need to be cleaned to use dog stage in analysis.

Tidiness

1. Dataframes can be combined. 3 can be combined into 2. The tweet_enhanced data can be combined with the api data as both have information about the tweets. We can keep the predictions data set separate and use a tweet_id join for analysis.
2. Dog stage could have been 1 column with 4 possible values instead of 4 different columns and many nulls.

3. Cleaning of the data

The last stage of the data wrangling before we went to discovering knowledge from the data was to clean the data based on the points laid out above. We used python methods like astype - to force data types of attributes, manually view and clean data for some fields, get rid of null columns as they wouldn’t contribute in the analysis. We also deleted some records which were irrelevant to the assignment like retweets which we did not need.

I would like to highlight one cleaning activity where observed that there were mistakes in the numerator values for a few tweets where the text meant a value that was not reflected in the rating_numerator column. For example, in text of one of the tweets, the tweet mentioned

11.27/10 but the column had 27 as the numerator value. I picked these tweets and fixed them manually. In case of this example, I set the numerator value to 11.

Also, for data types, we set the numerator and denominator to float type.

In order to address the tidiness points, we used functions like merge - to merge the twitter dataset and the api dataset together, and melt - to combine the 4 dog_stage columns into one to avoid too many null values. The step by step cleaning activity is present in the notebook of this project.

When melting the dog stage columns into one dog_stage field, we observe that there are 12 rows with multiple values for stage, like, doggo and pupper on the same row. In our cleaning step, we converted occurrence and nonoccurrence to 1 and 0 respectively. With this, we simply sum values of the 4 columns and a value of more than 1 indicates the tweet is one of the bad 12 tweets. We simply remove these 12 tweets from our dataset so we are left with tweets that have one or no value for dog_stage.

Conclusion

At the end of our data wrangling efforts, we end up with 1 data set after

- merging the twitter data and api data using an inner join
- And merging the first set with predictions data using a left join.

The merged dataset is saved in twitter_api_joined.csv