Data Analyst Nanodegree

Wrangle and Analyze Data : WeRateDogs wrangle report

Manon Philippot

March 18, 2019

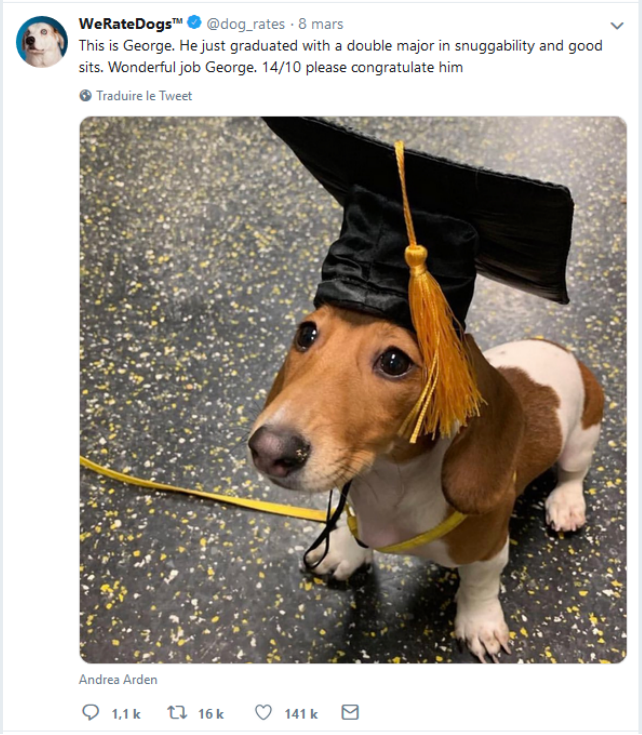
Contents

1 Introduction

The purpose of this project was to practice the concepts learned in the data wrangling course section from the Udacity Data Analyst Nanodegree.

This work is organized in three steps : gather data, assess data and clean data. Each of these steps is documented in this document.

The data comes from WeRateDogs, 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, though? Almost always greater than 10. 11/10, 12/10, 13/10, etc. Why? Because "they're good dogs Brent." WeRateDogs has over 4 million followers and has received international media coverage.



WeRateDogs™ @dog_rates · 8 mars

This is George. He just graduated with a double major in snuggability and good sits. Wonderful job George. 14/10 please congratulate him

Andrea Arden

The goal of this project was then to wrangle WeRateDogs Twitter data to create interesting and trustworthy analyses and visualizations !

2 Wrangling steps

2.1 Gathering data

I gathered data from 3 different sources :

- WeRateDogs Twitter Archive : contains basic tweet data for all 5000+ of their tweets, such as the tweet's text, the rating, the dog name, and the dog "stage" (i.e. doggo, floofer, pupper, and puppo).

- Tweet Image Predictions : contains a table full of image predictions (the top three breeds of dogs only) for a tweet, alongside each tweet ID, image URL, and the image number that corresponded to the most confident prediction (numbered 1 to 4 since tweets can have up to four images).

- Twitter API : contains additional data such as retweet count and favorite count.

2.1.1 WeRateDogs Twitter Archive

The twitter archive was a .csv file. This file was supposed to be a file on hand, so I loaded it locally.

2.1.2 Tweet Image Predictions

The Tweet Image Prediction file was a .tsv file. This file was hosted on Udacity's servers so I downloaded it programmatically using the Requests library.

2.1.3 Twitter API

I queried the Twitter API with the Tweepy Access Library. As this API requires authentication, I had to register on the Twitter Developer Website in order to be granted access tokens. I queried all the tweets whose ID was present in the archive and saved it locally. When loaded again in the Jupyter Notebook, I chose to only keep the *tweet_id*, *favorite_count*, *retweet_count* and *display_text_range* (number of characters of the tweet) columns.

2.2 Assessing data

I assessed the data in two ways :

- Visually : I displayed the entire dataset looking for easily detectable quality and tidiness issues.

- Programatically : I used the *.info*, *.describe*, *nunique()*, *.query()* and *.sample()* methods.

I encountered a total of 9 quality issues and 2 tidiness issues. Of course there are more issues, but these were the most important found for any later analysis.

2.2.1 Quality issues

Twitter Archive :

- some posts are not original ratings that have images (they are retweets)

- the type of the timestamp column should be a datetime object and not a string

- the rating_numerator and the rating_denominator column should be a float and have erroneous values (ex : if the numerator is a decimal number, the decimal part only is taken as the numerator; some values are not read well in the tweet). 7 tweets have been identified with a wrong numerator and/or denominator.

- the source column contains HTML tags and should be categorical data.

- some columns are not useful for our analysis : everything related to retweet or reply.

Tweet Image Predictions :

- not the same number of rows as the Twitter Archive table (some entries seems to be retweets, same jpg_url for multiple tweet_id)

- inconsistent capitalization in p1 / p2 / p3 (some values start with a capital letter, others with a lowercase letter)

- some columns are not useful for our analysis : we should keep only the prediction with the highest probability corresponding to a dog.

Twitter API :

- not the same number of rows as the Twitter Archive table (some entries seems to be retweets)

2.2.2 Tidiness issues

Twitter Archive :

- The doggo / floofer / pupper / puppo columns should be combined in a single column with categorical data. If a dog have more than one stage, the stage should then be defined as "multiple".

General :

- All tables should be in the same dataset.

2.3 Cleaning data

First of all, a copy of each dataset was made in order to not work on the original data and also to always have access to the uncleaned data at any time.

Then, each issue was fixed by cleaning the code in three steps :

- Define : Convert the assessments into cleaning tasks by writing little how-to guide. Serves also as documentation so others can look at the work done and reproduce it.

- Code : Produce code and run it to clean the data.
- Test : Always testing (visually or programmatically) to make sure the cleaning code works.

The easiest part was probably the simple conversion of data type, the removing of unnecessary columns or the capitalization of the first character of a column. The most challenging part was probably to combine the dog stages into one column as it requiered more thoughts, and find the prediction with the highest probability corresponding to a breed of dog among several values. It was also nice to practice the skills acquired with BeautifulSoup to retrieve HTML tag content. A major constraint was that only tweets with an original rating and an image should be kept in the final dataset. We then had to check this condition in each dataset. I also really tried to test every portion of my cleaning code. After all these steps were completed, the 3 datasets were merged into one large master dataset.

The final dataset was made of 1994 rows and 16 columns.

3 Storing data

After the cleaning step was done, the final dataset was saved with the to_csv method.

4 Conclusion

The Data Wrangling process is an important step that helps to analyse our data.

There is an important set of tools and techniques that helps gather, assess and clean our data. Python and its libraries is only one of them.

It is then important to take the time to visually and programmatically analyze any dataset before starting an analysis.