ND111 Project 02 - Data Science II

Wrangle and Analyze Data

Anderson Hitoshi Uyekita

30 December 2018

Contents

Wrangle Report

Synopsis

I have found several problems with dog's name, probably the regex used to gather is not well calibrated, and in many cases has gathered articles, nouns, etc.. I have also found several problems with rating_numerator due to the lack of standard way to rate the dogs.

In respect to the data analysis, I have observed a seasonality in the frequency of tweets, the user @dog_rates tend to tweets more in the begining of the week, monday and tuesday specially, and i have also identified seasonality along the year, there are much more tweets in december and november, going in opposite way these two month have the lowest rating_numerator.

In regard to the algorithms used to predict the dog's breed, I have realized the three algorithms has results very distinct, after a visual investigation plotting a graphic without sucess I have used the Correlation Map to found my insights.

1. Introduction

This project aims to combine data set from Twitter (from the user `@dog_rates`, as known as WeRateDogs) and two other sources to build a new data frame, to do so it is necessary to perform the entire process preconized by the Data Wrangling, which will be describe in details in the next chapters.

2. Data Gathering

I have "downloaded" the files `image_predictions.tsv` and `twitter_archive_enhanced.csv` using the `requests` package. For the additional info from twitter I have used the `tweepy` package to access the @dog_rates tweets.

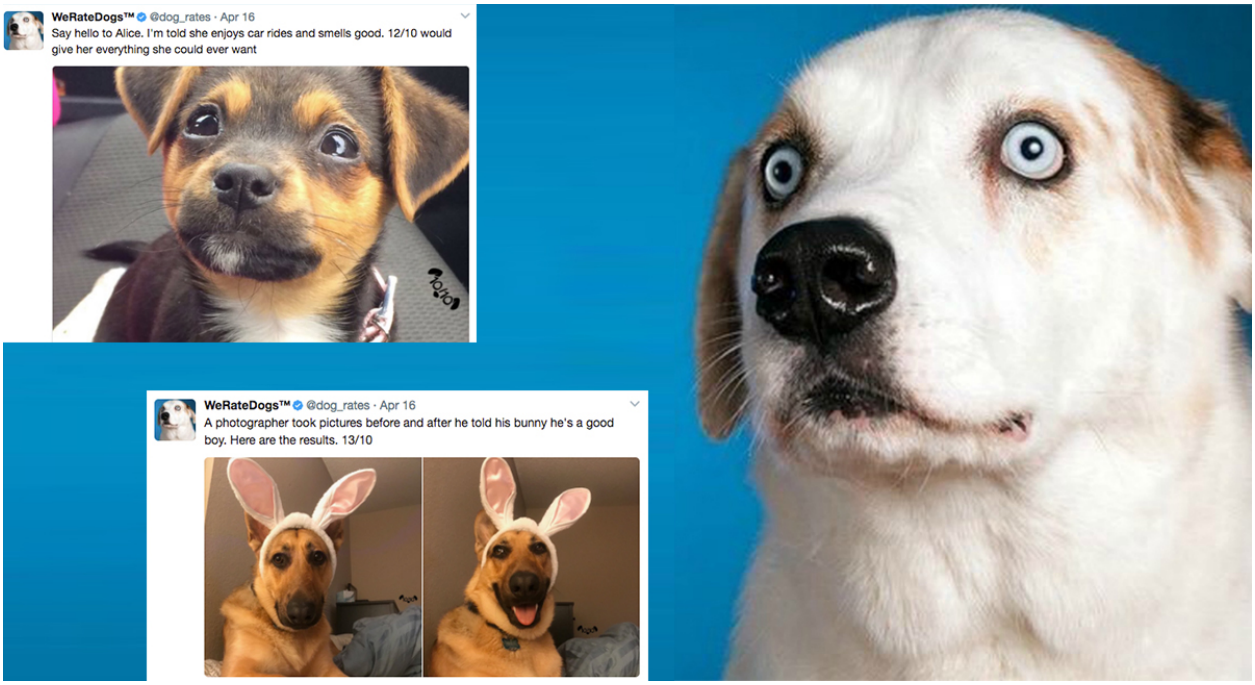

Figure 1:

3. Data Assessing

The Data Assessing process have found several issues, which will be shown in Table 1:

Table 1 - Summary of Issues Identified.

Issue ID	Table	Issue Type	Dimension	Method	Column	Description
1	df_ach	Quality	Validity	Visual	name	Invalid names or non-standard names.
2	df_ach	Tidiness	-	Visual	source	HTML tags, URL, and content in a single column.
3	df_ach	Quality	Validity	Programmatic	rating_numerator	Invalid ratings. Value varies from 1776 to 0.Data Strucutre must be converted from `int` to `float`.
4	df_ach	Quality	Validity	Programmatic	rating_denominator	Invalid denominator, I expected a fixed base.Data Strucutre must be converted from `int` to `float`.
5	df_ach	Tidiness	-	Programmatic	doggo, floofer, pupper, and puppo	This is a categorical variable and could be combine into one column.
6	df_ach	Tidiness	-	Programmatic	text	There are two information in a single column. Split the text from the URL.
7	df_ach	Quality	Validity	Programmatic	timestamp	Converted to date.
8	df_ach	Quality	Validity	Programmatic	tweet_id	Following the example of zip code, it must be string.
9	df_ach	Quality	Accuracy	Programmatic	retweeted_status_id	The same dog could be recorded twice or more in cases of retweets.

Issue ID	Table	Issue Type	Dimension	Method	Column	Description
10	df_ach	Quality	Accuracy	Programmatic	in_reply_to_status_id	The same dog could be recorded twice or more in cases of reply.
11	df_img	Quality	Consistency	Visual	p1, p2, and p3	Dog's breed is not standardized.
12	df_img	Quality	Validity	Programmatic	tweet_id	Convert it to string.
13	df_img	Quality	Validity	Programmatic	jpg_url	Duplicated images and consequently double entry.
14	twt_ach_mstr	Tidiness	-	Programmatic	-	Merging these two tables (`df_ach` and `df_img`) into one.
15	df_img	Quality	Completeness	Programmatic	"retweet count"	Gather additional info in `tweet_json.txt` file.
16	df_img	Quality	Completeness	Programmatic	"favorite count"	Gather additional info in `tweet_json.txt` file.
17	twt_ach_mstr	Quality	Validity	Programmatic	"many columns"	Remove `in_reply_to_status_id`, `in_reply_to_user_id`, `retweeted_status_timestamp`, `retweeted_status_id`, and `retweeted_status_user_id`.

Legend:

- df_ach: twitter_archive_enhanced.csv
- df_img: image_predictions.tsv
- twt_ach_mstr: twitter_archive_master.csv

4. Data Cleaning

Most of the issues involving wrong or non standard values were solved using a tailored regular expression, which allow me to fix it finding in the `text` column the original values. In respect to the data type problems, it was fixed using the `.astype()` method and `.loc()`. Finally, I have solved the tidiness issues combining two tables in one, and merging 4 columns (doggo, pupper, puppo, and floofer) into one so-called dogtionary.

In regard to the duplicated rows and "depricated" columns, I have removed to turn the final dataset much cleaner.

The data frame have started with XX rows and end up with YY rows. Have in mind, retweet_count and favorite_count do not have in all tweet_id, which means there are observation with NaN in these two rows.

5. Conclusions

Although I have written/documented 17 issues, the final file (twitter_archive_master.csv) is not totally free of issues, because I faced the Data Wrangle as an iterative process, what I did so far was the first iteration.

Additional Info

For further information about the UD111 - Project 02:

- ND111 - Project 02 - Repository (Github Repository)
- ND111 - Project 02 - Wrangle Act (Jupyter Notebook File)

- ND111 - Project 02 - Act Report (Markdown File)
- ND111 - Data Science II - Nanodegree Repository (Github Repository)