Wrangle Report

Introduction

The following report explains the steps of cleaning the WeRateDogs Twitter dataset. This dataset contains basic tweet data (tweet ID, timestamp, text, rating, etc.) of more than 2500 dog tweets written by Twitter user @dog_rates.

Gather

Gathering data encompassed three parts:

1. Importing the WeRateDogs Twitter archive as file on hand
2. Getting Image Predictions from a URL using the Python requests library
3. Getting each tweet's retweet count and favorite ("like") count, and any additional interesting data by looping through the tweet-json.txt file and creating the Pandas dataframe line by line.

Assess

The dataset was assessed both visually (using a spreadsheet application as well as Pandas) and programmatically using pandas functions. Following issues, concerning both quality and tidyness were identified:

Quality:

- There are retweets
- There are replies
- There are posts with no pictures (no expanded urls)
- There are tweets with pictures that do not contain dogs
- Dog names include wrong names such as (a, the, an, ...)
- There are different predictions of breeds which should be merged to one confident prediction per dog
- There are mistakes in the dog ratings (i.e. floats 13.5/10 which is interpreted as 5/10)
- There are names which were interpreted as "None" even though there exists a name in the text
- Dates are not timestamp format
- Sources are embedded in HTML a-tags and should be extracted
- Tweet ID's should be converted to string in case of leading zeroes and for compatibility when joining
- Dog breeds have underscores and first letters are not capitalised

Tidyness:

- Dog stages have a bulky format and should be converted to true/false
- There are different guesses and probabilites for the breed which should be merged into one into a format that adhere to the tidy data a column named breed and the most probable prediction of the Neural Net that contains a dog.

Misc:

- There are unneccessary columns which can be removed

Clean

All issues except from one were cleaned programmatically. Cleaning was performed by following the three steps Define, Code and Test. As a first step, the twitter dataset was duplicated with Python's `copy()` function.

Replacements were done using e.g. RegEx patterns. Cleaning the neural network prediction was special. In order to to get one classification result wether the picture contains a dog or not, a rule was applied to the three predictions.

Given the assumption that the posts most probably contain a dog picture, all three predictions were taken into accounts by applying the rule that if at least of the predictions says the picture includes a dog, then the picture should be classified as a dog picture. The breed with the highest probability was subsequently extracted. This approach proved quite robust, after testing the outcomes manually. One good example of a successful classification is a picture featuring a dog which was taken through a doughnut hole. The doughnut was (wrongly) classified as an orange but the second-most confident prediction voted for a dog. Adhering to the rule, the picture is classified as a dog in the clean dataset.

Tests were conducted mostly using the Python `assert` statement in order to check, whether the data was cleaned in the right way.

As an extension, after extracting dog names in a rule-based manner, remaining unrecognised dog names were extracted using the 3-Class Stanford Named-Entity Recognition (NER) Tagger on the first sentence of each post. The code can be found in the `clean_ner.ipynb` notebook.

Discussion

Out of the 2356 original observations, 1666 remained after cleaning the dataset which is a reasonable data loss. Following limitations were found during the approach: Extracting names in a rule-based manner can become arbitrarily complex. E.g. using the first match does not work in the case of the post

```
This is Bluebert. He just saw that both #FinalFur match ups are split
50/50. Amazed af. 11/10
```

Of course, it would be possbile to write a more complex rule which guesses the most probable rating out of the two but this would become quite complex. This quality issue was cleaned manually but manual cleaning becomes quite tedious on large dataset and is not recommended.

Using the tagger as an additional instance after applying the rules on extracting dog names took a relatively long time and less than 40 additional names were extracted. However, the approach scales better than manual on larger dataset and was more fun to implement than extracting each name by hand 😉.

All in all, a reasonably cleaned dataset has been created which enables further analyses on the data.

Conclusion

The data was prepared according to the three steps Gather, Assess and Clean. The cleaning step included quality issues as well as untidy data and was divided into definition, code and testing of each data issue.

Possible limitations of the chosen cleaning approach were discussed in the last section and a cleaned dataset ready for analysis was provided.