Wrangling Efforts

for the Project WeRateDogs

By Christoph Knoop

Step 1: gathering the necessary data

The **Twitter library of WeRateDogs** made it rather easy. The operators provided us with the document, the conversion with pd.read_csv() into a DataFrame turned out to be unproblematic.

Reading **in the results of the neural network** was a little more complex. It was available as a web resource, but could easily be downloaded with requests. For the conversion of the data into a dataframe I used the Python library "io" with the method io.StringIO.

The third data set obtained via the **Twitter API** was the biggest challenge. First, taking into account the download limit, the data record had to be downloaded using the existing tweet ids. This took about 30 minutes. The query itself produced two error messages: first, the API could not find some IDs, probably the tweets were deleted. I solved this problem with a try-except-statement. The second error message concerned exceeding the download limit. I could work around this error by setting the parameters of the Twitter API method to "wait_on_rate_limit=True, wait_on_rate_limit_notify=True".

The results of the API query were first written to a txt file as JSON. The advantage of caching in a text file: I only had to perform the tedious API query once, then I could access the text file as data source.

Step 2: assessing the data

To check the data, I use the usual tools in pandas like head(), info(), shape, value_counts etc.. After each data record I created a small list of quality and tidiness problems. Beside wrong formats and missing values I noticed with the names of the dogs the unusually many short names. A closer examination revealed that words such as "the" or "a" were often erroneously identified as names. I also identified the tidiness problems, for example the different columns in the neural network table. Basically I made the decision to form two tables from the tables in the cleaning process. The first should contain all twitter-relevant information, the second all information related to the dog (breed, name etc.). Another decision: The rating system of WeRateDogs usually uses a denominator of 10, but in some rare cases the page deviates from this. For the analysis I decided to use only those data that work with a denominator of 10 in the evaluation to prevent outliers.

Step 3 – Cleaning the data

At the beginning there was the treatment of the tidiness problems. In all data sets I first deleted the variables I didn't need for the analysis like the numerous retweet variables. Next, I converted the variables that deal with the dog-type. Instead of several columns I formed the column dog_type, in which the corresponding dog-type was entered as value. If there was no name, the cell assumed the value "None".

After that it was the turn of the dog breed. Here we had up to three possibilities through the neural network that had been identified on the respective images, including a probability of whether the result actually applies. There are different approaches here: One could only consider the results that are reasonably certain, for example with a probability of 80 percent and above. The problem: This would remove most of the data concerning the dog breed. I therefore decided on a somewhat more inaccurate path and transferred the best result of the neural network in which a dog breed was found into the data set as the respective dog breed. In order to be able to evaluate even more precisely in a later analysis, I have included the probability value for the respective result in the data set. Before I started with the revision of the quality errors, I merged the files into two dataframes.

During the revision of the quality errors I changed formats such as integer or floats, and I also converted the time of the respective tweet into a datetime format. Another step was to delete rows from the data set that were not to be included in the analysis, especially retweets.

The dog names were about identifying wrongly read names. In the course of the analysis it became apparent that not only words such as "a" and "the" were falsely identified as names. All names that do not begin with a capital letter turned out to be wrong on closer inspection.