Data Wrangling of Twitter Account “WeRateDogs”

The wrangling process was separated into three different phases: gathering, cleaning and assessing. The following report will dive into each of those points and will give an insight of the work which hast been done on the Dataset.

Gather

There were three different sources to be gathered. The first one was the *twitter-archive-enhanced.csv*, which was locally stored in the jupyter-workspace. This file was red to an dataframe using the pandas-function *pd.read_csv()*. The second source to be gathered, were image predictions store in an tsv- file available under the following URL: *https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-predictions/image-predictions.tsv*

This tsv-file was downloaded programmatically and saved locally using the *requests* and *os* library.

The last source needed to be gathered using the Twitter API and the library *tweepy*. To use the API it was necessary to sign up for API usage via Twitter, to get a consumer key, a consumer secret, an access token and an access secret. With these information’s and the tweet id’s out of the *twitter-archive-enhanced.csv* more information’s per tweet could be gathered as json-file for each tweet. All extractable json-files were saved to the text file *tweet_json.txt*

After gathering there were three DataFrame’s available. *df_twitter_archive* containing the information of the *twitter-archive-enhanced.csv, df_image_prediction* cointaining the information of the programmatically downloaded tsv-file and df_tweepy containing information’s of the Json-Files gathered via the API.

Assess

Assessing the DataFrames was done with programmatic approaches like *.head() .info()* *.vlaue_counts()* and visual approaches by opening the files in the text-editor or importing the .csv- and .tsv-files into Microsoft Excel. During this Process the following issues were discovered, separated by quality and tidiness issues:

Quality Issues:

- Tweets, which are reply’s or retweets, should not be included in the analysis: twitter archive
- reply’s and retweet columns are unnecessary for the analysis
- Some Rating numbers aren’t correct twitter archive
- names are not correctly, e.g. "a"
- Some Dogo-Classifications are not extracted correctly, e.g. if doggos is used in text it’s not classified as doggo. blep and snooter are missing completly as possible dog stages classification: twitter archive
- time is displayed as string: twitter archive
- id's are stored as integer'
- column names are not easy to understand in df_image_prediction
- only use tweets, which contain full information, since tweepy-info contains only 885 entries
- Sources have cryptic names

Tidiness Issues:

- doggo attribute should be a dummyvariables once extracted, to anable logisitc regression models: twitter archive check
- tables are not apropiate. One big table for easy analysis - for storing purpose three different tables should be use, one for tweet data one for picture data and one for dogo data

Clean

Before cleaning any data, a copy of each dataframe was made, so the original data remains untouched. For each issue listed in Gather there is a description, code and test part to verify that the changes have worked.

A variety of pandas functions were used to clean the data:

- Pd.todatetime was used to change the datatype of the timestamp.
- Df.query() was used to extract only the original tweets.
- Df.apply was used to apply a function which exchanged cryptic source names with more understanable names and to apply a function to extract the correct “dog stages” out of the text-column
- Df.drop() was used to drop columns which weren’t necessary for the analyses
- Df.rename() was used to gibe columns appropiated names
- .findall()[-1] was used to get the last and correct entry of rating numbers in the tweet text
- pd.get_dummies was used to enable regression moddels using the “dog stages”
- .merge() was used to combine the dataframes

The final clean dataframe was stored as *twitter_archive_master.csv.*