Wrangling Report

The first task of the project “Wrangle and Analyze” was to gather 3 pieces of data. Two of these (later assigned to twitter_archive and twitter_pred) were pretty straightforward to get, because they were provided by Udacity on their webpage in a CSV format. The third one was trickier since I had to scrape the data from a Twitter API. Sadly, the creation of a Twitter Developer Account did not work for me, so I imported the tweet-json.txt file in my Jupyter-Notebook manually and assigned it to “twitter_ext. This file was also provided by Udacity, in case of failure. After that, I assessed the three separated datasets, to come up with quality – and tidiness issues. The first quality issue I identified was, that the twitter_archive obviously contained duplicate posts. An indicator for that was the column “retweeted_status_id”. As a result, I reassigned twitter_archive using only the posts, which had a “retweeted_status_id” = NaN. The next quality issue I faced was tweet_id in twitter_archive, which was of type integer instead of object type. I simply cleaned that with the .astype(“str”) method. Furthermore, the timestamp feature was of type object instead of datetime. Here I used pandas method pd.to_datetime() to change the datatype to datetime. Additionally, I was not satisfied with the overview of the twitter_archive since it contained a lost of useless features. To make that Dataframe more clearly laid out, I dropped some of the features ("in_reply_to_user_id", "in_reply_to_status_id", "source"), I will definitely not use in my analysis later. The next quality issue I faced was, that some dogs, whose names were unknown, were named as “None” instead of the wanted representation of NaN. I solved this problem by replacing every entry “None” with np.nan, to be able to use pandas and numpys packages methods and functions later. Another quality issue, I randomly came up during my assessment phase, with was the entry “O” in the name column of twitter_archive. My interpretation of that entry was that someone probably wanted to type the name “O’Malley”, so I replaced the string “O” with “O’Malley”. In addition, in some cases the ratings in the text column were written in decimal form, which causes trouble in my subsequent analysis. I used regular expressions to extract the affected entries and after that the round function to come up with this issue. These were the problems I came up with in the first CSV, I imported and called “twitter_archive”. I did not detect any quality issues in the twitter_ext dataframe, but I was not satisfied with the duplicate entries in twitter_pred. I identified this redundancy with use of the column “jpg_url”. As a solution, I used the .drop_duplicates and kept the first of the duplicate entries. Another discontentment I came up with in the twitter.pred, was that p1, p3 and p3 were not consistent in capitalization. I easily solved this problem by applying the .str.capitalize() method to all of the affected columns. After cleaning up these quality issues, I noticed two major tidiness issues: First of all, the variables “doggo”, “floofer”, “pupper” and “puppo” in the twitter_archive dataframe. All these actually belong to the the same feature, I decided to “melt” to “dog_stage”. The second issue was that the two tables “twitter_archive” and “twitter_ext” actually belong to the same tweets. As a result, I decided to merge both of it with use of the “twitter_id”. Before I was able to do so, I had to rename the twitter_ext column “id_str” to “tweed_id”, that both tables have the same column name to merge.