Data wrangling on twitter archive dataset collected from @WeRateDogs.

Data wrangling was performed in three steps :

1.Gathering:

This was done on 3 types of dataset.

We were given the ‘twitter_archive_enhanced.csv’ file to start with. The file had 2356 tweet records on 17 columns. We downloaded it manually and loaded it to a dataframe.

Next we programatically downloaded the ‘image_predictions.tsv’ file from the provided url. We used request library for this purpose. Using requests.get() function we sent a request to the server via the url and saved the response to a variable. Then we saved the content of the response in a file called ‘image_predictions.tsv’ in our main folder.

Next we queried the twitter API for data on tweets using the tweepy library. For this purpose we opened an account in twitter to get the Consumer API keys, and the Access Token and Access Token Secret. Then we used the the API constructor from tweepy library to construct an api object. Next we used the api to get the status of each of the tweets using the tweet ids from first dataset. We obtained the json of each tweet and appended each of the json in a text file called ‘tweet_json.txt’. Some of the tweets were deleted so we used a try-catch block to get the deleted tweets in a list. Some of the tweets in the error list was found to be not deleted. So, we ran again to get their json data. Next we extracted the name, stage, rating from the full_text attribute of json. We used the search and findall methods of the re library for this purpose. We also extracted the retweet count, favorite count and number of images from the json file using the proper attributes. We first collected all this information as a dictionary, appended to a list. Next using the DataFrame method of pandas we converted this list of dictionary to a dataframe. We used a try-catch block here too, to obtain the problematic tweet ids in a list. These tweets were basically retweets or replies to another tweet and hence missing the required attributes.

2. Assessing:

The gathered data was then assessed.

The csv file was opened in excel and manually checked for data quality.

Next we went through each of the three files programmatically to find quality and tidiness issues in the datasets. Some of the important methods used were info(), value_counts(), .head().

Some of the quality issues included missing data, inaccurate data, incorrect object types, useless columns, data needing further extraction.

Some of the tidiness issues include melting four columns into one column. And merging, renaming, reordering columns.

3.Cleaning

We addressed each of the issues mentioned in the assessment step of the notebook file. We defined the procedure to solve the issue and then coded, applied and checked for the existence of the issue.

We moved between quality and tidiness issue, where ever deemed necessary. Cleaning involved some methods like melt, drop, merge, concat etc. Also we defined some functions to prevent repetition of code.

In case of some tweets we had to check the ‘text’ column or look up the tweet’s status here https://twitter.com/dog_rates/status/ID to make the necessary changes.

Conclusion:

After the three steps of Data wrangling we obtained a clean dataframe that was saved in a csv file and also in a table in a sqlite database. We then used that dataframe for our analysis.