Wrangle Report

Cody Mollenkopf

08/10/2020

In project number 4, we were asked to wrangle the twitter data (The WeRateDogs Twitter archive, The tweet image predictions, and Each tweet's retweet count and favorite ("like") count. I was able to download the WeRateDogs Twitter archive file manually by clicking the following link provided. Then I used the import and read csv operation to read and then create a new data frame called df_twitter_archive.

The next file that was required to wrangle was the tweet image predictions file. Now I was able to do this a couple of ways. The first way was using the import tsv method and use the tab separation as well to separate the data. The second way was to import requests import os. The next step was to create a folder_name = 'image_predictions' and we were able to request the url for all of the files in the image_predictions tsv file. The status code <Response [200]> informs you of the status of the request, which was a success.

For the third, I decided to use the alternate route due to not being able to get into twitter. I have had some issues, so I have downloaded, copied and paste it into the notebook. But it seems that we make sure to import tweepy which gives us the opportunity to use python which communicates with Twitter platform and use its API. And we are importing json so we are able to encode the data and use it to read and write to a new file. The goal is to query twitter api for each tweet in the Twitter archive and save JSON in a text file. Also, make sure that the all keys and tokens are 'Hidden' as best practice case of privacy. Then get tweets IDs for which to gather addition data via twitters API and use the len method to determine how many tweet id's. Then we query twitter's API for JSON data for each tweet ID in the twitter archive. Then save each tweet's returned JSON as a new line in a text file by writing the text as an outfile and using a for loop to count the tweet_id column.

Next was to move to the assess stage of the project; which began by using info() to recieve the summary of our new data frame(s). This provided the index dtype and column dtypes, non-null values and memory usage. Then I used the sample function to retirve multiple sample data to determine if there was any consistancy with the information in our dataframes. We ran the len function to determine the number of values in an df. Then we looked for any tidyness issues and one of those was to make sure if there was any duplicate column names present which there was. Now before cleaning the data we made sure to copy the data into new dataframes. We listed our 3 tidyness issues and 8+ cleaning issues. Then we deleted in_reply_to_status_id', in_reply_to_user_id', retweeted_status_id', retweeted_status_user_id',retweeted_status_timestamp columns which were not needed. In the clean: quality stage, We found any retweets and removed them from the table, and then merge together all of the dataframes together using the tweet_id as a primary key. I also merged the doggo, floofer, puppo, pupper into one column. I made sure to replace any title in p1, p2, p3 columns with none, wronganimal, wronganimal again for consistency. I think because this is a weRatedogs. The data should keep its integrity and have dog breeds, names, related items. Then we used the replace function again for 'a, 'an', and 'the'. This is a quality issue and those are not correct names that should be seen. So, we replaced it with NoName, NoName, and NoNameTwo. Next, we used the title() function to capitalize the first letter in all words within the following columns: name, p1, p2, and p3. I also dropped the timestamp column and separated those values into their own column, (day, month, year). As I continued, I tended to add more cleaning items to our list. The final quality cleaning items were Remove NaN values from p1, p2, p3 and the expanded_urls column. We were successful in dropping these values which allowed us to run value counts to determine the most common, least common dog breed and imputing values were 0 because they were initially null.