wrangle_report

September 12, 2022

Name: Emenike Goodluck

Project: Udacity data wrangling project

0.1 Reporting: wragle_report

0.1.1 Gathering the data

The data used in this analysis were gathered fron three different sources. First, the tweets from WeRateDogs was downloaded manually from `WeRateDogs archive`. This file is called `twitter-archive-enhanced.csv`. After downloading this file, I uploaded it to the jupyter notebook and loaded it as a pandas dataframe named *t_archive*.

I downloaded the second file `image_predictions.tsv` from the URL provided using `requests` and `os` libraries. After getting the file, I loaded it into a pandas dataframe that I called *image_pred*.

Lastly, the third file was meant to be downloaded from twitter API (`tweepy`). However, I had problems with getting my secret keys and tokens from twitter. So, I used the files provided by Udacity. After getting the `tweet-json.txt` file, I read the file line by line using JSON library to a pandas dataframe that I called *tweet_json*

0.1.2 Accessing the data

After gathering the data, I accessed each data both visually and programatically. I did the visual accessment by loading each dataframe and looking for mis-spelled column names, duplicated columns, structure of the data contained in each column, and missing values.

I did the programatic accessment using the following pandas functions: `df.info()` to see the summary of the files, the datatype of each column, and the number of non-empty elements in each column, `df.isna().sum()` to know the number of missing values in each column, `df.duplicated().sum()` to get the number of duplicated entries, `df.shape` to get the shape of the data, `df.describe()` to get the statistical summary of the data, `df.<column_name>.value_counts()` to get the number of each unique entry in the specified column, and other specialized codes to identify the posts that are retweets.

0.1.3 Cleaning the Data

With the issues identified, each of the dataframe was cleaned based on the issues identified. Before carrying out the cleaning process, I made a copy of each of the files and used the copies for the cleaning process.

I dropped all posts that are retweets and the ones that are replies in the archive data, dropped every column that had much missing data, and changed `timestamp` column from string to datetime.

After cleaning all the issues identified in all the files, I merged all the files to form one single master file. I first merged the JSON data with the archive data using `archive_and_json_df = archive_df.merge(json_df, on = 'tweet_id', how = 'left')`. After merging these files, I checked for missing values again, then I merged this file with the image prediction file by running: `tidy_master_df = archive_and_json_df.merge(image_pred, on = 'tweet_id', how ='left')`.

0.1.4 Saving the data

Lastly, the master dataframe was checked, and the missing values were cleaned. Then, I saved the master dataframe as a csv file by running: `tidy_master_df.to_csv('twitter_archive_master.csv',index=False)`