Wrangle Report

Table of Content

Gathering

Three sources to collect data from: Udacity website, Udacity server, Twitter API

Step #1: Download ***twitter-archive-enhanced.csv*** manually
The aforementioned file from the Udacity website is downloaded to the working directory

Step #2: Download ***image-predictions.tsv*** programmatically using the ***requests*** library
A TSV file from the Udacity servers is downloaded by sending a request to https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-predictions/image-predictions.tsv and saving the response content to a file in the current working directory

Step #3: Create JSON file ***tweet-json.txt*** by querying Twitter API using the ***Tweepy*** library
Using the ***Tweepy*** library, Twitter API is queried to get a JSON object, extract needed data, and store it in a .txt file, that then is read into a Pandas data frame.

Assessing

Step #1: Visual assessment
Explore each table in a spreadsheet and using Pandas
Step #2: Programmatic assessment
- methods used to explore the data include:
- .info() for extracting information regarding the data frame like number of observations, data types, columns, and missing data.
- .describe() for a statistical summary of quantitative variables
- .duplicated() for duplication in dataset

• .value_counts() for duplication of specific column
• using masks and .query() to find specific issues (like ratings less than 10)
• .isnull() to check for null values
• .sort_values() to find outliers and latest/earliest date

- Visual and programmatic assessment notes:

General
• Too many unnecessary tables
• Missing data in all tables

Twitter Archive Table
• doggo, floofer, pupper, puppo column names are not understandable
• Conventionally, all ratings should be greater than 10, some of them are not
• Source column is unreadable with each source inside a link.
• tweet_id as an integer allow for senseless computations that could be prevented
• Convert timestamp to datetime data type
• Dog names are inconsistently capitalized

Image Predictions Table
• prediction dog breed is inconsistent in terms of capitalization
• Column names are not descriptive enough
• tweet_id should be a string

Twitter API Table
• Tweet count and favorite count shouldn't be strings

Cleaning

Before cleaning, all three tables were copied. Then these measures were taken to clean the data:

Step #1: Quality issues

Problem	tweet_id as an integer allow for senseless computations that could be prevented.
Solution	Using .astype(str) to convert tweet_id data type to string (object) in all t_archive_clean and i_pred_clean tables
Problem	Convert timestamp to datetime data type
Solution	pd.to_datetime() is used to fix datatype from object to datetime
Problem	Convert retweet_count and favorite_count columns to integers

Solution	pd.to_numeric() converts the two series to numerical values, then filling NA rows with 0s with .fillna(0) and subsequently .astype('int64') to convert the rows data type to integers
Problem	Column names are not descriptive enough in image-predictions.tsv file
Solution	.rename() function with the columns argument to change all 9 column names in i_pred_clean table
Problem	Conventionally, all ratings should be greater than 10, some of them are not.
Solution	Targeting all rating_numerator less than 10 and adding 10 to the values assuming they are mistyped, except for the two columns with 0s since it's not clear what the rating should be. The zero columns were dropped using their indices in .drop()
Problem	Rating denominators are not all equal to 10
Solution	Using a mask to query all the rating_denominators greater or less than 10, then replacing theses values with 10 using .where() function with the 'other' argument and the mask.
Problem	Source column is unreadable with each source inside a link.
Solution	Extract the source from each link in the source column in t_archive_clean table and reassigning the column with the extracted value
Problem	Dog names are inconsistently capitalized
Solution	names are transformed to lowercase using .str.lower()
Problem	Prediction dog breeds are inconsistent in terms of capitalization
Solution	Also, transformed to lowercase using .str.lower()

Step #2: Tidiness issues

Problem	Too many unnecessary tables
Solution	Merge archive table with twitter api table where the first tweet_id agrees with the latter table, Later, merge the image prediction table with the prior merged table.
Problem	doggo, floofer, pupper, puppo column names are not understandable
Solution	melt all four columns into on called dog_stage with a value column called dog_stage_val using .melt() function

After each cleaning process, a reiteration or reassessment was made in order to make sure the needed alterations were successful. Then new merged table is saved as a CSV file to working directory.