wrangle_report

December 4, 2023

0.1 Reporting: wragle_report

- Create a **300-600 word written report** called "wrangle_report.pdf" or "wrangle_report.html" that briefly describes your wrangling efforts. This is to be framed as an internal document.

0.1.1 Objective:

The primary goal is gathering, assessing, and cleaning the raw data for further analysis, ensuring the quality and tidiness.

0.1.2 Data Sources:

1. `twitter-archive-enhanced.csv` dataset
2. `image-predictions.tsv` from `https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad` url
3. `tweet-json.txt`.

0.1.3 Project Details:

1. Gathering Data
2. Assessing Data
3. Cleaning Data

0.1.4 Gathering Data:

`twitter-archive-enhanced.csv` file: - Download the `twitter-archive-enhanced.csv` and then upload the dataset to the workspace. - Use the pandas read_csv() function to read the file into a dataframe named `twitter_df`.

`image-predictions.tsv` file: - Get the "https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2 predictions/image-predictions.tsv" url. - Import requests library and use the function to get the data from above url. - Use the content function to get the content of `image-predictions.tsv` and then write the content to a same name tsv file. - Use the pandas read_csv() with sep='^' argurment to read the file into a dataframe named `images_df`

`tweet-json.txt`: - This file supposed to be created by the twitter API. But currently I am struggling with the API because of the updating from v1 to v2. For pushing the progress faster, I used the `tweet-json.txt` provided by the project guidline. - Uses the pandas read_json() function to read the `tweet-json.txt` file and put it in the dataframe named `tweets_df`. - Extract the `tweets_df` dataframe to get relevant columns: 'id', 'retweet_count', 'favorite_count'. - Rename the columns for clarity, change the names to 'tweet_id', 'retweet_count', 'favorite_count'.

0.1.5 Assessing Data:

Visual Assessment: - Print three dataframes individually. - Learn about comprehensive meaning of the datasets.

Programmatic Assessment: - Identify mising values, outliers, inconsistency data, duplicated values ... using .describe(), .info(), .duplicated(), .value_counts()

0.1.6 Cleaning Data:

- Made a copy of the original data before cleaning: `twitter_clean`, `images_clean`, `tweets_clean`.
- Used the Define-Code-Test framework for cleaning each issue.

Clean up the missing data issue first: - Remove all the retweets rows to make only original dog ratings for the analysis. - Remove the columns retweeted_status_id, retweeted_status_user_id, retweeted_status_timestamp, in_reply_to_status_id and in_reply_to_user_id because not only is most of the data missing, but those columns are also irrelevant to project analytics

- Create new column named `rate`: rate = rating_numerator divided by rating_denominator.

Secondly is cleaning up the tidiness issue:

- Combine the four dog stages columns into one single column.
- Merge `twitter_clean`, `images_clean`, `tweets_clean` into one dataset named `Merge_df` using inner join on the `tweet_id`.

Lastly is cleaning up the quality issue:

- Fix the erroneous datatypes (timestamp should be a datetime, tweet_id should be a string).
- Remove the html tag in source column.
- Replace all cells containing names without actual names to "None".
- Manually change the name of 776201521193218049 cell from O to O'Malley.
- Change the dog_stage datatype to "category".
- Convert all values in `p1`, `p2`, `p3` to lowercase.

0.1.7 Storing Data:

Save the merged data in a csv file named `twitter_archive_master.csv`

0.1.8 Conclusion:

Transformed raw and disparate data into clen data for in-depth analysis.

In [ ]: