wrangle_report

June 25, 2022

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.

1 Data Wrangling Project

1.0.1 D.K. Oluwadare

1.1 Introduction

This project demonstrates the data wrangling process for the tweet archive of Twitter user @dog_rates, @dog_rates is a Twitter account that rates people's dogs with a humorous comment about the dog. In this analysis I demonstrate the data wrangling techniques that were used to gather, assess and clean the dog twitter archive.

1.2 Project Overview

1.2.1 Gather data

The following files were gathered for the analysis:

- The WeRateDogs (@dog_rates) Twitter archive - This file (archive.csv) was downloaded using Twitter's API and consists of basic tweet data for 2300+ tweets from WeRateDogs.
- The tweet image predictions - i.e., what breed of dog (or other object, animal, etc.) is present in each tweet according to a neural network. This file (image_predictions.tsv) was downloaded programmatically from Udacity.
- Each tweet's retweet_count and favorite_count -This file (tweet_json) contains JSON data for each tweet indicating the retweet and favorite counts.

1.2.2 Assess data

The three files obtained in the gathering phase were loaded into individual Pandas data frames for assessment. Each of the data frames were evaluated visually and programmatically.

Virtual assessment The involves mere looking at the data virtually to have an insight of what to work with in the data. Virtual assessment doesnt give detailed understanding of a data. I used some basic pandas methods like (.head(), .shape) to know the columns and rows to expect when doing my analysis

Programatic assessment Programatic assessment helped me idenified various quality and tidiness issues. The quality and tidiness issues that would be cleaned using programmatic techniques includes but not limited to:

```
Dropping unnecessary columns from the tables
Removing rows that consisted of null retweets
Removal of rows with duplicate information
Deleted rows that did not have any dog predictions at all
Combining all three data frames into a single data frame
```

1.2.3 Clean data

Quality

1. The column 'id' should be changed to tweet_id in the *newapi* Table
2. Some uppercase and lowercase letters identified in columns 'p1', 'p2', and 'p3' in the *image_prediction* Table
3. 'text' column has unnecesary HTML code in the *twitter_archive* Table
4. Missing values in columns: in_reply_to_status_id, in_reply_to_user_id, retweeted_status_id, retweeted_status_user_id, retweeted_status_timestamp, and expanded_urls in the *twitter_archive* Table
5. 'timestamp' should be a datetime64 dtype as well in the *Twitter_archive* Table
6. Remove the 'source' column in the *twitter_archive* Table
7. Drops rows with duplicates in the jpg_url column
8. Change timestamp format from "2017-07-26 15:59:51+00:00" to "2017, 2016, 2015...."

Tidiness

1. Merged all three tables into one.
2. Dog tests are spread in three columns.

1.2.4 Analyzing, Visualization and Storing

INSIGHT 1 : DOG CATEGORIES WITH FAVOURITE COUNTS INSIGHT 2 : DOG WITH THE MOST POPULAR NAME INSIGHT 3 : TOP 20 DOG NAMES WITH RETWEETS RATE INSIGHT 4 : DOG BREED WITH HIGHEST MEAN RATE NUMERATOR, AND HIGHEST VALUE COUNTS

1.2.5 Limitations

- Handling error from writing (w) a json_file wih the twitter API tokens.
- The Dog tests spread in three columns had so many 'none' rows, afetr combining it into dogtypes, i had to remove the rows and worked with the ones visible to analyse

1.2.6 Reference

https://stackabuse.com/reading-and-writing-json-to-a-file-in-python/
https://chrisalbon.com/code/python/data_wrangling/pandas_apply_operations_to_dataframes/
https://stackabuse.com/reading-and-writing-json-files-in-python-with-pandas/
https://www.youtube.com/watch?v=SLM5R59-b6g https://www.youtube.com/watch?v=MbKrSmoMads
https://www.youtube.com/watch?v=6GUZXDef2U0 https://seaborn.pydata.org/tutorial/function_overview.h

```
In [ ]:
```