Report: Data Wrangling

Introduction

The dataset I wrangled (and analyzed and visualized) is the tweet archive of Twitter user @dog_rates, also known as WeRateDogs. WeRateDogs is a Twitter account that rates people's dogs with a humorous comment about the dog. These ratings almost always have a denominator of 10. WeRateDogs has over 4 million followers and has received international media coverage.

Because real-world data rarely comes clean, I gathered data from a variety of sources and in a variety of formats, assessed its quality and tidiness, then cleaned it using Python and its libraries. My goal was to wrangle WeRateDogs Twitter data to create interesting and trustworthy analyses and visualizations.

I worked on this project outside of the Udacity classroom using Jupyter Notebook installed on my computer. Before starting, I installed the following packages (libraries) via conda:

- Pandas
- NumPy
- Requests
- Tweepy
- Json
- os

Data Gathering

For this project, I worked on three datasets. The methods required to gather each data were different.

To get the first dataset, 'WeRateDogs Twitter archive' I downloaded a .csv format file manually by clicking on a provided link. After downloading, I uploaded it and read the data into a pandas DataFrame.

The second dataset, 'tweet image predictions' is hosted on Udacity's servers. I downloaded it programmatically using Python's Requests library. I used os library to create a folder for the file, used requests to create a request for the file using the provided URL. After getting a successful response, I accessed the content and wrote to a file using the Requests .content method and used a basic file I/O to save this file to the computer. Finally, I read the file into a dataframe using pandas' "read_csv".

For the third dataset, 'Additional data from the Twitter API', I manually downloaded the resulting data from twitter_api.py, provided by Udacity. Then read the 'tweet_json.txt' file line by line into a pandas DataFrame with (at minimum) tweet ID, retweet count, and favorite count.

Assessing Data

After gathering all three pieces of data, I assessed them visually and programmatically for quality and tidiness issues.

a) Visual assessment: each piece of gathered data was displayed in the Jupyter Notebook for visual assessment purposes. I additionally assessed data using Excel, an external application. 8 issues I identified were documented under 'Quality Issues'
b) Programmatic assessment: pandas' functions and methods were used to assess the data and identified issues documented under 'Tidiness issues'.

Cleaning Data

Before I performed the cleaning, I made a copy of the original data. Then I used the define-code-test framework to clean all of the issues I documented while assessing the data.

- To meet specifications, I dropped non-null data from 'retweeted_status_id','retweeted_status_user_id' and 'retweeted_status_timestamp' columns because project needed only original ratings (no retweets) that have images.
- changed the timestamp column to datetime datatype
- removed lowercase erroneous dog names in name column (a, not, quite, incredibly, old, infuriating, etc.)
- filtered to get only rows with consistent rating_denominator
- extracted the text that was enclosed with html tags in the source column.
- dropped redundant columns I considered irrelevant for my analysis
- removed the false dogs predictions from the twitter image predictions dataset
- joined the dog type variables into one column
- merged the datasets into one master dataframe

Storing Data

I stored the cleaned master DataFrame in a CSV file named 'twitter_archive_master.csv'.