Udacity Data_Analyst Program

Wrangle_Act Report

Kolawole Ayoko

Introduction

In order to record the project's data wrangling efforts, this wrangle report is a component of the Wrangle and Analyze Data project. The WeRateDogs account on Twitter, @dog rates, provided the dataset for this project. A Twitter account called WeRateDogs awards stars to users' pets along with amusing comments about them. The three processes of data wrangling—data collection, assessment, and cleaning—are documented in this wrangle report.

Data Wrangling Process

- Data Gathering from various sources
- Assessing the data for quality and tidiness (visual and programmatic)
- Data cleaning based on assessment made

Data Gathering

For this wrangling process, I needed to acquire information for my assignment from several sources and in a variety of file formats.

1. The archive of tweets from WeRateDogs. The project provided the file, which is available for direct download from the Udacity website.
2. The predicted tweet images. The file is stored on servers owned by Udacity. I automate the download of this file using Python's Requests package.
3. I Retrieved from another file the retweet and favorite counts that are absent from the Twitter archive. Since I don't have a Twitter account, I decided to obtain the tweet JSON file programmatically using the Requests package.

Assessing The Data

After collecting the data, I evaluated it programmatically and visually to find any problems with data quality or organization. Tidiness is a function of data structure, whereas quality relates to content.

requirements for orderly data: each variable should be represented by a column, each observation by a row, and each sort of observational unit by a table. The files weren't that big, so I could load them on google sheets and skim through the data to look for any glaring flaws. To view particular subsets and summaries of the data, I also utilized code in Jupyter Notebook, such as the pandas .info(), .head(), .sample(), .value counts(), .duplicated(), and describe methods.

Quality Issues Documented

- Merging data prediction column to have one single column for dog_breed and prediction confidence ratings

- Only original tweets that have pictures should be retained, retweets will be removed
- Timestamp column is string, it should be converted to datetime object
- Source information provided is difficult to read and should be cleaned
- Some of the dog_breeds are not actually dogs and they should be capitalized
- Some columns will be dropped as they have a lot of missing values
- Incorrect Entries in rating numerators and denominators
- Correct all the wrong data types in the dataset, including changing source and dog_stages into category data type for future analysis
- Removing missing values in expanded_urls column by using `.dropna`

Tidiness Issues Documented

- The dog stages Column for df1 twitter_archive_enhanced.csv can be placed into one column
- Information about the tweets are spread across three different datasets. Therefore, these three datasets were merged as one.

Data Cleaning based on assessment made

While assessing, I cleaned each of the issues listed above. Even though the entire dataset has a lot of issues, fixing them all would take a lot of time. I therefore concentrated only on those relevant to my analysis. The three steps of the programmatic data cleaning process are to define, code, and test. Before cleaning, it's also crucial to make copies of the original data. Because certain issues will vanish when we clear them one at a time, it's crucial to address them in a logical order. For instance, once we fixed the problem that many photos in the image prediction are not dogs, several of the odd rating numerators and denominators earlier observed vanished.

Some of the abnormal ratings automatically disappeared when I eliminated posts that weren't about dogs. Retweets and replies were removed because the project only needs original tweets. The majority of cleanings were completed using programmatic tools, I was able to convert the timestamp column to a date time object. This conversion gave a deeper insight to the frequency of tweets based on months, days and year. Some of the columns in the original dataframes were dropped due to an abnormally large number of missing values. The confidence ratings for the image prediction dataframe was also optimized to have only one important rating based on the confidence values provided.

After fixing all the issues, I reassessed the dataset and iterated when necessary. Then I stored the clean data in a csv file.