Project: Wrangle and Analyze Data

WeRateDogs Twitter Archive Analysis

Introduction

Data wrangling is an essential procedure that must be done before data can be used. It is composed of three main processes:

- Data gathering, in which the data is collected from different sources, such as CSV files, databases, downloading, API queries, and web scrapping. It is a best practice to keep all these tasks done programmatically, in order to allow scalability and reproducibility.

- Data assessing, in which we verify the data quality and tidiness. The data quality is assessed by considering if it fits the purpose we want and considering some issues, such as missing, invalid, inaccurate, and inconsistent data. The data tidiness is assessed by verifying the structure of the data, this structure must facilitate the analysis. Data tidiness principles indicate that each variable forms a column, each observation forms a row and each type of observational unit forms a table.

- Data cleaning, in which we want to improve the quality and tidiness of data. This is done by considering the issues perceived during the data assessing procedure. For each of these issues, it is defined how to do the cleaning, it is implemented a code and it is tested if the code was effective.

After the data wrangling, it is useful to store clean data in order to continue with the data analysis and visualization. This allows us to easily have the same piece of clean data to be analyzed and, if during the analysis procedure some change needs to be made, it is possible to come back to this first piece of clean data.

Data Gathering

In the current project, it was aimed to analyze the WeRateDogs Twitter archive. In order to do so, data was gathered from three different sources. A WeRateDogs Twitter enhanced archive in CSV format was download manually from Udacity's servers, an image prediction file was downloaded programmatically from Udacity's servers and the JSON data for all tweet IDs in the enhanced archive were gathered by querying the Twitter API making use of the Tweepy library in Python. The JSON data was stored in a file called tweet_json.txt, being each tweet information stored in one line. This JSON data was then read in Python and information about tweet ID, retweet count and favorite count was gathered.

Data Assessing

The data was assessed visually and programmatically, verifying for the completeness, validity, accuracy and consistency of the data. The visual assessment was done by displaying the tables and considering the number of rows and columns, the name of columns, the data type for each cell and what each row represents. The programmatic assessment was done by using Pandas functions and methods, verifying the issues mentioned earlier.

The following quality issues were encountered during the assessment procedure:

* We should not include tweets that are retweets.

* `expanded_url` has missing entries.

* `timestamp` should be a datetime type.

* `source` column has confusing data.

* The dog `name` is none for many entries. Change data type.

* The dog `name` has names identified wrongly.

* The dog `name` identified as a or by.

* Display full content of text column.

* Remove url from text column.

* `rating_denominator` has really high maximum values (should it be up to 10?).

* `rating_numerator` is not accurate for all entries, i.e. those that have a ..

* Dog stages are defined as None instead of NaN.

* There missing values (there are only 2075 ID entries, less than the ID entries number for twitter_archive)

* There are images that do not identify dogs.

* Multiple types of dog bread spelling.

* There missing values (there are only 2331 ID entries).

* Favorite counts should be integer.

* Tweet counts should be integer.

The following tidiness issues were encountered during the assessment procedure:

* `expanded_url` has multiple urls in one cell.

* Dog stage should be just one column.

* Need just one column for the true prediction.

* Only one table is necessary and it could have the following columns: 'tweet_id', 'timestamp', 'source, 'text', 'url', 'rating_numerator', 'rating_denominator', 'name', 'dog_stage', 'dog_prediction', 'retweet_count', 'favorite_count'.

Data Cleaning

After assessing the data, it is necessary to clean it. The first step performed was to make a copy of the data. Next, it was tacked the tidiness issues. The tidiness issues were considered first in order to have a better organization of DataFrames and do not have unnecessary data when merging DataFrames. This step was followed by quality issues, starting with the missing data problems. When dealing with the missing data, the DataFrames were merged in order to identify the tweet IDs that were common among them. For both tidiness and quality issues, it was considered a procedure in which it was defined how to clean the data, the cleaning code and a code to test whether the cleaning was successful.

Data Storing

Having a high quality and tidy master DataFrame, this information was stored in a final CSV file. This file can be used as the starting point of future analysis.