Wrangle Efforts Report

Introduction

The purpose of the project was to put in practice what has been learnt in the data wrangling module from Udacity Data Analysis Nanodegree program.

About the data

The dataset that will be wrangling is the tweet archive of Twitter user @dog_rates, also known as WeRateDogs. A Twitter account that rates people's dogs with a humorous comment about the dog. These ratings almost always have a denominator of 10 but numerators almost always greater than 10.

Gathering the Data

All the data used in the project had to be gathered from different sources and with different formats.

1. Enhanced Twitter Archive

The WeRateDogs Twitter archive contains basic tweet data for all their tweets that contains ratings.

It was provided in a CSV file by Udacity, it was downloaded manually and uploaded to the folder where the analysis took place. Can be found in this same directory under the name "Data/twitter_archive_enhanced.csv".

2. Retweet and favourites count via the Twitter API

The tweet IDs in the WeRateDogs Twitter archive was used to query the Twitter API for each tweet's JSON data using Python's Tweepy library and store each tweet's entire set of JSON data in a file called tweet_json.txt file.

The gathering of this dataset was the most challenging one. Although, tweetpy was very straightforward to use I had issues on the first tries because once the rate limit was reached all tweet request started failing. By adding the variables *wait_on_rate_limit* and *wait_on_rate_limit_notify* to the API I was able to run the request one single time only.

Afterwards, the file was turned into a data frame by using a loop to read line by line.

3. Image Predictions File

A table full of image predictions alongside each tweet ID and image URL.

This file (image_predictions.tsv) is hosted on Udacity's servers and was downloaded programmatically using the Requests library.

Assessing the Data

After gathering all the data and storing it into three different pandas dataframes the assessment of the data took place as following:

1. Visually

All data frames were sampled on the Jupyter Notebook to check how the structure and the contents look. A first assessment on some missing data and wrong formatting of the data and the structure took place.

2. Programmatically

Additional to the visual assessment, several functions were used to evaluate the basic information on the data: data types, number of records, number of missing values, duplicated values, etc.

By using these two methods and following the key points given in the instructions of the project, several Data issues were encountered and separated into quality and tidiness issues.

Cleaning data

At first the datasets were manipulated one by one, and finally the data was merged into one single dataset with the columns relevant to the analysis.

The cleaning was the most consuming part of the project, and it certainly helped having the issues already defined and ordered.

Once the data was cleaned it was stored into a csv file. This step was crucial for the analysis, because it makes both stages independent, and the clean data more accessible.

Conclusion

Having done in the past most of the data wrangling using tools like Excel or SQL, I learned that Python gives a greater amount of advantages on the data gathering, assessment and cleaning, that impact directly on the quality of the analysis.

The greater advantages on my opinion:

- Functions specially developed to provide information about the data, no extra calculation or coding needed. Eg. Describe, dtypes, shape.
- The range of different types of data gathering, no need to combine different tools, since all can be done from the same Jupyter notebook.
- Combining multiple datasets from different sources is easier, no need to convert any data, or copy paste manually with the risk of losing data.
- It’s easy to document along with the code using the markdown cells.