Data Wrangling

Introduction

Real-world data rarely comes clean. Using Python and its libraries, you will gather data from a variety of sources and in a variety of formats, assess its quality and tidiness, then clean it. This is called data wrangling. You will document your wrangling efforts in a Jupyter Notebook, plus showcase them through analyses and visualizations using Python (and its libraries) and/or SQL. The dataset that you will be wrangling (and analyzing and visualizing) 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. The numerators, though? Almost always greater than 10. 11/10, 12/10, 13/10, etc. Why? Because "they're good dogs Brent." WeRateDogs has over 4 million followers and has received international media coverage.

WeRateDogs downloaded their Twitter archive and sent it to Udacity via email exclusively for you to use in this project. This archive contains basic tweet data (tweet ID, timestamp, text, etc.) for all 5000+ of their tweets as they stood on August 1, 2017. More on this soon.

Project Details

Data wrangling, which consists of:

- Gathering data
- Assessing data
- Cleaning data
- Storing, analyzing, and visualizing your wrangled data

Data Gathering

In this project we are using three dataset

1) The WeRateDogs Twitter data file which holds 2356 entries with 17 columns.I manually download this file from the link twitter_archive_enhanced.csv

2) 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) is hosted on Udacity's servers and should be downloaded programmatically using the Requests library and the following URL: image_predictions.tsv

3) Each tweet's retweet count and favorite ("like") count at minimum, and any additional data you find interesting. Using the tweet IDs in the WeRateDogs Twitter archive, 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. Each tweet's JSON data should be written to its own line. Then read this .txt file line by line into a pandas DataFrame with (at minimum) tweet ID, retweet count, and favorite count.

Data Assessing

Once after gathered the data frames. Now its time assess the data and it is the place where we could find the issues in the dataset. This can be done either visually or programmatically.

- Visually, I used two tools. One was by printing the three entire data frame separate in Jupyter Notebook and two by checking the csv files in Excel
- Programmatically, by using different methods (e.g. info, value_counts, sample, duplicated, groupby, etc)

Then I separated the issues encountered in quality issues and tidiness issues. Key points to keep in mind for this process was that original ratings with images were wanted.

Cleaning data

This part of the data wrangling was divided in three parts: Define, code and test the code. These three steps were on each of the issues described in the assess section. Other interesting cleaning code was to melt the dog stages in one column instead of four columns as original presented in twitter archive.

First and very helpful step was to create a copy of the three original dataframes. I wrote the codes to manipulate the copies.If there was an error, I could create a new copy from the original. Whenever I made a mistake, I could create another copy of the dataframes and continue working on the cleaning part.

There were a couple of cleaning steps that were very challenging. One of them was in the image prediction table. I had to create a ‘nested if’ inside a function in order to capture the first true prediction of the type of dog. The original table had three predictions and confidence levels.I filtered this into one column for dog type and one column for confidence level.

Conclusion

Data wrangling is a core skill that whoever handles data should be familiar with. I have used Python programming language and some of its packages. There are several advantages of this tool (as compared to e.g. Excel) that is used by many data scientists (including the guys at Facebook).

- For gathering data there are several packages that help scraping data off the web, that help using APIs to collect data (Tweepy for Twitter) or to communicate with SQL databases.
- It is strong in dealing with big data (much better than Excel)

- It can deal with a large variety of data (unstructured data like JSON (Tweets) or also structured data from ERP/SQL databases.