Data Wrangling

By Shilin Li

July 10th, 2018

Introduction

Real-world data rarely comes clean. Using Python and its libraries, we can 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.

The dataset that I am 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.

This part is extremely challenging as it requires both cautiously actions and advanced techniques, I've learned substantial new stuff adding to my reserve. The instructor mentioned that we can't learn everything during the training, moreover, new libraries and programming tools are emerging, thus good searching skills is key important for being a qualified data analyst.

Gathering

Data sources came from three ways shown as below:

1. The WeRateDogs Twitter archive is download manually by clicking the following link: https://d17h27t6h515a5.cloudfront.net/topher/2017/August/59a4e958_twitter-archive-enhanced/twitter-archive-enhanced.csv.

2. The tweet image predictions, i.e., what breed of dog (or another 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: https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-predictions/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.

During the gathering, we've been learning 3 useful techniques to import data, namely, loading from local, downloading using URL and API. Those 3 techniques are cover all the ways we can access to data, and thus it could be applied to any other data analysis projects. Nowadays it is inevitable to access most of our primary data through internet, and giving the advantages of API, such as automation, efficiency, personalization etc., it becomes the core of data gathering.

Accessing

Access is the second step, here we access them visually and programmatically, then recording any quality and tidiness issues found.

Data quality dimensions can help guide thought process while assessing and also cleaning. The four main data quality dimensions are: completeness, validity, accuracy and consistency.

Tidiness objective is to find issues with structure that prevent easy analysis.

Cleaning

This is the third step of data wrangling, according to the points denoted from accessing, quality and tidiness issues are fixed. Basically, corrections are based on the two approaches as following:

Visually: scrolling through the data in your preferred software application (Google Sheets, Excel, a text editor, etc.).

Programmatically: using code to view specific portions and summaries of the data (pandas' head, tail, and info methods, for example).

Summary

This project is second most challenging work just inferior to "R exploratory data analysis" in Chapter 2. The biggest harvest in current project is getting familiar with

request library and API, which are core for accessing of any kinds of data analysis job.