

Lab Assignment 5: Web Scraping

DS 6001: Practice and Application of Data Science

Instructions

Please answer the following questions as completely as possible using text, code, and the results of code as needed. Format your answers in a Jupyter notebook. To receive full credit, make sure you address every part of the problem, and make sure your document is formatted in a clean and professional way.

For the following problems, you will be scraping <http://books.toscrape.com/>. This website is a fake book retailer, designed to mimic the design of many retail websites. It exists solely to help students practice web-scraping, so there aren't going to be any ethical concerns with this particular exercise, and there shouldn't be any issues with rate limits or other gates that could prevent web-scraping. Take a moment and look at this website, so that you know what you will be working with.

Your goal is to generate a dataframe with four columns: one for the title, one for the price, one for the star-rating, and one for the book cover JPEG's URL. The dataframe will also 1000 rows, one for each of the 1000 books listed on the 50 pages of this website.

Problem 0

Import the following libraries:

```
In [1]: import numpy as np
import pandas as pd
import requests
from bs4 import BeautifulSoup
import sys
sys.tracebacklimit = 0 #turn off the error tracebacks
```

Problem 1

Pull the HTML code from <http://books.toscrape.com/>. Make sure you provide a user agent string. Then parse this HTML code and save the parsed code as a separate Python variable. [3 points]

Problem 2

Extract all 20 of the book titles and save them in a list. [2 points]

Problem 3

Extract the price of each of the 20 books and save these prices in a list. (The prices are listed in British pounds, and include the £ symbol. Remove the £ symbols: if you've saved the prices in a list named `prices`, then the following code should work: `prices = [s.replace('Â£', '') for s in prices]`.) [2 points]

Problem 4

Extract the star level ratings for the 20 books. [Hint: for tags such as `<p class="star-rating One">` in which the class has a space, the class is actually a list in which the first item in the list is `"star-rating"` and the second item in the list is `"One"`. It's possible to search on either item in this list.] [3 points]

Problem 5

Extract the URLs for the JPEG thumbnail images that show the covers of the 20 books. (Maybe we want to mine the images to build models that predict the star level, literally judging books by their covers.) [2 points]

Problem 6

Create a dataframe with one row for each of the 20 books, and the book titles, prices, star ratings, and cover JPEG URLs as the four columns. [2 points]

Problem 7

Create a function that takes the URL of the webpage to scrape as an input, applies the code you wrote for questions 1 through 6, and generates the dataframe from question 6 as the output. [3 points]

Problem 8

Notice that there are many pages to <http://books.toscrape.com/>. When you click on "Next" in the bottom-right corner of the screen, it takes you to <http://books.toscrape.com/catalogue/page-2.html>. The front page is the same as <http://books.toscrape.com/catalogue/page-1.html>, and there are 50 total pages.

Write a loop that uses the function you wrote in question 7 to scrape each of the 50 pages, and append each of these data frames together. If you write this loop correctly, your dataframe will have 1000 rows (20 books on each of the 50 pages).

Some hints:

- Typing `new_df = pd.DataFrame()` with nothing in the parentheses will create an empty data frame on which new data can be appended.

- There are many loops you can use, but the most straightforward one is a for-values loop that counts from 1 to 50. In Python, you can initialize such a loop with `for i in range(1, 51):`, and indenting every line below it that belongs inside the loop. Inside the loop, the letter `i` is now a stand-in for the number currently being considered.
- You will need to figure out how to replace the number in URLs like <http://books.toscrape.com/catalogue/page-2.html> with the number currently under consideration in the loop. You might need the `str()` function, which turns numeric values into strings.

[3 points]