Data Collection, Integration and Preprocessing - Notes

Nils Rechberger

2026-02-09

Week 01: Introduction and Scraping

Big Picture of CIP

Typical data science projects can be grouped into sub-tasks that require different skills.

- Data Collection
- Data Preprocessing/Cleaning
- Analysis and Visualization
- Communication

HTML Introduction

An 'HTML-Element' always has a start- and end-tag (e.g. paragraph tags `<p>` and `</p>`)

```html
<head>
</head>
<body>
    <h2>Hello World!</h2>
        <p>
        Click
        <a href="https://google.com">
            HERE
        </a>
        for a google search.
        </p>
</body>
```

Tip

Use click-selector in the browser inspector to jump to selected html-tag

ID and Class Attributes

The id attribute specifies a unique id for an HTML element. It is used to point to a specific style declaration in a style sheet. A class name can be used by multiple HTML elements. Classes are used to apply a specific style declaration to multiple HTML elements.

Scraping Static Websites

Web scraping, web harvesting, or web data extraction is data scraping used for extracting data from websites. It is a form of copying, in which specific data is gathered and copied from the web, typically into a central local database or spreadsheet, for later retrieval or analysis.

BeautifulSoup

BeautifulSoup is a Python library which helps you searching and extracting data from HTML files.

> Note: Check out the documentation: https://www.crummy.com/software/BeautifulSoup/bs4/doc/

Beautiful soup allows for selecting elements using #id, .class and custom attributes (called css-selectors)

`.css` Selection

```
soup.find_all("p", class_="strikeout")
soup.find_all("p", class_="body strikeout")
soup.select("p.strikeout.body")
```

`#id` Selection

```
soup.find_all("p", id="example")
```

Custom Attributes

```
soup.find_all("p", attrs={"name": "myName"})
```

Scraping Dynamic Websites

Static websites provide the full content at once. Dynamic websites only provide parts of the content, that is required for first decisions. On request, it provides customized details. Requests are sequentially sent to the web server. Scraping dynamic websites comprises two key challenges:

1. Detail/product pages are not always structured in the same (or expected) way and may change in time.
2. When we get website contents, it is possible that NO or just SOME are returned in the .

How can you find out if a website is dynamic?

Combinations of client- and server-side scripting:

- Client-side scripting involves code that is executed by the viewer's browser using scripts like JavaScript. These scripts are responsible for rendering changes to a website as response to actions taken, such as mouse clicks, hovering or keyboard inputs.
- Server-side scripting, on the other hand, refers to code that is executed by the server before sending the content to the viewer's browser. This affects a website when it is loaded or visited, like with login pages, submission forms, and shopping carts.

In combination, modern websites can adapt user's view while reducing server's load time

Selenium

Selenium is an open-source package for automating web browsers. It provides a single interface that lets create test scripts.

Web-Scraping: Legal Aspects

For web scraping, in Switzerland we follow two restrictions:

1. New Federal Act on Data Protection nFADP (in effect since September 1st 2023).
2. The international convention to scrape only pages allowed by the owner in the robots.txt. A widely accepted convention for robots.txt is given by Google.

⚠ Warning

Never scrape personalized data.