Webscraping mit Python

Tutorial

im Rahmen des Studiengangs
Big Data und Business Analytics
der FOM Hochschule für Ökonomie und Management

vorgelegt von
Stefan Schött (Matrikel-Nr. 598645)

Frankfurt am Main, den 28. August 2022

Inhaltsverzeichnis

1 Einleitung

Dieses Tutorial behandelt das Thema Web Scraping mit Python. Web Scraping ermöglicht die Extraktion einzelner Daten aus den komplexen Strukturen von Webinhalten. Auf diese Weise lassen sich Webseiten, auf denen Daten in einer semi- oder unstrukturierten Form vorliegen, als Datenquelle anbinden.

1.1 Setup

Das Tutorial orientiert sich am Python Version 3 Standard. Die Version dient als Grundlage für den Nachvollzug der nachfolgend beschriebenen Schritte. Für das Web Scraping wird das Python Package BeautifulSoup Version 4 (kurz: BS4) verwendet. HMTL-Inhalte werden mit dem Python Paket `requests` aus dem Internet geladen.

1.2 Aufbau

Neben dieser Einleitung gliedert sich das Tutorial in die folgenden Abschnitte.

Kapitel 2 beschreibt die Installation von BS4. BS4 ist, zusammen mit Python 3, die grundlegende Technik für dieses Tutorial.

Kapitel 3 behandelt drei Schritte, die für ein erfolgreiches Web Scraping nötig sind. Den Bezug von HTML-Daten über das Python Paket `requests`, das Einparsen und letzendlich die Selektion der Daten mit dem Python Paket `bs4`.

2 Installation

Dieser Abschnitt beschreibt die notwendigen Schritte zur Installation von BS4 und Requests. Eine vorhandene Installation von Python 3 wird vorausgesetzt.

2.1 BeautifulSoup (BS4)

Die Installation von BS4 kann gem. Dokumentation[bs422a] bei den Linux Derivaten Debian oder Ubuntu über den integrierten System Paket Manager apt-get durchgeführt werden.

```
$ apt-get install python3-bs4
```

Ist eine Installation über einen System Paket Manager nicht möglich, gibt es die Möglichkeit, BS4 über die Paket Manager Easy Install und PIP zu installieren.

```
$ easy_install beautifulsoup4
```

```
$ pip install beautifulsoup4
```

Bei älteren Versionen von Python, kann es zu Problemen bei der Installation kommen. Es ist darauf zu achten, dass die passenden Paket Manager für Version 3 verwendet werden. Gegebenenfalls muss die Installation alternativ mit den Kommandos `easy_install3` oder `pip3` durchgeführt werden.

2.2 Requests

Die Installation von `requests` erfolgt ebenfalls über den Paket Manager `pip`[pyp22].

```
$ pip install requests
```

In der offiziellen Dokumentation des Pakets wird zusätzlich eine Variante über den Source Code bschrieben, sollte es Probleme geben[req22].

3 Anwendung

Dieses Kapitel beschreibt die einzelnen Schritte, die nötig sind, um ein einfaches Web Scraping unmittelbar auf eine Webseite anzuwenden. Es untergliedert sich in Abschnitte für den Request[req22], das Parsen[bs422a] und die Suche bzw. Selektion von Daten[bs422a]. Als Quellen dienen die jeweiligen Dokumentationen zu den Paketen.

3.1 Abfrage und Bezug von HTML Inhalten

In diesem Abschnitt wird der erste Schritt behandelt, um Web Scraping erfolgreich durchführen zu können. Der Bezug von HTML-Inhalten von einer Webseite. Hierfür wird das Python Paket `requests` verwendet. Die Verwendung des Paketes ist nach einem entsprechenden Import im Python Code möglich.

```
import requests
```

Um eine Webseite abzufragen wird die `get` Funktion von `requests` verwendet. Dabei wird der Funktion die URL (der Link der Webseite) mitgegeben. Auf den Rückgabewert `get` wird die Funktion `content` angewendet, bevor der das Ergebnis in einer Variable gespeichert wird.

```
html = requests.get(request_url).content
```

`content` bewirkt eine vollständige und lesbare Rückgabe der gesamten Webseiten Struktur[req22].

3.2 Parsen von HTML

Dieser Abschnitt beschreibt einen wesentlichen Schritt, der durchgeführt werden muss, bevor das eigentliche Scrapen von Inhalten beginnen kann. Das Einparsen bzw. das Parsen von HTML-Inhalt.

Damit BS4 verwendet werden kann, muss ein Import im Python Code erfolgen. Das Paket lässt sich wie folgt importieren

```
from bs4 import BeautifulSoup
```

3.2.1 Einparsen einer Webseite

Der Parser stellt eine externe Abhängigkeit von BS4 dar. Das Paket bringt keinen eigenen Parser mit. BS4 unterstützt den integrierten HTML Parser von Python. Daneben können weitere Parser als Alternative eingebunden werden, um zum Beispiel die Performance beim Parsen oder den Umfang an Funktionen zu verbessern. Alternativen sind die Parser lxml und html5lib. In diesem Tutorial wird der Standard HTML-Parser aus der Grundinstallation von Python verwendet.

Zum Einparsen des HTML-Inhalts einer Webseite muss dieser unter Nennung des Parsers an den BS4 Konstruktur übergeben werden. Dies kann in zwei Formen erfolgen. Entweder wird der Inhalt direkt als String übergeben oder als Dokument über das Dateihandling von Python.

Übergabe als String:

```
html = "<html>Inhalt der Webseite</html>"
soup = BeautifulSoup(html, 'html.parser')
```

Übergabe aus einer Datei:

```
with open("main.html") as file_to_parse:
  soup = BeautifulSoup(file_to_parse, 'html.parser')
```

Die Variable soup aus den beiden Codeausschnitten vorher, enthält den geparsten HTML Inhalt. Der Inhalt ist nun in einer kompatiblen Form gespeichert, mit der BS4 Funktionen zur Selektion von Daten darauf angewendet werden können.

3.2.2 Zwischenstand Request und Parsen

Anhand von Fall 1 (String) wird der HTML Inhalt durch `response.get()` geladen und in einer Variable `html` gespeichert. Der Python Code befindet sich nun im folgenden Zustand.

```
import requests
from bs4 import BeautifulSoup

html = requests.get("https://www.tagesschau.de/").content
soup = BeautifulSoup(html, 'html.parser')
```

Der Code importiert die benötigten Pakete, fragt die Webseite der Tagesschau ab und speichert nach dem Parsen den vollständigen HTML Inhalt für die weitere Verwendung in einer Variablen ab. Im nächsten Schritt können darauf die eigentlichen Scraping Funktionen von `bs4` angewendet werden.

3.3 Suchen und Selektieren

Die beiden vorbereitenden Schritte wurden zuvor durchgeführt. In der Variable `soup` befindet sich nun der gesamte HTML Inhalt einer Webseite (Im Beispiel tagesschau.de).

Mithilfe von `bs4` kann nach verschiedenen Typen von HTML Objekten unter Anwendung der meisten CSS Selektoren gesucht werden. Eine umfassende Beschreibung zu HTML[moz22] ist über folgende Link zu finden.

`https://developer.mozilla.org/de/docs/Learn/Getting_started_with_the_web/HTML_basics`

Eine ausführliche Anleitung zu CSS Selektoren ist auf w3schools (`https://www.w3schools.com/cssref/css_selectors.asp`) abrufbar[css22].

3.3.1 Die Funktionen find_all und find

Die Funktion `find_all` ist eine der wesentlichen Funktionen aus `bs4` für die Anwendung von Web Scraping. Ihr wird ein CSS Selektor übergeben und sie gibt alle Elemente zurück, auf die der Selektor zutrifft[bs422b]. So lassen sich gleiche Strukturen einer Seite scrapen, z. B. Textblöcke einer Webseite. Nimmt man als Beispiel Artikel von der Seite tagesschau.de, lassen sich so die Texte eines Artikel extrahieren. Dieser Fall lässt sich wie folgt darstellen:

```
textabsatz = html.find_all(class_='textabsatz')
```

Als Selektor dient hier der Klassenname der jeweiligen HTML Tags. Eine Alternative zu `find_all` ist `find`. Mit dieser Funktion wird lediglich ein einzelner Wert zurückgegeben (bei mehreren Treffern der erste).

Es existieren weitere Funktionen zur Selektion. Diese ermöglichen z. B. die Selektion anhand von Regex oder die Navigation auf Eltern- oder Geschwisterknoten. Eine vollständige Beschreibung der verfügbaren Funktionen befindet sich in der Dokumentation von BS4[bs422a].

4 Fazit und Ausblick

Das Thema Web Scraping ist umfangreich. Der Abruf und das Parsen von HTML sind vergleichsweise einfach. Geht es um die Selektion bieten sich umfangreiche Möglichkeiten an. Das Paket BeautifulSoup (BS4) gilt als einfach anzuwenden, bietet aber dennoch eine Vielzahl von Möglichkeiten.

Dieses Tutorial beschreibt in aller Kürze die einzelnen Schritte, die für ein erfolgreiches Web Scraping, also den Bezug einzelner Daten aus Web Inhalten. Zur Umsetzung von brauchbaren Anwendungsfällen wird empfohlen, die mehrfach referenzierten Dokumentationen zu vertiefen und sich mit HTML und CSS vertraut zu machen.

Abkürzungsverzeichnis

BS4	BeautifulSoup 4
CSS	Cascading Style Sheets
HTML	Hypertext Markup Language

Quellenverzeichnis

[bs422a] https://www.crummy.com/software/BeautifulSoup/bs4/doc/

[bs422b] https://www.crummy.com/software/BeautifulSoup/bs4/doc/#calling-a-tag-is-like-calling-find-all

[css22] https://www.w3schools.com/cssref/css_selectors.asp

[moz22] https://developer.mozilla.org/de/docs/Learn/Getting_started_with_the_web/HTML_basics

[pyp22] https://pypi.org/project/requests/

[req22] https://requests.readthedocs.io/en/latest/