Einleitung

Was ist Data Science?

In diesem Buch geht es darum, Data Science mithilfe von Python zu betreiben, daher stellt sich unmittelbar die Frage: Was ist *Data Science* überhaupt? Das genau zu definieren, erweist sich als überraschend schwierig, insbesondere in Anbetracht der Tatsache, wie geläufig dieser Begriff inzwischen geworden ist. Von lautstarken Kritikern wird dieser Begriff mitunter als eine überflüssige Bezeichnung abgetan (denn letzten Endes kommt keine Wissenschaft ohne Daten aus) oder für ein leeres Schlagwort gehalten, das lediglich dazu dient, Lebensläufe aufzupolieren, um die Aufmerksamkeit übereifriger Personalverantwortlicher zu erlangen.

Meiner Ansicht nach übersehen diese Kritiker dabei einen wichtigen Punkt. Trotz des mit dem Begriff einhergehenden Hypes ist Data Science wohl die beste Beschreibung für fachübergreifende Fähigkeiten, die in vielen industriellen und akademischen Anwendungsbereichen immer wichtiger werden. Entscheidend ist hier die Interdisziplinarität: Ich halte Drew Conways Venn-Diagramm, das er im September 2010 erstmals in seinem Blog veröffentlichte, für die beste Definition von Data Science (siehe Abbildung 0.1).

Abb. 0.1: Das Venn-Diagramm zur Data Science von Drew Conway

Zwar sind einige der Bezeichnungen für die Schnittmengen nicht ganz ernst gemeint, aber dennoch erfasst dieses Diagramm das Wesentliche dessen, was gemeint ist, wenn man von »Data Science« spricht: Es handelt sich um ein grundlegend interdisziplinäres Thema. Data Science umfasst drei verschiedene und sich überschneidende Bereiche: die Aufgaben

eines Statistikers, der (immer größer werdende) Datenmengen modellieren und zusammenfassen kann, die Arbeit des Informatikers, der Algorithmen für die effiziente Speicherung, Verarbeitung und Visualisierung dieser Daten entwerfen kann, und das erforderliche Fachwissen – das wir uns als das »klassisch« Erlernte eines Fachgebiets vorstellen können –, um sowohl die angemessenen Fragen zu stellen als auch die Antworten im richtigen Kontext zu bewerten.

Das habe ich im Sinn, wenn ich Sie dazu auffordere, Data Science nicht als ein neu zu erlernendes Fachwissensgebiet zu begreifen, sondern als neue Fähigkeiten, die Sie im Rahmen Ihres vorhandenen Fachwissens anwenden können. Ob Sie über Wahlergebnisse berichten, Aktienrenditen vorhersagen, Mausklicks auf Onlinewerbung optimieren, Mikroorganismen auf Mikroskopbildern identifizieren, nach neuen Arten astronomischer Objekte suchen oder mit irgendwelchen anderen Daten arbeiten: Ziel dieses Buchs ist es, Ihnen die Fähigkeit zu vermitteln, neuartige Fragen über das von Ihnen gewählte Fachgebiet zu stellen und diese zu beantworten.

An wen richtet sich das Buch?

Sowohl in meinen Vorlesungen an der Universität Washington als auch auf verschiedenen technisch orientierten Konferenzen und Treffen wird mir am häufigsten diese Frage gestellt: »Wie kann man Python am besten erlernen?« Bei den Fragenden handelt es sich im Allgemeinen um technisch interessierte Studenten, Entwickler oder Forscher, die oftmals schon über umfangreiche Erfahrung mit dem Schreiben von Code und der Verwendung von rechnergestützten und numerischen Tools verfügen. Die meisten dieser Personen möchten Python erlernen, um die Programmiersprache als Tool für datenintensive und rechnergestützte wissenschaftliche Aufgaben zu nutzen. Für diese Zielgruppe ist eine Vielzahl von Lernvideos, Blogbeiträgen und Tutorials online verfügbar. Allerdings frustriert mich bereits seit geraumer Zeit, dass es auf obige Frage keine wirklich eindeutige und gute Antwort gibt – und das war der Anlass für dieses Buch.

Das Buch ist nicht als Einführung in Python oder die Programmierung im Allgemeinen gedacht. Ich setze voraus, dass der Leser mit der Programmiersprache Python vertraut ist. Dazu gehören das Definieren von Funktionen, die Zuweisung von Variablen, das Aufrufen der Methoden von Objekten, die Steuerung des Programmablaufs und weitere grundlegende Aufgaben. Das Buch soll vielmehr Python-Usern dabei helfen, die zum Betreiben von Data Science verfügbaren Pakete zu nutzen – Bibliotheken wie IPython, NumPy, Pandas, Matplotlib, Scikit-Learn und ähnliche Tools –, um Daten effektiv zu speichern, zu handhaben und Einblick in diese Daten zu gewinnen.

Warum Python?

Python hat sich in den letzten Jahrzehnten zu einem erstklassigen Tool für wissenschaftliche Berechnungen entwickelt, insbesondere auch für die Analyse und Visualisierung großer Datensätze. Die ersten Anhänger der Programmiersprache Python dürfte das ein wenig überraschen: Beim eigentlichen Design der Sprache wurde weder der Datenanalyse noch wissenschaftlichen Berechnungen besondere Beachtung geschenkt.

Dass sich Python für die Data Science als so nützlich erweist, ist vor allem dem großen und aktiven Ökosystem der Programmpakete von Drittherstellern zu verdanken: Da gibt es

NumPy für die Handhabung gleichartiger Array-basierter Daten, Pandas für die Verarbeitung verschiedenartiger und gekennzeichneter Daten, SciPy für gängige wissenschaftliche Berechnungen, Matplotlib für druckreife Visualisierungen, IPython für die interaktive Ausführung und zum Teilen von Code, Scikit-Learn für Machine Learning sowie viele weitere Tools, die später im Buch noch Erwähnung finden.

Falls Sie auf der Suche nach einer Einführung in die Programmiersprache Python sind, empfehle ich das dieses Buch ergänzende Projekt *A Whirlwind Tour of the Python Language* (`https://github.com/jakevdp/WhirlwindTourOfPython`). Hierbei handelt es sich um eine Tour durch die wesentlichen Features der Sprache Python, die sich an Data Scientists richtet, die bereits mit anderen Programmiersprachen vertraut sind.

Python 2 kontra Python 3

In diesem Buch wird die Syntax von Python 3 verwendet, die Spracherweiterungen enthält, die mit Python 2 inkompatibel sind. Zwar wurde Python 3 schon 2008 veröffentlicht, allerdings verbreitete sich diese Version insbesondere in den Communitys von Wissenschaft und Webentwicklung nur langsam. Das lag vor allem daran, dass die Anpassung vieler wichtiger Pakete von Drittherstellern an die neue Sprachversion Zeit benötigte. Seit Anfang 2014 gibt es jedoch stabile Versionen der für die Data Science wichtigsten Tools, die sowohl mit Python 2 als auch mit Python 3 kompatibel sind, daher wird in diesem Buch die neuere Syntax von Python 3 genutzt. Allerdings funktionieren die meisten Codeabschnitte dieses Buchs ohne Änderungen auch in Python 2. Wenn Py2-inkompatible Syntax verwendet wird, weise ich ausdrücklich darauf hin.

Inhaltsübersicht

Alle Kapitel in diesem Buch konzentrieren sich auf ein bestimmtes Paket oder Tool, das für die mit Python betriebene Data Science von grundlegender Bedeutung ist.

IPython und Jupyter (Kapitel 1)
: Diese Pakete bieten eine Umgebung für Berechnungen, die von vielen Data Scientists genutzt wird, die Python einsetzen.

NumPy (Kapitel 2)
: Diese Bibliothek stellt das `ndarray`-Objekt zur Verfügung, das ein effizientes Speichern und die Handhabung dicht gepackter Datenarrays in Python ermöglicht.

Pandas (Kapitel 3)
: Diese Bibliothek verfügt über das `DataFrame`-Objekt, das ein effizientes Speichern und die Handhabung gekennzeichneter bzw. spaltenorientierter Daten in Python gestattet.

Matplotlib (Kapitel 4)
: Diese Bibliothek ermöglicht flexible und vielfältige Visualisierungen von Daten in Python.

Scikit-Learn (Kapitel 5)
: Diese Bibliothek stellt eine effiziente Implementierung der wichtigsten und gebräuchlichsten Machine-Learning-Algorithmen zur Verfügung.

Natürlich umfasst die PyData-Welt viel mehr als diese fünf Pakete – und sie wächst mit jedem Tag weiter. Ich werde mich im Folgenden daher bemühen, Hinweise auf andere interessante Projekte, Bestrebungen und Pakete zu geben, die die Grenzen des mit Python Machbaren erweitern. Dessen ungeachtet sind die fünf genannten Pakete derzeit für viele der mit Python möglichen Aufgaben der Data Science von grundlegender Bedeutung, und ich erwarte, dass sie wichtig bleiben, auch wenn das sie umgebende Ökosystem weiterhin wächst.

Verwendung der Codebeispiele

Unter `https://github.com/jakevdp/PythonDataScienceHandbook` steht ergänzendes Material (Codebeispiele, Abbildungen usw.) zum Herunterladen zur Verfügung. Das Buch soll Ihnen helfen, Ihre Arbeit zu erledigen. Den im Buch aufgeführten Code können Sie generell in Ihren eigenen Programmen und der Dokumentation verwenden. Sie brauchen uns nicht um Erlaubnis zu fragen, solange Sie nicht erhebliche Teile des Codes nutzen. Wenn Sie beispielsweise ein Programm schreiben, das einige der im Buch aufgeführten Codeschnipsel verwendet, benötigen Sie dafür keine Erlaubnis. Der Verkauf oder Vertrieb einer CD-ROM, die Codebeispiele aus dem Buch enthält, bedarf hingegen einer Genehmigung. Das Beantworten von Fragen durch Verwendung von Zitaten oder Beispielcode aus diesem Buch muss nicht extra genehmigt werden. Die Verwendung erheblicher Teile des Beispielcodes in der Dokumentation Ihres eigenen Produkts erfordert jedoch eine Genehmigung.

Wir freuen uns über Quellennennungen, machen sie jedoch nicht zur Bedingung. Üblich ist die Nennung von Titel, Autor(en), Verlag, Erscheinungsjahr und ISBN, also beispielsweise *»Data Science mit Python« von Jake VanderPlas (mitp Verlag 2017), ISBN 978-3-95845-695-2*.

Installation der Software

Die Installation von Python und der für wissenschaftliche Berechnungen erforderlichen Bibliotheken ist unkompliziert. In diesem Abschnitt finden Sie einige Überlegungen, denen Sie bei der Einrichtung Ihres Computers Beachtung schenken sollten.

Es gibt zwar verschiedene Möglichkeiten, Python zu installieren, allerdings empfehle ich zum Betreiben von Data Science die Anaconda-Distribution, die unter Windows, Linux und macOS auf ähnliche Weise funktioniert. Es gibt zwei Varianten der Anaconda-Distribution:

- Miniconda (`http://conda.pydata.org/miniconda.html`) besteht aus dem eigentlichen Python-Interpreter und einem Kommandozeilenprogramm namens *conda*, das als plattformübergreifender Paketmanager für Python-Pakete fungiert. Das Programm arbeitet in ähnlicher Weise wie die Tools apt oder yum, die Linux-Usern bekannt sein dürften.
- Anaconda (`https://www.continuum.io/downloads`) enthält sowohl Python als auch conda und darüber hinaus eine Reihe vorinstallierter Pakete, die für wissenschaftliche Berechnungen konzipiert sind. Aufgrund der Größe dieser Pakete müssen Sie davon ausgehen, dass die Installation mehrere Gigabyte Speicherplatz auf der Festplatte belegt.

Alle in Anaconda enthaltenen Pakete können auch nachträglich der Miniconda-Installation hinzugefügt werden. Daher empfehle ich, mit Miniconda anzufangen.

Laden Sie zunächst das Miniconda-Paket herunter und installieren Sie es. Vergewissern Sie sich, dass Sie eine Version auswählen, die Python 3 enthält. Installieren Sie dann die in diesem Buch verwendeten Pakete:

```
[~]$ conda install numpy pandas scikit-learn matplotlib seaborn ipython-notebook
```

Wir werden im gesamten Buch noch weitere, spezialisiertere Tools verwenden, die zum wissenschaftlich orientierten Ökosystem in Python gehören. Für gewöhnlich ist zur Installation lediglich eine Eingabe wie `conda install paketname` erforderlich. Weitere Informationen über conda, beispielsweise über das Erstellen und Verwenden von conda-Umgebungen (die ich nur nachdrücklich empfehlen kann), finden Sie in der Onlinedokumentation (`http://conda.pydata.org/docs/`).

Konventionen dieses Buchs

In diesem Buch gelten die folgenden typografischen Konventionen:

Kursive Schrift
: Kennzeichnet neue Begriffe, Dateinamen und Dateinamenserweiterungen.

`Nicht proportionale Schrift`
: Wird für URLs, Programmlistings und im Fließtext verwendet, um Programmbestandteile wie Variablen- oder Funktionsbezeichnungen, Datenbanken, Datentypen Umgebungsvariablen, Anweisungen und Schlüsselwörter zu kennzeichnen.

`Fette nicht proportionale Schrift`
: Kommandos oder sonstiger Text, der vom User buchstabengetreu eingegeben werden soll.

`Kursive nicht proportionale Schrift`
: Text, der durch eigene Werte oder durch kontextabhängige Werte zu ersetzen ist.

Die Webseite zum Buch

Der Verlag hält auf seiner Website weiteres Material zum Buch bereit. Unter `http://www.mitp.de/695` können Sie sich Beispielcode herunterladen.