Steffen Herbold

Data Science Crashkurs

Eine interaktive
und praktische Einführung

Mit Jupyter Notebooks

dpunkt.verlag

Dr. Steffen Herbold ist Professor für Methoden und Anwendungen maschinellen Lernens am Institut für Software und Systems Engineering der Technischen Universität Clausthal, wo er die Forschungsgruppe AI Engineering leitet. Zuvor hat er an der Universität Göttingen promoviert und habilitiert und am Karlsruher Institut für Technologie einen Lehrstuhl vertreten. In der Forschung beschäftigt er sich mit der Entwicklung und Qualitätssicherung der Lösung von Problemen durch maschinelles Lernen, z.B. zur effizienteren Softwareentwicklung, der Prognose von Ernteerträgen oder auch der Erkennung von aeroakustischen Geräuschquellen.

Steffen Herbold

Data-Science-Crashkurs

Eine interaktive und praktische Einführung

dpunkt.verlag

Steffen Herbold
steffen.herbold@tu-clausthal.de

Lektorat: Christa Preisendanz
Lektoratsassistenz: Anja Weimer
Copy-Editing: Ursula Zimpfer, Herrenberg
Layout & Satz: Birgit Bäuerlein
Herstellung: Stefanie Weidner
Umschlaggestaltung: Helmut Kraus, *www.exclam.de*
Druck und Bindung: mediaprint solutions GmbH, 33100 Paderborn

Bibliografische Information der Deutschen Nationalbibliothek
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über *http://dnb.d-nb.de* abrufbar.

ISBN:
Print 978-3-86490-862-0
PDF 978-3-96910-618-1
ePub 978-3-96910-619-8
mobi 978-3-96910-620-4

Hinweis:
Dieses Buch wurde auf PEFC-zertifiziertem Papier aus nachhaltiger Waldwirtschaft gedruckt. Der Umwelt zuliebe verzichten wir zusätzlich auf die Einschweißfolie.

Schreiben Sie uns:
Falls Sie Anregungen, Wünsche und Kommentare haben, lassen Sie es uns wissen: *hallo@dpunkt.de*.

5 4 3 2 1 0

Vorwort

Willkommen beim Data-Science-Crashkurs. Wenn Sie bereits öfter etwas von Big Data, maschinellem Lernen, der künstlichen Intelligenz oder Data Science gehört haben und wissen wollen, welche Methoden sich hinter diesen Begriffen verbergen, sind Sie hier genau richtig. Dieses Buch richtet sich an alle, die mehr über die Möglichkeiten der Datenanalyse lernen wollen, ohne gleich tief in die Theorie oder bestimmte Methoden einzusteigen. Auch wenn Sie sich am besten schon etwas in der Informatik und/oder Mathematik auskennen, so kann man den Großteil auch verstehen, wenn man sich einfach nur für Daten interessiert und vor Mathe in der Schule keine Angst hatte.

Wir fangen an, indem wir die Definitionen der Begriffe einführen und dann betrachten, wie Data-Science-Projekte üblicherweise ablaufen. Dann geht's los mit den Daten. Zuerst lernen wir die Daten selbst mithilfe von Statistiken und Visualisierungen kennen. Sodann tauchen wir in die Welt der Algorithmik ein: Assoziationsregeln zum Auffinden von Beziehungen, Clustering, um Gruppen ähnlicher Daten zu finden, Klassifikation, um Kategorien zuzuweisen, Regression, um Zusammenhänge zu lernen, Zeitreihenanalyse, um zeitliche Zusammenhänge auszunutzen. Im letzten Teil betrachten wir, wie wir Texte zu Zahlen werden lassen, mit denen wir rechnen können, welche Rolle die Statistik für die Bewertung von Ergebnissen spielt und wie man mit Big Data arbeiten kann.

Wenn Sie wollen, können Sie sich nur mit den Methoden beschäftigen. Alle Methoden werden aber auch praktisch demonstriert: Im ganzen Buch befinden sich Quelltextbeispiele und das Ergebnis der Ausgabe. Hierdurch können Sie gleichzeitig verstehen, welche Methoden es gibt und wie Sie diese anwenden. Das Buch selbst wurde komplett mithilfe von *Jupyter Notebooks* geschrieben. In diesen Notebooks können Sie direkt im Webbrowser den Quelltext selbst ausführen, um die Ergebnisse interaktiv nachzuvollziehen. Sie können den Quelltext hierbei auch beliebig anpassen, zum Beispiel, um besser zu verstehen, was passiert, wenn sich Parameter eines Modells ändern. Im Anhang wird erklärt, wie Sie sich das Buch auf Ihrem eigenen Rechner »installieren«.

Am Ende der meisten Kapitel gibt es praktische Übungen, die Sie zur Vertiefung und für ein besseres Verständnis bearbeiten können. Beispiellösungen für diese Aufgaben können Sie sich in der Onlineversion des Buches anschauen[1]. Wenn Sie mal unterwegs sind, können Sie auch einfach in der Onlineversion weiterlesen. Für einige Kapitel gibt es keine Übungen. In diesen Kapiteln geht es großteils um Definitionen (Kap. 1 und 2) oder einen Ausblick (Kap. 13). Bei Kapitel 12 wird eine Big-Data-Umgebung benötigt. Auch wenn man hierfür kleinere Beispiele definieren könnte, müssten sowohl eine Python- als auch eine Java-Umgebung für eine Übung eingerichtet und passend konfiguriert werden, was aber über den bei einem Crashkurs angebrachten Aufwand hinausgeht.

Mit Quellenangaben wird in diesem Buch insgesamt eher sparsam umgegangen: Das Ziel ist ein breiter Überblick und ein Verständnis des Themas. Für die Vertiefung gibt es zu jedem Kapitel, häufig sogar zu den Abschnitten innerhalb der Kapitel, ausreichend eigene Fachliteratur. Welche Bücher jeweils für Sie geeignet sind, kann man nicht pauschal sagen, das hängt vom jeweiligen Ziel und Wissensstand der Leserschaft ab. Durch die aktuelle Verbreitung findet man zu jedem Thema auch zusätzliche Informationen im Internet, wenn man nach den entsprechenden Begriffen sucht. Die meisten Quellen, die hier im Buch genannt sind, verweisen auf besonders wichtige Definitionen oder Anwendungen; in wenigen Fällen, wenn ein Thema wirklich nur sehr kurz behandelt wird, auch zur weiterführenden Fachliteratur. In der Printversion des Buches sind diese Quellen durch die Autoren, Titel und Jahreszahlen angegeben, wie es in Literaturlisten üblich ist. In der Onlineversion wird direkt auf die Quellen verlinkt, falls möglich mithilfe von *Document Object Identifiern* (DOIs): Dies sind persistente Identifier, die auch in vielen Jahren noch funktionieren sollten und zu den Homepages der Verlage weiterleiten.

Der Fokus des Buches liegt darauf, wie Analysen für Daten erstellt werden. Drei für die Anwendung wichtige Aspekte betrachten wir nicht: Wie man Daten sammelt, wie man Daten aus einer Datenbank laden kann und den operativen Einsatz von erstellten Analysen. Das Datensammeln ist von Anwendungsfall zu Anwendungsfall verschieden. Oft liegen schon Daten vor. Andernfalls muss individuell auf Basis der Problemstellung eine Lösung entwickelt werden. Das Laden von Daten hängt vom Datenformat ab. Bei kleineren Projekten werden häufig CSV-Dateien verwendet, wie wir es in den Übungen machen. Aber auch andere Formate wie JSON sind bei Dateien üblich. Hier muss man einfach bei den Bibliotheken nach den entsprechenden Möglichkeiten zum Laden suchen. Im Fall von Datenbanken kann man die Daten häufig mit der Anfragesprache SQL laden. SQL ist für sich genommen jedoch bereits ein Thema, das ganze Bücher füllt. Da das Laden von Daten für unseren Crashkurs sekundär ist, verzichten wir daher auf

1. *https://data-science-crashkurs.de*

eine Einführung in Datenbanken und SQL. Beim operativen Einsatz geht es nicht nur um die Modelle und ihre Güte, sondern auch darum, wie diese in ein Softwaresystem eingebunden werden, wie dieses System getestet wird und wie hiermit bei einer Continuous Integration umgegangen wird. Hierbei handelt es sich um für die Softwaretechnik wesentliche Fragen, die jedoch die Analysen nicht direkt beeinflussen.

In diesem Buch verwenden wir vorwiegend die weibliche Bezeichnung von Rollen. Die Ausnahme ist die Rolle des Data Scientist. Im Englischen ist das Wort geschlechtslos und die Begriffe »Scientistin« oder »die Data Scientist« klingen eher komisch.

Ich wünsche Ihnen, liebe Leserinnen und Leser, viel Spaß und einen maximalen Erkenntnisgewinn mit diesem Buch.

Über Ihr Feedback würde ich mich freuen (*steffen.herbold@tu-clausthal.de*).

Steffen Herbold
Clausthal-Zellerfeld, Oktober 2021

Inhaltsübersicht

Inhaltsverzeichnis

1 Big Data und Data Science

Zuerst wollen wir uns etwas mit Begriffen beschäftigen, um zu verstehen, worum es beim Thema Data Science geht. Aufbauend auf dem Begriff Big Data wird aufgezeigt, was eigentlich alles zu Data Science gehört und welche Fähigkeiten Data Scientists benötigen.

1.1 Einführung in Big Data

Den Begriff *Big Data* gibt es jetzt bereits seit einigen Jahren und der ursprüngliche mit diesem Thema verbundene Hype ist längst Vergangenheit. Stattdessen gibt es neue Buzzwords, wie das *Internet der Dinge* (engl. *Internet of Things*), die *künstliche Intelligenz* (engl. *Artificial Intelligence*), und hierbei insbesondere auch die *tiefen neuronalen Netze* (engl. *Deep Neural Network*, *Deep Learning*). Nichtsdestotrotz ist Big Data mit diesen neuen Themen eng verbunden und häufig eine Voraussetzung oder zumindest eine verwandte Technologie.

Trotz der anhaltenden Relevanz des Themas ist dennoch häufig kein gutes Verständnis für den Unterschied zwischen vielen Daten und Big Data vorhanden. Ein gutes Verständnis der Besonderheiten und Eigenschaften von Big Data und von den damit verbundenen Implikationen und Problemen ist jedoch zwingend notwendig, wenn man auf Big Data aufbauende Technologien in Projekten einsetzen will. Der Grund für Missverständnisse rund um den Begriff Big Data ist einfach: Wir denken intuitiv an »große Datenmengen«. Eine derart vereinfachte Begriffsdefinition ignoriert jedoch wesentliche Aspekte von Big Data. Backups sind ein gutes Beispiel für große Datenmengen, die nicht Big Data sind. In modernen Rechenzentren werden Backups auf Hintergrundspeichern mit einer hohen Bitstabilität, aber auch einer hohen Latenz gespeichert. Dort lagern häufig riesige Datenmengen in der Hoffnung, dass sie nie gebraucht werden, bevor sie gelöscht oder überschrieben werden. Es gibt noch einen weiteren Grund, warum es unpraktisch ist, Big Data nur über das Datenvolumen zu definieren: Wir müssten die Definition ständig anpassen, da die Speicherkapazitäten, die Rechenkraft und der Arbeitsspeicher stetig wachsen.

Eine bessere und allgemein akzeptierte Definition für Big Data basiert auf den *drei Vs*[1].

Definition von Big Data:

Als Big Data bezeichnet man Daten, die ein hohes *Volumen*, eine hohe *Geschwindigkeit* (engl. *velocity*) und eine hohe *Vielfalt* (engl. *variety*) haben, sodass man kosteneffiziente und innovative Informationsverarbeitungsmethoden benötigt, um Wissen zu generieren, Entscheidungen zu treffen oder Prozesse zu automatisieren.

Zum besseren Verständnis zerlegen wir nun diese Definition in ihre Einzelteile und betrachten diese genauer. Hierbei wird klar werden, warum Big Data mehr ist als nur Datenvolumen.

1.1.1 Volumen

Auch wenn das Datenvolumen nicht der einzig wichtige Faktor ist, ist es dennoch entscheidend. Nicht umsonst heißt es Big Data. Dass keine bestimmte Datengröße das Kriterium sein kann, wird schon klar, wenn man sich überlegt, dass Google die Forschungsarbeit, in der MapReduce vorgestellt wurde, bereits 2006 publiziert hat [Dean & Ghemawat 2008]. Zu diesem Zeitpunkt war ein Terabyte noch ein sehr großes Datenvolumen. Im Jahr 2021 ist dies lediglich die Festplattengröße des Laptops, auf dem dieses Buch geschrieben wurde. Ein weiteres Beispiel ist das Wachstum des Datenvolumens, das im Internet jährlich übertragen wird (Abb. 1–1).

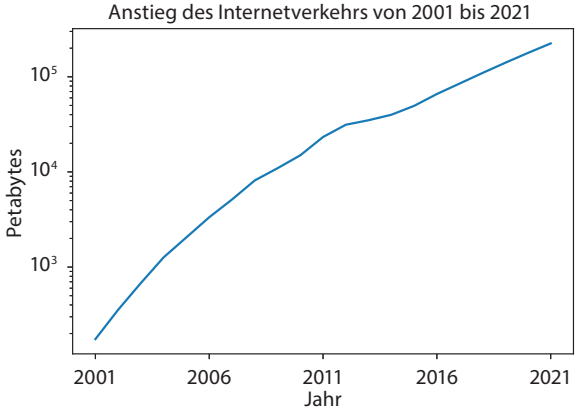

Abb. 1–1 *Wachstum des Datenvolumens im Internetverkehr*

1. *https://www.gartner.com/en/information-technology/glossary/big-data*

Eine vereinfachte Richtlinie für das Datenvolumen lautet, dass es mehr Daten sein müssen, als in den Arbeitsspeicher moderner Server passen. Besser ist es jedoch, wenn man sich einfach die Frage stellt, ob es möglich ist, die Daten (oft) zu kopieren, insbesondere auch über Netzwerkverbindungen. Ist dies nicht mehr der Fall, handelt es sich vermutlich um genug Daten, um von Big Data zu sprechen. In extremen Fällen sind die Daten sogar so groß, dass man sie gar nicht über das Netzwerk kopieren kann. Stattdessen nutzt man das *Sneaker Net*[2]: Die Daten werden direkt auf Festplatten verschickt. In Bezug auf den Datendurchsatz ist ein mit Festplatten beladenes Transportflugzeug unschlagbar. Die Latenz lässt jedoch zu wünschen übrig. Ein Beispiel für eine Anwendung, die ohne das Sneaker Net nicht geklappt hätte, ist die Erstellung des ersten Bilds von einem schwarzen Loch [Whitwam 2019].

1.1.2 Velocity/Geschwindigkeit

Die Velocity ist die *Geschwindigkeit,* mit der neue Daten generiert, verarbeitet und/oder ausgewertet werden müssen. Es gibt viele Beispiele für Daten, die eine hohe Geschwindigkeit haben, zum Beispiel die durch Sensoren wie LIDAR und Kameras erfassten Daten von autonomen Fahrzeugen. Derartige Daten können in kürzester Zeit ein sehr hohes Volumen erreichen. Die Firma Waymo hat zum Beispiel einen zwei Terabyte großen Datensatz, der während elf Fahrstunden gesammelt wurde, veröffentlicht[3]. Daten, die mehr oder weniger kontinuierlich in hoher Geschwindigkeit generiert werden, nennt man auch *Streamingdaten*.

Eine besondere Schwierigkeit beim Umgang mit Streamingdaten besteht darin, dass diese oft in nahezu Echtzeit verarbeitet werden müssen. Beim autonomen Fahren ist dies sofort klar, schon alleine wegen der Sicherheit. Doch das gilt auch für viele andere Anwendungen, zum Beispiel für das Sortieren des Nachrichtenstreams in sozialen Netzwerken. Hier kommt zwar niemand zu schaden, die Nutzer würden einen Dienst aber schnell verlassen, wenn die Ladezeiten zu lang sind. Entsprechend müssen beim Umgang mit Streamingdaten in der Regel zwei Anforderungen gleichzeitig erfüllt werden: Daten müssen sehr schnell empfangen werden und dürfen dann auch nicht lange in einem Zwischenspeicher landen bzw. sich dort befinden, sondern müssen sofort verarbeitet und ausgewertet werden. Hierdurch ergibt sich eine Art inverse Korrelation zwischen der Geschwindigkeit und dem Datenvolumen: Je höher die Geschwindigkeit, desto weniger Daten reichen aus, um Daten zu Big Data werden zu lassen. Oder an einem Beispiel: Ein Gigabyte pro Tag zu verarbeiten ist einfacher als ein Gigabyte pro Sekunde.

2. *https://en.wikipedia.org/wiki/Sneakernet*
3. *https://waymo.com/open/*

1.1.3 Variety/Vielfalt

Die Vielfalt der Daten ist der dritte große Aspekt von Big Data. Mittlerweile ist die Analyse von Bildern, Videos und Texten zu einer normalen Anwendung geworden. Dies war jedoch noch nicht der Fall, als der Begriff Big Data geprägt wurde. Im Zeitraum um die Jahrtausendwende lagen die Daten, die analysiert werden sollten, üblicherweise strukturiert vor, zum Beispiel in relationalen Datenbanken. Die Daten waren entweder numerisch oder in feste Kategorien eingeteilt. Das änderte sich im Laufe der 2000er-Jahre, dadurch dass das Internet allgegenwärtig wurde und wir immer mehr Computertechnik in unseren Alltag übernommen haben, zum Beispiel in Form von Smartphones. Hier entstanden Daten eher auf unstrukturierte Weise, zum Beispiel durch Webseiten, die ad hoc von Nutzern erstellt wurden. Es ist daher kein Zufall, dass Google den Begriff Big Data und die damit verbundenen Technologien mitgeprägt hat: Die Indizierung des stetig wachsenden und komplexer werdenden Internets zwang dazu, die vorhandenen Techniken rapide weiterzuentwickeln. Hierbei mussten nicht nur immer größere Datenmengen verarbeitet werden, sondern vor allem eine Vielfalt von Datenformaten, insbesondere Text- und Bilddaten, später auch Videodaten.

Insgesamt gibt es viel mehr unstrukturierte Daten als strukturierte Daten. Dies wird üblicherweise als Pyramide dargestellt (Abb. 1–2), in der zwischen *unstrukturierten*, *quasistrukturierten*, *semistrukturierten* und *strukturierten* Daten unterschieden wird.

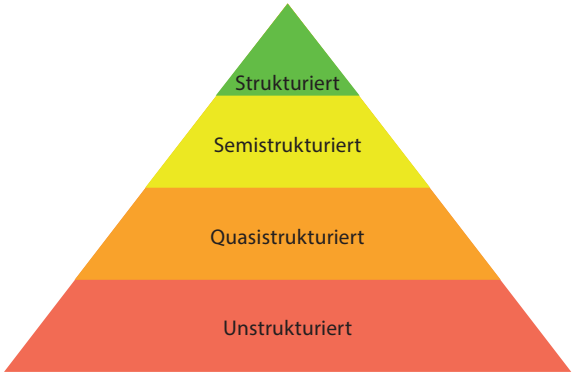

Abb. 1–2 *Datenpyramide*

An der Spitze der Pyramide sind die strukturierten Daten, zum Beispiel Tabellen in relationalen Datenbanken, *Comma-Separated-Value*-(CSV-)Dateien und Ähnliches. Strukturierte Daten kann man im Normalfall direkt in ein Analysetool laden, ohne dass Vorverarbeitungsschritte notwendig sind. Die Vorverarbeitung beschränkt sich daher bei strukturierten Daten höchstens auf Aufgaben wie das Säubern der Daten, um beispielsweise ungültige Datenpunkte oder Ausreißer zu filtern.

Als Nächstes kommen die semistrukturierten Daten, zum Beispiel XML und JSON. Der Hauptunterschied zwischen strukturierten und semistrukturierten Daten ist die Flexibilität der Datenformate. Bei strukturierten Daten ist beispielsweise in der Regel der Typ einer Spalte fest definiert. Dies ist bei semistrukturierten Daten anders: Hier trifft man oftmals auf verschachtelte Strukturen, die man für die Analyse erst aufbrechen muss. Außerdem gibt es häufig optionale Felder, wodurch die Verarbeitung komplizierter werden kann. Dennoch kann man mit semistrukturierten Daten überwiegend einfach arbeiten, da diese auch ohne großen Aufwand in viele Analyseumgebungen importiert werden können.

Im Allgemeinen gilt für strukturierte und semistrukturierte Daten gemeinsam, dass es feste Datenformate und/oder Anfragesprachen gibt, mit denen man einfach die benötigten Informationen extrahieren und laden kann. Dies ist in den beiden unteren Ebenen der Pyramide nicht mehr der Fall. Quasistrukturierte Daten haben zwar eine fest definierte Struktur, es ist aber ein gewisser Aufwand erforderlich, um an die benötigten Informationen zu kommen. Als Beispiel betrachten wir die Ausgabe des Befehls `ls -l`, mit dem man sich in einem Linuxterminal die Dateien in einem Ordner anzeigen lassen kann.

```
%ls -l
```

```
total 6792
drwxr-xr-x 1 sherbold sherbold     512 Mar 24 14:23 data/
-rw-r--r-- 1 sherbold sherbold    2957 May  5 14:30 howto.ipynb
drwxr-xr-x 1 sherbold sherbold     512 Apr 27 17:04 images/
-rw-r--r-- 1 sherbold sherbold   63683 May  6 11:52 kapitel_01.ipynb
-rw-r--r-- 1 sherbold sherbold  113117 May 10 10:11 kapitel_02.ipynb
-rw-r--r-- 1 sherbold sherbold   24576 May 10 10:08 kapitel_03.ipynb
-rw-r--r-- 1 sherbold sherbold  886609 May 10 10:10 kapitel_04.ipynb
-rw-r--r-- 1 sherbold sherbold   39543 May  6 16:44 kapitel_05.ipynb
-rw-r--r-- 1 sherbold sherbold 1664391 May  6 18:50 kapitel_06.ipynb
-rw-r--r-- 1 sherbold sherbold 2679078 May 10 09:05 kapitel_07.ipynb
-rw-r--r-- 1 sherbold sherbold  199328 May 10 09:23 kapitel_08.ipynb
-rw-r--r-- 1 sherbold sherbold  446103 May 10 09:39 kapitel_09.ipynb
-rw-r--r-- 1 sherbold sherbold  579843 May 10 09:50 kapitel_10.ipynb
-rw-r--r-- 1 sherbold sherbold  148082 May 10 09:58 kapitel_11.ipynb
-rw-r--r-- 1 sherbold sherbold   54300 May  6 12:11 kapitel_12.ipynb
-rw-r--r-- 1 sherbold sherbold    5967 May  6 15:16 kapitel_13.ipynb
-rw-r--r-- 1 sherbold sherbold    4927 May  5 14:30 notations.ipynb
-rw-r--r-- 1 sherbold sherbold    3887 May  6 11:59 vorwort.ipynb
```

Man sieht eine klare Struktur in den Daten: Die meisten Zeilen beinhalten die Benutzerrechte, gefolgt von der Anzahl der Symlinks auf die Datei, dem Benutzer und der Gruppe, die die Datei besitzen, der Dateigröße, dem Datum der letzten Änderung und zuletzt dem Namen. Diese Struktur kann man nutzen, um einen Parser zu schreiben, der die Daten einliest, zum Beispiel mithilfe von einem regulären Ausdruck. Wir können also eine Struktur über quasistrukturierte Daten legen, indem wir die Struktur durch einen Parser selbst definieren. Dies ist zwar mehr Aufwand

als bei strukturierten und semistrukturierten Daten, aber man kommt dennoch zuverlässig an die benötigten Informationen. Trotzdem kann es sehr leicht passieren, dass sich die Struktur ändert und der selbst geschriebene Parser nicht mehr funktioniert. Daher ist das Lesen von quasistrukturierten Daten fehleranfällig, da man eventuell Sonderfälle übersieht oder sich das Datenformat ändern kann. Man sollte also den oft signifikanten Wartungsaufwand bei der Verarbeitung von quasistrukturierten Daten für die Nutzung im Produktivbetrieb berücksichtigen.

Die unstrukturierten Daten sind auf der untersten Ebene der Pyramide. Hier befindet sich der Großteil der Daten: Bilder, Videos und Text. Die Herausforderung bei diesen Daten ist es, eine Struktur zu bestimmen, die man später verarbeiten kann. Hier gibt es keine Faustformel, es hängt stattdessen von den konkreten Daten und der geplanten Anwendung ab. Hinzu kommt, dass unstrukturierte Daten häufig vermischt sind. Dieses Buch ist ein gutes Beispiel: Wir haben eine Mischung aus natürlicher Sprache, Bildern, Formatinformationen (z.B. Überschriften, Listen) und Quelltext.

1.1.4 Innovative Informationsverarbeitungsmethoden

Auch wenn die drei *Vs* als die zentralen Eigenschaften von Big Data angesehen werden, sind die anderen Teile der Definition auch wichtig, um zu verstehen, dass Big Data mehr ist als einfach nur viele Daten, die möglicherweise schnell generiert werden und verschiedene Formate haben. Der nächste Teil der Definition spricht von dem Bedarf an *innovativen Informationsverarbeitungsmethoden*. Das bedeutet, dass man für Big Data nicht einen normalen Arbeitsplatzrechner oder sogar ein traditionelles Batch-System in einem Großrechner, in dem sich viele Rechenknoten einen Speicher über das Netzwerk teilen, nutzen kann. Stattdessen ist die *Datenlokalität* (engl. *data locality*) wichtig, da man in der Regel keine Kopien der Daten über das Netzwerk erzeugen kann. Dies hat zu einer Transformation geführt, sodass es immer mehr Infrastrukturen gibt, in denen Rechenkraft und schneller, verteilter Speicher direkt integriert sind. Als Big Data ein neues Konzept war, gab es solche Technologien noch nicht. Heutzutage hat man viele Möglichkeiten, allein bei der Apache Foundation[4] gibt es unter anderem Hadoop, Spark, Storm, Kafka, Cassandra, Hive, HBase, Giraph und viele weitere Technologien, die hochprofessionell als Open Source entwickelt und von vielen Unternehmen zur Verarbeitung von Big Data eingesetzt werden.

4. *https://www.apache.org/*

1.1.5 Wissen generieren, Entscheidungen treffen, Prozesse automatisieren

Der letzte Teil der Big-Data-Definition bedeutet, dass Big Data kein Selbstzweck ist. Man spricht nur dann von Big Data, wenn man die Daten auch zum Erreichen eines Ziels nutzt. Ziele können der reine Erkenntnisgewinn sein, die Unterstützung der Entscheidungsfindung oder sogar die Automatisierung ganzer Geschäftsprozesse. Dieser Aspekt der Definition ist so wichtig, dass er häufig als weiteres V betrachtet wird: *Value*.

1.1.6 Noch mehr Vs

Die Definition von Gartner, die wir hier im Buch verwenden, hat »nur« drei Vs. Die Definition von Big Data durch Wörter, die mit V anfangen, ist jedoch so populär, dass es verschiedene Erweiterungen gibt mit bis zu 42 (!) Vs [Shafer 2017]. Die 42 Vs sollte man aber eher als Satire verstehen, die zeigen soll, dass mehr Vs nicht immer zu einer besseren Definition führen. Nichtsdestotrotz gibt es seriöse Definitionen mit bis zu zehn Vs[5]. Ein zusätzliches V hatten wir bereits: *Value*, also die Wertschöpfung durch Big Data. Die *Korrektheit* (engl. *veracity*) ist ein weiteres V, was häufig als hochrelevant eingeschätzt wird. Je mehr Daten man auswertet, desto schwieriger wird es, sicherzustellen, dass die Daten zuverlässig sind und sich Ergebnisse reproduzieren lassen. Dies ist insbesondere dann schwer, wenn sich die Datenquellen oft ändern, zum Beispiel bei der Analyse von Nachrichten oder der sozialen Medien. Volume, Velocity, Variety, Veracity und Value zusammen ergeben die Fünf-V-Definition von Big Data, die stark verbreitet ist. Weitere Vs betrachten wir an dieser Stelle nicht mehr.

1.2 Einführung in Data Science

Auch wenn der Begriff *Data Science* als Buzzword sehr populär ist, existiert noch keine allgemein akzeptierte Definition. Hierfür gibt es vermutlich zwei Gründe: Erstens ist der Begriff sehr generisch, sodass jede Verwendung von Daten, die in irgendeiner Form als »wissenschaftlich« betrachtet wird, als Data Science bezeichnet werden kann. Und zweitens gibt es einen großen Hype um diesen Begriff, weshalb Firmen, Fördermittelgeber und öffentliche Institutionen mit diesem Begriff Werbung für sich betreiben wollen.

Aus diesem Grund können wir hier leider auch keine kurze und einprägsame Definition für diesen Begriff geben. Stattdessen betrachten wir, welche Konzepte unter anderem als Data Science angesehen werden, und schauen uns Beispiele für Data-Science-Anwendungen an.

---

5. *https://tdwi.org/articles/2017/02/08/10-vs-of-big-data.aspx*

1.2.1 Was gehört zu Data Science?

Data Science kombiniert Methoden aus der Mathematik, Statistik und Informatik mit dem Ziel, datengetriebene Anwendungen zu entwickeln oder Wissen zu generieren. Die Wertschöpfung ist also sehr ähnlich zu Big Data. Der Hauptunterschied zwischen den Begriffen liegt auf dem Fokus auf große Datenmengen bei Big Data, der bei Data Science nicht gegeben sein muss.

Mathematik ist häufig das Fundament, auf dem die Datenanalyse definiert wird. Die Modelle über die Daten, die erstellt werden, sind im Endeffekt nichts anderes als eine mathematische Beschreibung von Aspekten der Daten. Man könnte also Data Science als mathematische Modellierung verstehen. Die Rolle der Mathematik ist jedoch größer als die einer »Beschreibungssprache« für Modelle. Teilgebiete der Mathematik liefern die Methoden, die man braucht, um Modelle zu bestimmen.

- *Optimierung* beschreibt, wie man die optimale Lösung für eine Zielfunktion findet, sodass die gefundene Lösung gewisse Nebenbedingungen erfüllt. Die Zielfunktion und die Nebenbedingungen werden bei Data Science häufig direkt aus den Daten ermittelt, sodass die Lösung optimal für die gegebenen Daten ist.
- *Stochastik* ist ein Teilgebiet der Mathematik, das sich mit zufälligen Verhalten durch Zufallsvariablen und stochastischen Prozessen befasst. Stochastik bildet daher eine wichtige Grundlage für die Theorie des maschinellen Lernens, und stochastische Modelle werden häufig genutzt, um Daten zu beschreiben.
- Ohne die *Geometrie* könnte man keine Daten, die räumlich verteilt sind, analysieren, zum Beispiel geografische Daten oder der 3-dimensionale Raum vor einem Fahrzeug.
- *Wissenschaftliches Rechnen* wird immer häufiger nicht nur genutzt, um Daten für Analysen zu simulieren, sondern auch, um Modelle über Daten durch Simulation zu bestimmen.

Die Statistik befasst sich mit der Analyse von Daten durch Stichproben, zum Beispiel das Schätzen der den Daten zugrunde liegenden Verteilungen, Zeitreihenanalysen oder die Entwicklung von statistischen Tests, mit denen man auswerten kann, ob Effekte zufällig oder signifikant sind.

- *Lineare Modelle* sind eine vielfältig einsetzbare Methode, um Daten zu beschreiben, um daraus Zusammenhänge zu erkennen oder Trends zu ermitteln.
- *Inferenz* ist ein ähnliches Verfahren, nur dass Wahrscheinlichkeitsverteilungen statt linearer Modelle genutzt werden, um die Daten zu beschreiben.
- *Zeitreihenanalyse* nutzt die interne Struktur von Daten, die über die Zeit gemessen wurden. Hierbei werden zum Beispiel saisonale Effekte oder andere sich wiederholende Muster genutzt, um die Struktur der Zeitreihe zu modellieren und zukünftige Werte vorherzusagen.

Ohne die Informatik wären die mathematischen und statistischen Verfahren nicht durch Computer ausführbar. Doch auch die theoretische Informatik liefert wichtige Grundlagen für Data Science.

- *Datenstrukturen und Algorithmen* sind die Grundlage von jeder effizienten Umsetzung von Algorithmen. Ohne das Verständnis von Datenstrukturen, wie Bäumen, Hashmaps und Listen, sowie von der Laufzeitkomplexität von Algorithmen wären Datenanalysemethoden nicht skalierbar auf große Datenmengen.
- Die *Informationstheorie* liefert das nötige Verständnis über die Entropie (Unsicherheit in den Daten) und gemeinsame Information von Daten und ist damit die Grundlage für viele Algorithmen des maschinellen Lernens.
- *Datenbanken* werden benötigt, um Daten effizient zu speichern und zugreifbar zu machen. SQL ist als Anfragesprache nicht nur bei relationalen Datenbanken, sondern auch bei NoSQL-Datenbanken allgegenwärtig.
- *Paralleles und verteiltes Rechnen* sind notwendig, um das Arbeiten mit großen Datenmengen und hoher Rechenkraft zu ermöglichen.
- Die klassische *künstliche Intelligenz* liefert die Grundlagen für die logische Modellierung und die Definition von Regelsystemen für viele Data-Science-Anwendungen. In diesem Buch unterscheiden wir explizit zwischen künstlicher Intelligenz und maschinellem Lernen. Wir benutzen den Begriff *künstliche Intelligenz,* um Anwendungen wie Deep Blue [Newborn 1997], die regelbasierte Schach-Engine, die als erster Computer Gary Kasparow im Schach besiegt hat, zu beschreiben.
- *Softwaretechnik* ist für die Operationalisierung von Anwendungen und das Management von Data-Science-Projekten notwendig.

Und zuletzt gibt es natürlich noch das *maschinelle Lernen*, was häufig auch als definierender Aspekt von Data Science gesehen wird. Das maschinelle Lernen kombiniert Mathematik, Statistik und Informatik auf geschickte Art und Weise. Je nachdem welche Methoden man betrachtet, können alle drei Disziplinen das maschinelle Lernen für sich beanspruchen. Durch maschinelles Lernen versucht man, Wissen aus Daten zu gewinnen, sodass das Wissen die Daten nicht nur beschreibt, sondern auch auf weitere Daten und Applikationen angewendet werden kann, zum Beispiel durch neuronale Netze, Entscheidungsbäume und Mustererkennung.

1.2.2 Beispielanwendungen

So unterschiedlich wie die Grundlagen von Data Science sind, so verschieden sind auch die Anwendungen in der Forschung, Industrie und Gesellschaft. Hier sind fünf kurze Beispiele:

- *Alpha Go* ist ein Beispiel für ein intelligentes selbstlernendes System. Vor einigen Jahren überraschte Alpha Go die Fachwelt, weil es scheinbar aus dem Nichts kam und einen der weltbesten Spieler im Go besiegte. Go gilt als besonders schwieriges Spiel, zum Beispiel im Verhältnis zu Schach, und Computer waren bis dahin gerade mal auf dem Niveau von Amateuren und stellten keine Herausforderung für professionelle Spieler dar. Alpha Go kombinierte klassische regelbasierte künstliche Intelligenz mit statistischen Monte-Carlo-Simulationen und selbstlernenden neuronalen Netzen, um dies zu erreichen.
- Die *Robotik* nutzt maschinelles Lernen, um den Robotern beizubringen, sich zu bewegen. Mit der Zeit können Roboter zum Beispiel lernen, durch welche Bewegungen sie das Umfallen verhindern können [Hwangbo et al. 2019].
- *Marketing* setzt auf Data Science, um im Internet gezielt Werbung schalten zu können. Die dahinter liegende Industrie erwirtschaftet Milliarden, indem Nutzern relevante Werbung basierend auf ihrem Verhalten im Internet gezeigt wird.
- In der *Medizin* werden Daten immer häufiger genutzt, um Entscheidungen zu unterstützen. IBM Watson, das erste Computerprogramm, das Menschen im Jeopardy besiegt hat[6], wird heutzutage zum Beispiel auch eingesetzt, um geeignete Krebstherapien auszuwählen[7]. (Auch wenn das nicht so gut klappt, wie man es sich ursprünglich erhofft hat [Strickland 2019].)
- Im *autonomen Fahren* wird maschinelles Lernen für verschiedene Aufgaben genutzt, zum Beispiel für die Erkennung von Objekten, wie Verkehrsschildern, anderen Autos oder Fußgängern.

6. *https://www.ibm.com/ibm/history/ibm100/us/en/icons/watson/*
7. *https://www.ibm.com/marketplace/clinical-decision-support-oncology*

1.3 Fähigkeiten von Data Scientists

Data Scientists sind weder Informatikerinnen, Mathematikerinnen, Statistikerinnen oder Domänenexpertinnen. Der perfekte Data Scientist bringt eine Kombination von allem als Fähigkeiten mit:

- Gute mathematische Fähigkeiten, insbesondere über Optimierung und Stochastik
- Sicherer Umgang mit Methoden aus der Statistik, insbesondere Regression, statistische Tests und Inferenz
- Gute Programmierkenntnisse, sicherer Umgang mit Datenbanken, Datenstrukturen, parallelem Rechnen und Big-Data-Technologien
- Problemloser Wechsel zwischen den obigen Fähigkeiten und sicherer Umgang mit Technologien, die in der Schnittstelle liegen, insbesondere dem maschinellen Lernen
- Genug Wissen über die Domäne, um die Daten zu verstehen, Fragestellungen zu definieren und zu erarbeiten, ob und wie diese Fragen mithilfe von Daten beantwortet werden können

Außerdem müssen Data Scientists teamfähig sein, um mit Domänenexpertinnen auf der einen Seite und technischen Expertinnen auf der anderen Seite zusammenarbeiten zu können. Die Domänenexpertinnen helfen den Data Scientists, die Daten, Fragestellungen und Projektziele zu verstehen. Die technischen Expertinnen helfen bei der Umsetzung von Projekten, insbesondere bei der Operationalisierung.

Nicht zuletzt sollten Data Scientists zwar den notwendigen Enthusiasmus mitbringen, um sich für die Arbeit mit Daten begeistern zu können, aber auch die notwendige Skepsis, um die Problemstellung nach wissenschaftlichen Prinzipien anzugehen. Das heißt insbesondere auch, dass man alles tun sollte, um auszuschließen, dass etwas nur aus Zufall funktioniert, und rigoros überprüfen muss, ob Modelle wirklich wie gewünscht funktionieren.

Wenn man sich dieses Fähigkeitsprofil anschaut, wird schnell klar, dass die Anzahl der Personen, die alles mitbringen, begrenzt ist. Microsoft Research hat sich daher mit der Fragestellung befasst, was Data Scientists im Arbeitsalltag leisten und welche Arten von Data Scientists es gibt [Kim et al. 2017]. Hierbei wurden acht Arten von Data Scientists bestimmt:

- *Polymath* sind die Alleskönner, die das gesamte oben beschriebene Profil erfüllen und alles von der zugrunde liegenden Mathematik bis hin zu den Big-Data-Infrastrukturen verstehen.
- *Data Evangelists* analysieren selbst Daten, verbreiten aber auch die Erkenntnisse und Modelle. Sie setzen sich insbesondere auch dafür ein, dass aus ihren Modellen Produkte entwickelt werden.

- *Data Preparers* sammeln Daten und bereiten diese für die Analyse auf.
- *Data Analyzers* analysieren Daten, die ihnen zur Verfügung gestellt werden.
- *Data Shapers* kombinieren die beiden vorigen Rollen, das heißt, sie sammeln und analysieren Daten.
- *Platform Builders* sammeln nicht nur Daten, sondern entwickeln und administrieren ganze Plattformen, die zur Datensammlung und Analyse genutzt werden können.
- *Moonlighters 50%* und *Moonlighters 20%* sind Teilzeit-Data-Scientists, die zwar auch eine Data-Science-Rolle ausfüllen, aber nur als ein Bruchteil ihrer täglichen Arbeit.
- *Insight Actors* sind die Nutzer von Analysen und Modellen.

2 Der Prozess von Data-Science-Projekten

Prozesse sind der Kern jeder Aktivität, auch wenn man sich dessen oft gar nicht bewusst ist. Menschen führen Aktivitäten durch das Anwenden von Techniken durch. Die Prozesse steuern und organisieren diese Aktivitäten und beschreiben die Techniken, die eingesetzt werden. Abbildung 2–1 zeigt die Beziehung von Menschen, Techniken und Prozessen.

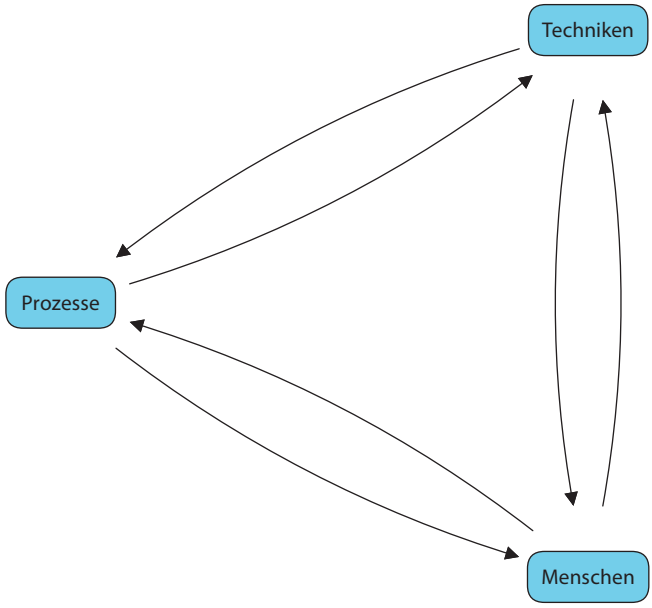

Abb. 2–1 *Beziehung von Menschen, Techniken und Prozessen*

Das Ziel eines guten Prozesses ist es, die Menschen zu unterstützen, zum Beispiel indem sichergestellt wird, dass wichtige Aktivitäten nicht vergessen werden, oder durch die Verwendung von geeigneten Werkzeugen zum Lösen von Problemen. Prozesse erreichen dies, indem sie geeignete Best Practices beschreiben. Diese Best Practices sollten auf Basis der erfolgreichen Anwendung in der Vergangenheit bestimmt werden. Hierdurch soll das Wissen und der Erfolg aus vergangenen Pro-

jekten konserviert und genutzt werden, um die Fähigkeiten der Menschen zu unterstützen und das Risiko, dass ein Projekt fehlschlägt, zu reduzieren. Dies funktioniert jedoch nur, wenn die Prozesse von Menschen auch unterstützt werden.

Wenn die Menschen den Prozess nicht akzeptieren oder nicht an seinen Nutzen glauben, erreicht man das Gegenteil und erhöht stattdessen das Risiko von Projekten. Daher sollten die Personen die notwendigen Schulungen erhalten, um die Techniken einzusetzen und ihren Nutzen zu kennen. Außerdem muss man sicherstellen, dass die Techniken auch zum Prozess passen.

Man sollte sich auch immer bewusst sein, dass es nicht *den einen Prozess* gibt, der perfekt zu jedem Projekt passt. Man sollte den Prozess daher immer an die Situation anpassen, man spricht hier auch vom *Tailoring*. Hierbei sind die zur Verfügung stehenden Techniken und der Projektkontext zu berücksichtigen, zum Beispiel die Größe und Priorität des Projekts, ob es sicherheitskritische Aspekte gibt oder ob die Mitarbeiterinnen Vorwissen aus ähnlichen Projekten mitbringen.

2.1 Der generische Data-Science-Prozess

Abbildung 2–2 zeigt den generischen Prozess für Data-Science-Projekte, der aus sechs Phasen besteht.

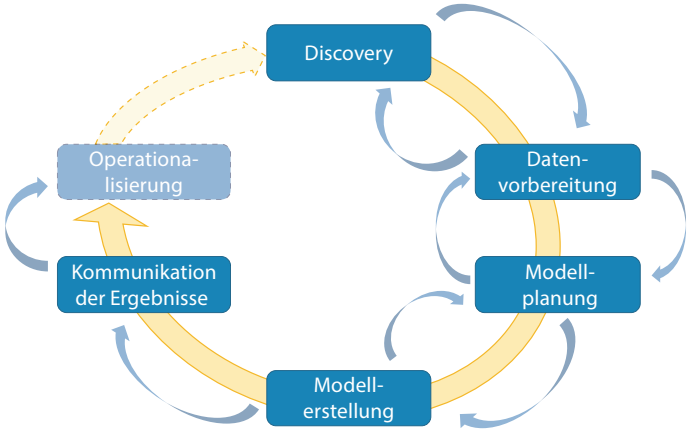

Abb. 2–2 *Überblick über den generellen Prozess von Data-Science-Projekten*

Der Prozess ist iterativ, das heißt, dass mehrere Wiederholungen aller Phasen innerhalb eines Projekts möglich sind. Innerhalb einer Iteration kann man nur zu den vorherigen Phasen zurück, solange man die Ergebnisse der Iteration noch nicht kommuniziert hat. Der Grund hierfür ist offensichtlich: Sobald man die Projektergebnisse übermittelt hat, zum Beispiel an das Management, die Kunden oder andere Forscher in Form einer Publikation, kann man diese nicht ohne Weiteres ändern. Im Folgenden betrachten wir die Projektphasen im Detail.