

Benjamin M. Abdel-Karim *Hrsg.*

Data Science

Best Practices mit Python

EBOOK INSIDE

 Springer Vieweg



Data Science

Benjamin M. Abdel-Karim
(Hrsg.)

Data Science

Best Practices mit Python

Hrsg.
Benjamin M. Abdel-Karim
Frankfurt am Main, Hessen, Deutschland

ISBN 978-3-658-33459-8 ISBN 978-3-658-33460-4 (eBook)
<https://doi.org/10.1007/978-3-658-33460-4>

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

© Der/die Herausgeber bzw. der/die Autor(en), exklusiv lizenziert durch Springer Fachmedien Wiesbaden GmbH, ein Teil von Springer Nature 2022

Das Werk einschließlich aller seiner Teile ist urheberrechtlich geschützt. Jede Verwertung, die nicht ausdrücklich vom Urheberrechtsgesetz zugelassen ist, bedarf der vorherigen Zustimmung des Verlags. Das gilt insbesondere für Vervielfältigungen, Bearbeitungen, Übersetzungen, Mikroverfilmungen und die Einspeicherung und Verarbeitung in elektronischen Systemen.

Die Wiedergabe von allgemein beschreibenden Bezeichnungen, Marken, Unternehmensnamen etc. in diesem Werk bedeutet nicht, dass diese frei durch jedermann benutzt werden dürfen. Die Berechtigung zur Benutzung unterliegt, auch ohne gesonderten Hinweis hierzu, den Regeln des Markenrechts. Die Rechte des jeweiligen Zeicheninhabers sind zu beachten.

Der Verlag, die Autoren und die Herausgeber gehen davon aus, dass die Angaben und Informationen in diesem Werk zum Zeitpunkt der Veröffentlichung vollständig und korrekt sind. Weder der Verlag noch die Autoren oder die Herausgeber übernehmen, ausdrücklich oder implizit, Gewähr für den Inhalt des Werkes, etwaige Fehler oder Äußerungen. Der Verlag bleibt im Hinblick auf geografische Zuordnungen und Gebietsbezeichnungen in veröffentlichten Karten und Institutionsadressen neutral.

Planung: Petra Steinmueller

Springer Vieweg ist ein Imprint der eingetragenen Gesellschaft Springer Fachmedien Wiesbaden GmbH und ist ein Teil von Springer Nature.

Die Anschrift der Gesellschaft ist: Abraham-Lincoln-Str. 46, 65189 Wiesbaden, Germany

Vorwort

Im Rahmen meines Studiums der Wirtschaftsinformatik und innerhalb meiner Zeit als wissenschaftlicher Mitarbeiter und Forscher habe ich zahlreiche Probleme mithilfe der Programmierung lösen können, wie zum Beispiel die professionelle Datenanalyse oder das Implementieren von künstlichen Intelligenzen. Vor diesem Hintergrund kann ich gut nachvollziehen, welchen Herausforderungen Einsteiger¹ im Bereich der Programmierung gegenüberstehen. Ausgehend von den zahlreichen Lehrbüchern und Vorlesungen habe ich mir mit diesem Buch das persönliche Ziel gesetzt, ein Werk zu schaffen, das mit simplen Beispielen und einfachen Erklärungen den Zugang zur Programmierung, insbesondere im Bereich der Datenanalyse, ermöglicht. Hierbei möchte ich dem interessierten Leser ein Werk an die Hand geben, das meine wertvollsten Quellcodeartefakte beinhaltet, um somit einen Ansatzpunkt zu schaffen, die eigenen (alltäglichen) Datenanalyseherausforderungen zu lösen.

Ich habe dieses Buch aus der Motivation heraus geschaffen, eines der ersten deutschsprachigen Nachschlagewerke zu entwickeln, in dem relativ simple Quellcodebeispiele enthalten sind, um Lösungsansätze für die (wiederkehrenden) Programmierprobleme in der Datenanalyse weiterzugeben. Hinzu kommt, dass ich selbst wiederholt Lösungswege für ähnliche Probleme erarbeitet habe, die ich bereits in der Vergangenheit entwickelt hatte. Zweifellos gehört das Nachschlagen von Lösungsansätzen in Büchern oder im Internet zur normalen Arbeit eines Programmierers. Allerdings ist diese Suche in der Regel ein unstrukturierter und somit meist zeitaufwendiger Prozess. Entsprechend ist dieses Werk auch für mich eine Zusammenfassung wichtiger Lösungswege für die Grundprobleme des Programmierens.

Unabhängig davon, ob Sie das Buch als Studierender, Mitarbeiter, Gründer eines Start-ups oder als Unternehmer lesen, hoffe ich, dass Ihnen dieses Nachschlagewerk ein wertvoller Helfer für die ersten Anfänge sein wird. Ich gehe davon aus, dass jede Person

¹Aus Gründen der besseren Lesbarkeit wird im Folgenden in der Regel die männliche Form verwendet. Allerdings betonen alle Autoren, dass gleichermaßen alle Geschlechterformen angesprochen sind.

die Grundlagen der Datenanalyse mithilfe moderner Programmiersprachen erlernen kann. Der Erfolg der Programmiersprache Python zeigt, dass die Programmierung kein aufwendiges Unterfangen sein muss. In den letzten Jahren ist Python in Forschung und Praxis zu einer der beliebtesten Programmiersprachen avanciert. Ein zentraler Vorteil von Python ist seine Verständlichkeit, besonderes für Anfänger. Hinzu kommt, dass Python über eine große Entwicklergemeinschaft verfügt. Dies führt zur Bereitstellung zahlreicher und kostenfreier Erweiterungen. Durch meine Erfahrungen als Wissenschaftler im Bereich der künstlichen Intelligenz, Universitätsdozent und Redner, weiß ich, dass viele Menschen eine gewisse Einstiegshürde mit Blick auf die Programmierung verspüren, die sich aber schnell nach den ersten Erfolgserlebnissen legt. Genau jenes Erfolgserlebnis kann zu einem regelrechten Motivator werden, um immer tiefer in die Programmierung einzusteigen.

Ausgehend von meinen Erfahrungen, lernt man am besten durch einfache Beispiele. Das Lösen solcher Aufgaben ist aus meiner Perspektive besonderes wichtig, um ein Grundverständnis zu entwickeln. Vor diesem Hintergrund ist eine zentrale Prämisse des Buchs, dass alle Lösungsansätze so einfach wie möglich gehalten sind, was Sie als Leser befähigen soll, Ansätze für die Lösung Ihrer eigenen Probleme zu entwickeln. Sie benötigen für dieses Buch keine tiefgreifenden Vorkenntnisse in den Bereichen Programmierung oder Datenanalyse. Sofern Sie einen Computer bedienen können und in der Lage sind, Software zu installieren, können Sie mit dem Lesen des Buchs beginnen.

Dieses Werk grenzt sich von anderen Büchern ab, da es ein Nachschlagewerk für Quellcodeansätze im täglichen Data Science sein soll. Aus meiner Erfahrung sind viele einführende Lehrbücher in Python als universelle Bücher konzipiert, sodass sie viele Teilbereiche abdecken und ausführliche Erklärungen bieten. Damit sind diese Lehrbücher als Lernwerke im Sinn eines ausführlichen Studiums konzipiert, was dazu führt, dass der interessierte Leser einem solchen Lehrbuch ausreichend Zeit widmen muss. Im Gegensatz dazu soll mein Werk praktisch anwendbar sein und beim aktiven Quellcode schreiben helfen, sodass es einfach aus dem Regal genommen werden kann, um eine schnelle Hilfestellung bei aufkommenden Fragen zu liefern. Alternativ kann das Buch als Inspirationsgeber für die Umsetzung eigenständiger Projekte dienen. Mit diesem Buch möchte ich den Leser dazu befähigen, den Data-Science-Herausforderungen zu begegnen. Meine Co-Autoren und ich haben in diesem Werk möglichst auf Fachvokabular zugunsten von Alltagssprache verzichtet, um die Verständlichkeit zu erhöhen. Notwendige Fachbegriffe werden an geeigneten Stellen ausführlich erklärt. Durch das Stichwortverzeichnis lassen sich außerdem entsprechende Begrifflichkeiten schnell finden und das Lexikon bietet eine Übersicht mit Kurzerklärungen des Data-Science-Sprachgebrauchs.

An dieser Stelle möchte ich mich bei allen Personen bedanken, die sich unmittelbar oder indirekt an diesem Buchprojekt beteiligt haben. Mein Dank gilt besonders meinen Co-Autoren, die durch ihre unterschiedlichen Ideen und ihre Mitarbeit dieses Buch maßgeblich vorangebracht haben. Ich verzichte aus Platzgründen bewusst auf

die einzelne namentliche Nennung, möchte mich aber neben den Co-Autoren auch bei meinen Kollegen, Studenten und Familienmitgliedern bedanken.

Programmierung ist für mich vergleichbar mit dem Erstellen eines eigenen Kunstwerks. Jeder eigene Programmcode trägt die Handschrift des Schöpfers. Daher freue ich mich besonders darüber, dass der interessierte Leser in diesem Buch die Gelegenheit erhält, unterschiedliche Stile der Programmierung kennenzulernen. Daher wäre es schön, wenn Sie als Leser die unterschiedlichen Ideen und Quellcodes so betrachten, als würden Sie durch eine Galerie schlendern.

Sie finden alle Inhalte zu diesem Buch auf der Webseite des Buchs: <https://github.com/BenjaminMAK/DataScienceBestPractices>

Über Anregungen und Kritik zu diesem Buch, auch über Hinweise zu Entwicklungsmöglichkeiten oder Wünsche für zukünftige Auflagen, freue ich mich sehr. Gern können Sie mich über die Webseite zum Buch kontaktieren.

Nun wünsche ich allen Lesern einen interessanten Einstieg in die Programmierung und viel Freude beim Lesen und Ausprobieren.

Hochachtungsvoll im Namen aller Co-Autoren Ihr:
Benjamin M. Abdel-Karim

Herausgeber- und Autorenverzeichnis

Über den Herausgeber



Dr. Benjamin M. Abdel-Karim ist seit März 2018 als wissenschaftlicher Mitarbeiter an der Professur für Wirtschaftsinformatik und Informationsmanagement von Prof. Dr. Oliver Hinz tätig. Er hat Wirtschaftsinformatik an der Technischen Universität in Darmstadt (M.Sc.) studiert. Im Masterstudium lagen seine Schwerpunkte in den Bereichen Data Knowledge Engineering/Artificial Intelligence und Finanzierung. Die Masterarbeit befasst sich mit der Modellierung spezieller neuronaler Netze zur Abbildung komplexer Finanzmarktstrukturen zwecks der Prognose. Sein Bachelorstudium in Wirtschaftsinformatik (B.Sc.) absolvierte er an der Universität Bremen. Der Schwerpunkt seines Bachelorstudiums war Computational Finance. Praktische Erfahrungen konnte Benjamin M. Abdel-Karim unter anderem durch seine Bankausbildung und Mitarbeit in zahlreichen Banken, wie beispielsweise der Deutsche Asset & Wealth Management, sammeln. Zudem kommen zahlreiche Auftritte als Keynote Speaker im Bereich maschinelles Lernen und Data Science dazu.

Autorenverzeichnis



Dr. Kevin Bauer trat 2013 in das Doktorandenprogramm der GSEFM ein, nachdem er seinen Bachelor-Abschluss in Wirtschaftswissenschaften an der Goethe-Universität erworben hatte. Im Jahr 2017 schloss er seinen ersten Master-Abschluss in Law and Quantitative Economics ab. Im Jahr 2018 schloss er seine Promotion mit dem akademischen Titel Dr. rer. pol. mit einem summa cum laude ab. Seine Doktorarbeit *Ön the Economic Significance of Social Groups: New Evidence on Self-Selection, Social Identity and Social Preferences* konzentriert sich auf die empirische Analyse von Wirtschaftsbeziehungen. Während seiner Promotion begann er einen zweiten Masterstudien-gang in Wirtschaftsinformatik an der Goethe-Universität mit den Schwerpunkten maschinelles Lernen und Künstliche Intelligenz. Nach seiner Promotion trat Kevin Bauer als KI-Spezialist dem TechQuartier bei, wo er weiterhin als externer Berater für KI-bezogene Themen tätig ist. Seit 2020 ist Kevin Bauer als Postdoc-Forscher am Leibniz SAFE in der Forschungsgruppe Digitalisierung in der Finanzindustrie (Leitung: Prof. Hinz) beschäftigt.



Daniel Franzmann erlangte seinen Bachelor in Wirtschaftswissenschaften mit dem Schwerpunkt in Finanzen und Accounting an der Goethe Universität Frankfurt. Nach einem Auslandssemester an der York University in Großbritannien und erster Berufserfahrung bei PwC und Senacor Technologies AG absolvierte er im Jahr 2016 seinen Master in Wirtschaftsinformatik mit dem Fokus auf Data Science an der Goethe Universität Frankfurt. Von 2016 bis 2020 war Daniel Franzmann am Lehrstuhl für Information Systems Engineering der Goethe Universität Frankfurt als Data Scientist angestellt und promovierte über das Thema Software Updates. Seit Oktober 2020 arbeitet Daniel Franzmann im Data Analytics Team der Deutschen Bank. Privat interessiert er sich fürs Laufen, Kochen und surft gerne auf r/dataisbeautiful.



Hendrik Jöntgen absolvierte seinen Bachelor und Master in Wirtschaftsinformatik an der Technischen Universität Darmstadt. Im Masterstudium lag sein Schwerpunkt in dem Bereich Informationsvisualisierung und durch langjährige Arbeitserfahrung als studentische Hilfskraft an verschiedenen Lehrstühlen konnte er bereits früh Erfahrungen mit dem Erheben und Auswerten von Social Media Daten sammeln. In seiner Masterarbeit untersuchte er Vertrauensbildung innerhalb der Sharing Economy und die Effekte von unterschiedlichen Profil-Charakteristiken auf die Wahrnehmung der Plattform-Nutzer. Seit März 2019 ist Hendrik Jöntgen als wissenschaftlicher Mitarbeiter am Lehrstuhl für Wirtschaftsinformatik und Informationsmanagement von Prof. Dr. Oliver Hinz tätig und erforscht Vertrauen und soziales Kapital auf Social Media Plattformen. Seine Freizeit verbringt er gerne mit Brettspielen, am Schlagzeug oder auf Waldwegen.



Katharina Keller studierte Wirtschaftsinformatik an der Goethe-Universität Frankfurt (M.Sc.). Ihr Bachelorstudium absolvierte sie in Wirtschaftswissenschaften mit Schwerpunkt Marketing/Management. Während ihres Bachelorstudiums verbrachte sie ein Auslandssemester an der Católica Lisbon School of Business and Economics. Im Juni 2017 schloss sie ihr Studium mit dem akademischen Grad Master of Science ab. Ihre Masterthesis schrieb sie zum Thema „Technology Choice Behavior in Post-Adoption IS Usage“. Seit Juli 2017 ist Katharina Keller als wissenschaftliche Mitarbeiterin an der Professur von Prof. Dr. Oliver Hinz tätig. Zuerst am Lehrstuhl für Wirtschaftsinformatik | Electronic Markets an der TU Darmstadt und seit September 2017 an der Professur für Wirtschaftsinformatik und Informationsmanagement an der Goethe-Universität. Sie ist Teil des Sonderforschungsbereichs 1053 MAKI (Multi-Mechanismen-Adaption für das künftige Internet), der sich mit Fragen zum Thema künftiges Internet und dessen Auswirkung auf das Kommunikationsverhalten befasst. Außerdem betrachtet der SFB 1053 die Mechanismen in Kommunikationssystemen sowie die daraus resultierenden

Anforderungen an die Infrastruktur. Im Speziellen ist Katharina Keller im Teilprojekt B3 beteiligt, das die Transitionen in Kommunikationssystemen aus ökonomischer Perspektive untersucht und dezentrale Analyse- und Planungslösungen für autonome Endgeräte erstellt. Ziel ist eine effiziente und realistische Anwendbarkeit proaktiver Transitionen für koexistente Multi-Mechanismen in drahtlosen Netzen software-definierter autonomer Knoten.



Marcel Zeuch studierte Mathematik an der Technischen Universität in Darmstadt (M.Sc.). Sein erstes Mastersemester absolvierte er dabei 2016 als Erasmus-Stipendiat an der Politecnico di Milano in Mailand, Italien. Seither sammelte er Erfahrungen in einer Managementberatung, einer Top Tier Investmentbank sowie seit seinem Abschluss im Jahr 2018 in einer datengetriebenen Boutiqueberatung im Capital Markets Umfeld. In seiner aktuellen Position im Bereich Corporates & Markets bei einer deutsch-französischen Investmentbank beschäftigt er sich mit Themen der internen Prozessautomatisierung und -weiterentwicklung. In seiner Freizeit verfasste er bereits mehrfach Artikel für einen deutschen Finanzblog und interessiert sich für Themen aus den Bereichen Capital Markets, Data Science und Entrepreneurship.

Inhaltsverzeichnis

Teil I Grundlagen der Programmierung

1	Einleitung	3
	Benjamin M. Abdel-Karim	
1.1	Gegenstandsbereich dieses Buchs	3
1.2	Aufbau und Zielsetzung	4
1.3	Warum Python?	5
2	Python: Installation und Einstieg	7
	Benjamin M. Abdel-Karim	
2.1	Python	7
2.2	Installation	8
2.3	Entwicklungsumgebung	9
2.4	Formalitäten für ein geeignetes Skript	13
3	Primitive Datentypen	17
	Benjamin M. Abdel-Karim	
3.1	Grundlagen zum Verständnis von Programmiersprachen	18
3.2	Ganze Zahlen	19
3.3	Gleitkommazahlen	20
3.4	Zeichen und Zeichenketten in der Programmierung	21
3.5	Boolean	21
4	Datenstruktur	25
	Benjamin M. Abdel-Karim	
4.1	Listen in Python	25
4.1.1	Liste als Stapelspeicher (Stack)	28
4.1.2	Listen als Warteschlange (Queue)	29
4.2	Dictionaries in Python	30
4.3	Mengen (Sets) in Python	32

5	Kontrollstrukturen	35
	Benjamin M. Abdel-Karim	
5.1	Verzweigungen	36
5.1.1	Die If-Else-Bedingungen	36
5.1.2	Verschachtelte If-else-Bedingungen	39
5.2	Schleifen	41
5.2.1	Die For-Schleife	41
5.2.2	Die While-Schleife	44
5.3	Try-except-Bedingung	45
6	Funktionen	49
	Benjamin M. Abdel-Karim	
6.1	Built-in-Funktionen	49
6.2	Funktionen	50
6.3	Bibliotheken (Module) in Python	53
Teil II Data Science		
7	Data Science	57
	Benjamin M. Abdel-Karim	
7.1	Einordnung Data Science	57
7.2	Data-Science-Prozess	60
7.3	Data-Science-Projekte für dieses Buch	61
8	Data Science und Maschinelles Lernen	63
	Benjamin M. Abdel-Karim	
8.1	Definitionen des maschinellen Lernens	63
8.2	Herausforderungen des Maschinellen Lernens im Kontext von Data Science	66
Teil III Produktanalyse		
9	Anwendungsbeispiel: Meine besten Videospiele	73
	Benjamin M. Abdel-Karim	
9.1	Videospiel: Datensatz und Fragestellung	74
9.2	Videospiel: Preprocessing	75
9.3	Videospiel: Explore the Data	80
9.4	Videospiel: Model the Data (Regression)	88
9.4.1	Theoretische Modellgrundlagen	88
9.4.2	Implementierung des Modells	90
9.5	Videospiel: Interpretation der Ergebnisse	92

10 Anwendungsbeispiel: Conjoint-Analyse – Mehr als die Summe seiner Teile	93
Katharina Keller	
10.1 Conjoint-Analyse: Einführung in die Methodik	94
10.2 Conjoint-Analyse: Datensatz und Fragestellung	96
10.3 Conjoint-Analyse: Preprocessing	98
10.4 Conjoint-Analyse: Explore the Data	101
10.5 Conjoint-Analyse: Model the Data	103
10.6 Conjoint-Analyse: Interpretation der Ergebnisse	106
Teil IV Kunden- und soziale Medienanalyse	
11 Anwendungsbeispiel: Game of Social Networks	113
Benjamin M. Abdel-Karim	
11.1 Soziales Netzwerk: Datensatz und Fragestellung	114
11.2 Soziales Netzwerk: Pre-processing	115
11.3 Soziales Netzwerk: Explore the Data	117
11.4 Soziales Netzwerk: Model the Data (Logistische Regression)	126
11.4.1 Theoretische Grundlage des Modells	126
11.4.2 Implementierung des Modells	128
11.5 Soziales Netzwerk: Interpretation der Ergebnisse	129
12 Erhebung und Auswertung von Social-Media-Daten	131
Hendrik Jöntgen	
12.1 Social Media: Datensatz und Fragestellung	132
12.2 Social Media: Pre-processing	135
12.3 Social Media: Explore the Data	138
12.4 Social Media: Model the Data	144
12.5 Social Media: Interpretation der Ergebnisse	148
13 Anwendungsbeispiel: Cloud Web Services	151
Daniel Franzmann	
13.1 Cloud Web Services: Fragestellung	152
13.2 Cloud Web Services: Pre-processing	152
13.3 Cloud Web Services: Explore the Data and More	161
13.4 Cloud Web Services: Zusammenfassung	169

Teil V Mitarbeiteranalyse

14 Anwendungsbeispiel: Mitarbeiterabwanderung	173
Kevin Bauer	
14.1 Mitarbeiterabwanderung: Vorbereitung und Definition des Problems. . .	174
14.2 Mitarbeiterabwanderung: Auf-/Vorbereitung der Daten	177
14.2.1 Mitarbeiterabwanderung: Sampling und Erzeugung von Trainings- und Testdaten	192
14.2.2 Explorative Analysen auf Trainingsdaten	198
14.2.3 Auswahl und Training von ML Modellen	203
15 Anwendungsbeispiel: Get Your Things Done – Modernes Zeitmanagement	225
Benjamin M. Abdel-Karim	
15.1 Zeitmanagement: Datensatz und Fragestellung	225
15.2 Zeitmanagement: Pre-processing	228
15.3 Zeitmanagement: Explore the Data	232
15.4 Zeitmanagement: Model the data.	240
15.4.1 Theoretische Modellgrundlagen	244
15.4.2 Implementierung des Modells	247
15.5 Zeitmanagement: Interpretation der Ergebnisse	248

Teil VI Finanzanalyse

16 Anwendungsbeispiel: Portfolioanalyse	253
Marcel Zeuch	
16.1 Portfolioanalyse: Datensatz und Fragestellung	254
16.2 Bereitstellung einer Datenbank	255
16.3 Portfolioanalyse: Pre-processing	258
16.4 Portfolioanalyse: Explore the Data	262
16.5 Portfolioanalyse: Model the Data	266
16.5.1 Datenaggregation und Speicherung in der Datenbank.	267
16.5.2 Verarbeitung der Daten und Implementierung der Benutzeroberfläche.	273
16.6 Portfolioanalyse: Interpretation der Ergebnisse	293

Thematisches Lexikon	295
-----------------------------------	-----

Literatur	301
------------------------	-----

Stichwortverzeichnis	307
-----------------------------------	-----

Abbildungsverzeichnis

Abb. 2.1	Das Python-Installationsfenster	9
Abb. 2.2	Willkommenfenster PyCharm	10
Abb. 2.3	Projektkonfiguration unter PyCharm	11
Abb. 2.4	Das erste Skript im Projektordner	12
Abb. 2.5	Die IDE und Hello World.	13
Abb. 3.1	Ausgehend von den übergeordneten Klassen der primitiven Datentypen zeigt die Abbildung eine Auswahl der wichtigsten Datentypen, die in Python zum Einsatz kommen können	18
Abb. 4.1	Ausgehend von der übergeordneten Klasse der Datenstruktur wird eine Auswahl der wichtigsten Datenstrukturen gezeigt, die in Python zum Einsatz kommen können	26
Abb. 5.1	Die Kontrollstrukturen in der Übersicht. Die Abbildung visualisiert ausgewählte Kontrollstrukturen	36
Abb. 5.2	Die If-else-Bedingung prüft, ob eine Bedingung zutrifft. Sofern die deklarierte Bedingung erfüllt ist, wird der nächste Programmabschnitt ausgeführt. Trifft die Bedingung nicht zu, wird ein anderer Programmabschnitt ausgeführt	37
Abb. 5.3	Die verschachtelten If-else-Bedingungen. Zunächst erfolgt eine If-Bedingungsprüfung. Ist diese Bedingung erfüllt, erfolgt die Ausführung des nächsten Programmabschnitts. Ist diese Bedingung nicht erfüllt, wird im Else-Zweig eine weitere If-Bedingungsprüfung ausgeführt	40
Abb. 5.4	Die For-Schleife. Der Schleifenkopf legt durch die Deklaration der Bedingung die Lauflänge fest. Solange die Bedingung Gültigkeit besitzt, wird der Quellcode im Schleifenrumpf ausgeführt. Sobald die maximale Lauflänge erreicht und damit die Bedingung nicht mehr erfüllt ist, wird die Schleife ihren Dienst einstellen und der nächste Quellcode außerhalb der For-Schleife ausgeführt	42

Abb. 5.5	Die While-Schleife. Der Schleifenkopf der While-Schleife enthält eine Bedingung. Solange diese erfüllt ist, wird die Schleife ihren Betrieb fortsetzen. Diese Bedingung wird bei jedem Schleifendurchlauf geprüft. Erst wenn die Bedingung ihre Gültigkeit verloren hat, stellt die Schleife ihren Betrieb ein, was dazu führt, dass der nächste Codeabschnitt außerhalb der Schleife ausgeführt wird.	44
Abb. 5.6	Die Try-except-Kontrollstruktur. Zunächst wird Python versuchen, den Try-Block auszuführen. Dieser umfasst das Codesegment, das ausgeführt werden soll. Der Except-Block greift dann, wenn der Try-Block zu einem Ausnahmefall führt und der Code eigentlich abstürzen würde.	46
Abb. 7.1	Data Science – ein Versuch der Einordnung	59
Abb. 7.2	Data-Science-Prozess	61
Abb. 8.1	Definitionsbaum	64
Abb. 8.2	Maschinelles Lernen – Taxonomie.	65
Abb. 9.1	Der Variableninspektor in PyCharm zur Visualisierung der Wertebelegung eines DataFrame	78
Abb. 9.2	Der Pairplot für die Videospiele.	83
Abb. 9.3	Der Barplot für die Videospieleinnahmen im Zeitverlauf	85
Abb. 9.4	Der horizontale Barplot für die Top-Ten-Videospiel-Publisher gemessen an den Einnahmen	87
Abb. 10.1	Beispielhaftes Choice Set.	98
Abb. 10.2	Plot Bedeutungsgewichte	109
Abb. 11.1	Das soziale Game-of-Thrones-Netzwerk in einer ersten Übersicht.	119
Abb. 11.2	Game-of-Thrones-Netzwerkstruktur auf Basis des Degree	126
Abb. 12.1	Zeitverlauf der Tweets pro Stunde	139
Abb. 12.2	Zeitverlauf der Tweets pro Minute	140
Abb. 12.3	Zeitverlauf des Sentiment.	141
Abb. 12.4	Zeitverlauf des Sentiment mit Einbettung der Timestamps	142
Abb. 12.5	Verteilung der Tweets auf die unterschiedlichen Videospiele	144
Abb. 12.6	Sentiment-Boxplots der einzelnen Videospiele	145
Abb. 12.7	Grafische Überprüfung der Normalverteilung des Sentiment	147
Abb. 13.1	Der FileZilla Server Manager in der Übersicht	159
Abb. 13.2	Erfolgreicher Verbindungsaufbau zur EC2-Instanz	160
Abb. 13.3	Fake- (rot) vs. Non-fake-Kommentar-Sentiments (blau)	164
Abb. 13.4	Non-fake vs. fake word clouds	166
Abb. 13.5	Das neu erstellte S3-Bucket	167
Abb. 13.6	Erfolgreiche Visualisierung auf dem S3-Bucket	169
Abb. 14.1	Der Pairplot des Trainsets.	201

Abb. 14.2	(a) Der Relplot auf Basis der Daten. (b) Rel plot: Label = 0. (c) Rel plot: Label = 1. (d) Catplot der Variante ‚bar‘. (e) Catplot der Variante ‚box‘.	203
Abb. 14.3	(a) Konfusionsmatrix Trainingsdaten. (b) Konfusionsmatrix Testdaten.	208
Abb. 14.4	Optimaler Hyperparameter des KNN Modells.	212
Abb. 14.5	(a) Konfusionsmatrix Trainingsdaten. (b) Konfusionsmatrix Testdaten.	213
Abb. 14.6	(a) Konfusionsmatrix Trainingsdaten. (b) Konfusionsmatrix Testdaten.	216
Abb. 14.7	(a) Konfusionsmatrix Trainingsdaten. (b) Konfusionsmatrix Trainingsdaten	221
Abb. 14.8	Feature importances	222
Abb. 15.1	Grundfunktionalität des Time Tracker Wearable	226
Abb. 15.2	Zeitliche Verteilung der Aktivitäten in Form von Boxplots	235
Abb. 15.3	Zeitliche Verteilung der Aktivitäten in Form eines Ringdiagramms . . .	238
Abb. 15.4	Gegenüberstellung der Daten aus der Datentransformation von Minuten zu Stunden.	243
Abb. 15.5	Der Informationsverarbeitungsprozess eines Neurons.	246
Abb. 16.1	Implementierung des vorgestellten Portfolioanalysetools	255
Abb. 16.2	Darstellung der zu implementierenden Benutzeroberfläche	274
Abb. 16.3	Darstellung der unterschiedlichen HTML-Komponenten der zu implementierenden Benutzeroberfläche.	286

Tabellenverzeichnis

Tab. 3.1	Übersicht nützlicher arithmetischer Operatoren mit Integer. $iX =$ Integer-Variable; $iY =$ weitere Integer-Variable.	20
Tab. 3.2	Übersicht nützlicher arithmetischer Operatoren mit Float. $dX =$ Float-Variable; $dY =$ weitere Float-Variable	20
Tab. 3.3	Übersicht der Vergleichsoperatoren.	23
Tab. 6.1	Auswahl einiger Built-in-Funktionen in Python	50
Tab. 9.1	Übersicht nützlicher Funktionen des pandas-Moduls für die Videospieldanalyse im ersten Pre-Processing	80
Tab. 9.2	Übersicht nützlicher Funktionen für die Datenexploration	89
Tab. 9.3	Übersicht nützlicher Funktionen für die Modellkonzeption	91
Tab. 10.1	Übersicht Attribute und Attributlevel	97
Tab. 11.1	Top-Ten-Charaktere in „Game of Thrones“ basierend auf ihrem Degree, Betweenness Centrality und der Degree Centrality. Betweenness steht für Betweenness Centrality und Centrality steht für Degree Centrality	124
Tab. 11.2	Übersicht nützlicher Funktionen für die Datenexploration	127
Tab. 11.3	Übersicht nützlicher Funktionen für die Modellkonzeption	130
Tab. 15.1	Abrechenbare Stunden in Euro für das Arbeitsjahr.	239
Tab. 16.1	Auszug der Eingabeparameter des Musterportfolios	255

Listings

Listing 2.1	Professionelle Quellcodedokumentation	15
Listing 3.1	Deklarierung eines	21
Listing 3.2	Operationen mit Boolean in Python	22
Listing 3.3	Verknüpfungsoperatoren mit Logicals in Python	22
Listing 4.1	Eine Liste in Python anlegen	26
Listing 4.2	Ein Element in eine Liste aufnehmen	27
Listing 4.3	Zugriff auf das erste Listenelement	27
Listing 4.4	Liste in Liste	28
Listing 4.5	Liste als Stapelspeicher	29
Listing 4.6	Liste als Warteschlange	30
Listing 4.7	Dictionary	31
Listing 4.8	Mengen	32
Listing 5.1	If-Else in Python	37
Listing 5.2	If-Else mit Konjunktion in Python.	38
Listing 5.3	If-Else mit oder Operation in Python.	39
Listing 5.4	Verschachtelte If-else-Bedingung mit Oder-Operation in Python.	39
Listing 5.5	For-Schleife in Python.	41
Listing 5.6	For-Schleifen Ergebnis	42
Listing 5.7	For-Schleife in Python	43
Listing 5.8	For-Schleifen-Ergebnis für Listeniteration	43
Listing 5.9	While-Schleife in Python.	44
Listing 5.10	While-Schleifen Ergebnis	45
Listing 5.11	Try-exception-Kontrollstruktur in Python.	46
Listing 5.12	Try-Exception mit genauer Exception Spezifikation in Python	47
Listing 6.1	Eine eigene Funktion in Python.	52
Listing 6.2	Nutzung des random-Moduls in Python	54
Listing 9.1	Import des Moduls Pandas.	76
Listing 9.2	Import einer .CSV-Datei als pandas DataFrame	77
Listing 9.3	Aufruf der df.head()-Funktion	77

Listing 9.4	Konsolenausgabe: <code>df.head()</code>	77
Listing 9.5	Aufruf der <code>df.info()</code> -Funktion	78
Listing 9.6	Konsolenausgabe: <code>df.info()</code>	79
Listing 9.7	Aufruf der Funktion <code>df.describe()</code>	80
Listing 9.8	Konsolenausgabe: <code>df.describe()</code>	80
Listing 9.9	Erstellung eines Pairplot für die Videospiele	82
Listing 9.10	Erstellung eines Barplot für die Videospiele im zeitlichen Verlauf	83
Listing 9.11	Erstellung eines Barplot für die Top-Ten-Publisher	86
Listing 9.12	Top-Ten-Videospiele	87
Listing 9.13	Konsolenausgabe: Top-Ten-Videospiele	88
Listing 9.14	Konsolenausgabe: Lineare Regression	91
Listing 10.1	Import der benötigten Module	98
Listing 10.2	Einlesen der Daten	99
Listing 10.3	Zusammenführen der Daten	99
Listing 10.4	Vorbereitung der Daten	100
Listing 10.5	Endogene und exogene Variable trennen	101
Listing 10.6	Dummycodierung	101
Listing 10.7	Effektcodierung	102
Listing 10.8	Schätzung der Parameterwerte	104
Listing 10.9	Speichern der Ergebnisse	104
Listing 10.10	Iteration	105
Listing 10.11	Berechnung der fehlenden Werte	105
Listing 10.12	Berechnung der Spannweiten	106
Listing 10.13	Berechnung der Bedeutungsgewichte	107
Listing 10.14	Konsolenausgabe: Die Bedeutungsgewichte	107
Listing 10.15	Plotten der Ergebnisse	107
Listing 11.1	Import der Module für die Soziale Netzwerk Analyse	115
Listing 11.2	Import der Daten für die Soziale Netzwerk Analyse	115
Listing 11.3	Konsolenausgabe: <code>len(df.columns)</code>	116
Listing 11.4	Konsolenausgabe: <code>df.columns.to_list()</code>	116
Listing 11.5	Konsolenausgabe: <code>df.info()</code>	116
Listing 11.6	Konsolenausgabe: <code>df.isnull().sum()</code>	117
Listing 11.7	Nutzung von <code>df.describe()</code>	117
Listing 11.8	Konsolenausgabe: <code>df.describe()</code>	118
Listing 11.9	Nutzung von <code>nx.from_pandas_edgelist</code>	118
Listing 11.10	Abbildung eines Graphen durch die eigene Funktion <code>fGeneratePlot</code>	119
Listing 11.11	Durchführung einer ersten Sozialen Netzwerk Analyse	122
Listing 11.12	Verteilungswahrscheinlichkeit der Degrees auf Basis des Logarithmus	124
Listing 11.13	Logistische Regressionsanalyse	128

Listing 11.14	Konsolenausgabe: Logistische Regression	129
Listing 12.1	Initialisierung der Twitter API	133
Listing 12.2	Abfrage der Twitter-Query	134
Listing 12.3	Umwandelung der Query-Ergebnisse in ein DataFrame	134
Listing 12.4	Speicherung der Twitter-Daten in eine MongoDB	135
Listing 12.5	Laden der Twitter-Daten von einer MongoDB	135
Listing 12.6	Aufbereitung der Tweet-Zeitdaten	135
Listing 12.7	Aufbereitung der Tweet-Zeitdaten	136
Listing 12.8	Definition von Regex	136
Listing 12.9	Filtern der Tweet-Texte	137
Listing 12.10	Berechnung der Tweet-Sentiments	138
Listing 12.11	Zeitverlauf der Tweets pro Stunde	138
Listing 12.12	Zeitverlauf der Tweets pro Minute	139
Listing 12.13	Zeitverlauf des Sentiment	140
Listing 12.14	Einbettung der Timestamps	141
Listing 12.15	Zuteilung der Tweets	142
Listing 12.16	Erstellung eines Bar-Plots	143
Listing 12.17	Erstellung von Boxplots für jedes Videospiel	145
Listing 12.18	Durchführung einer ANOVA zum Testen auf Unterschieden im Sentiment zwischen Videospielen	146
Listing 12.19	Shapiro-Wilk-Test zum Überprüfen der Normalverteilung des Sentiment für jedes Videospiel	146
Listing 12.20	Plot der Residuen	146
Listing 12.21	Durchführung eines Kruskal-Willis Tests auf Unterschiede im Sentiment zwischen Videospielen	148
Listing 13.1	Benötigte Libraries und Initialisierung des API-Schlüssels	156
Listing 13.2	Suche nach dem Begriff „corona“ und Speichern der Video-Links	156
Listing 13.3	Initialisierung der Ergebnislisten	157
Listing 13.4	YouTube-Kommentar Web Crawler	157
Listing 13.5	Speichern der Ergebnisse	158
Listing 13.6	Benötigte Libraries	162
Listing 13.7	Import der CSV-Datei und Filtern nach englischen Kommentaren	162
Listing 13.8	Generierung der Sentiments und Transformation des Datum-/Zeit-Formats	163
Listing 13.9	Aufteilung in zwei Gruppen und Aggregation der Durchschnitts-Sentiment-Werte	163
Listing 13.10	Visualisierung der Resultate	164
Listing 13.11	Verknüpfung der täglichen Kommentare durch Gruppierung nach Datum	165
Listing 13.12	Visualisierung der beiden Gruppen in täglichen Word Clouds	165

Listing 13.13	index.html	168
Listing 14.1	Import pandas as pd	174
Listing 14.2	Import numpy as np	175
Listing 14.3	Import matplotlib	175
Listing 14.4	Import seaborn	175
Listing 14.5	Import Sklearn	175
Listing 14.6	Import der Daten	178
Listing 14.7	DataFrame ansicht	178
Listing 14.8	Konsolenausgabe	179
Listing 14.9	DataFrame dtypes	179
Listing 14.10	Konsolenausgabe	179
Listing 14.11	DataFrame columns	179
Listing 14.12	Konsolenausgabe: Datentypen im DataFrame	180
Listing 14.13	Teil des DataFrame	180
Listing 14.14	Teil des DataFrame	180
Listing 14.15	Slice-Indexierer	181
Listing 14.16	Slice-Indexierer mit Bedingung	181
Listing 14.17	Konsolenausgabe	182
Listing 14.18	DataFrame weitere Datenmanipulationen	182
Listing 14.19	DataFrame auf null nan Werte überprüfen	183
Listing 14.20	Konsolenausgabe	183
Listing 14.21	Optionen zur Handhabung fehlender Werte	184
Listing 14.22	Verwendung von value_counts	185
Listing 14.23	Konsolenausgabe	185
Listing 14.24	Verwendung des one-hot-encoders	186
Listing 14.25	Schleife für Dummy Variablen	186
Listing 14.26	Matrix to DataFrame	187
Listing 14.27	Konsolenausgabe	187
Listing 14.28	Matrix to DataFrame	187
Listing 14.29	Implementierung einer Methode zum automatisierten One-Hot-Encoding	188
Listing 14.30	Reskalierung von Variablen	190
Listing 14.31	Implementierung einer Funktion zur Reskalierung von Variablen	191
Listing 14.32	Verwendung der Funktion sample_without_replacement	193
Listing 14.33	Identifikation von Imbalances	194
Listing 14.34	Konsolenausgabe	194
Listing 14.35	Verwendung der Funktion Undersampling	194
Listing 14.36	Implementierung einer Methode zum Ausbalancieren von Daten	195
Listing 14.37	Konsolenausgabe	197
Listing 14.38	Verwendung der Funktion train_test_split	197

Listing 14.39	Konkatenieren der Features und Labels und Korrelationsberechnung	199
Listing 14.40	Konsolenausgabe	199
Listing 14.41	Pairplot der ausgewählten Variablen	200
Listing 14.42	Weitere Abbildungen	202
Listing 14.43	Partition der Daten in Trainings- und Testdaten und Training des Modells	205
Listing 14.44	Implementierung einer Methode zur Überprüfung der Modell Performance	206
Listing 14.45	Konsolenausgabe	208
Listing 14.46	Implementierung der Hyperparameter Optimierung des KNN Modells	210
Listing 14.47	Konsolenausgabe	212
Listing 14.48	Implementierung des optimierten KNN Modells	213
Listing 14.49	Konsolenausgabe	213
Listing 14.50	Abpeichern und Laden eines trainierten Modells	214
Listing 14.51	Implementierung eines naiven Random Forest Modells	217
Listing 14.52	Konsolenausgabe	217
Listing 14.53	Optimierung der Hyperparameter des Random Forests	218
Listing 14.54	Konsolenausgabe	220
Listing 14.55	Implementierung des optimierten Modells	220
Listing 14.56	Konsolenausgabe	220
Listing 14.57	Implementierung der Ausgabe der Feature Importances	223
Listing 15.1	Datenimport	228
Listing 15.2	Konsolenausgabe: df.columns.tolist() und df.shape()	229
Listing 15.3	Spalten entfernen aus einem DataFrame	230
Listing 15.4	Konsolenausgabe: print(df.info())	230
Listing 15.5	Von Strings zu DateTime Objekten	231
Listing 15.6	Nutzung von unique()	232
Listing 15.7	Konsolenausgabe: LActivity aus unique()	233
Listing 15.8	Nutzung von describe()	233
Listing 15.9	Konsolenausgabe: Describe	233
Listing 15.10	Boxplot für die Aktivitäten entsprechend ihres zeitlichen Anspruchs	234
Listing 15.11	Finde Aufgabe im Date Frame mit loc	234
Listing 15.12	Verwendung von Groupby für die Aktivitäten	236
Listing 15.13	Anteil der Aktivitäten an der genutzten Arbeitszeit	236
Listing 15.14	Werteüberprüfung in DataFrame zur Klassifikation von Werten mit loc	237
Listing 15.15	Berechnung der abrechenbaren Stunden	239
Listing 15.16	Datentransformation von Minuten zu Stunden	242
Listing 15.17	Umsetzung des One Hot Encoding	243

Listing 15.18	Umsetzung des One Hot Encoding	244
Listing 15.19	Implementierung eines Perceptron-Netzwerks	247
Listing 16.1	Anlegen des Schemas data_science in MySQL.	257
Listing 16.2	Anlegen der Tabelle statics in MySQL	257
Listing 16.3	Anlegen der Tabelle portfolio_transactions in MySQL.	257
Listing 16.4	Anlegen der Tabelle marketdata_daily in MySQL	257
Listing 16.5	Anlegen der Tabelle marketdata_fx_daily in MySQL.	257
Listing 16.6	Upload einer .csv Datei in eine Tabelle der Datenbank.	258
Listing 16.7	Umsetzung der Anfrage des Tickers zu einer spezifizierten ISIN	260
Listing 16.8	Konsolenausgabe: print(out) zur Funktion SYMBOL_SEARCH.	260
Listing 16.9	Umsetzung der Datenspeicherung aus dem Dictionary.	261
Listing 16.10	Umsetzung der Anfrage von Marktdaten zu einem gegebenen Ticker.	261
Listing 16.11	Speicherung der Daten in die Datenbank	261
Listing 16.12	Konsolenausgabe: print(out) zur Funktion TIME_SERIES_DAILY	262
Listing 16.13	Überführung der bereinigten Marktdaten in einen DataFrame	263
Listing 16.14	Konsolenausgabe: print(df) der Marktdaten als DataFrame	263
Listing 16.15	Erweiterung des DataFrame df	263
Listing 16.16	Upload der Marktdaten in die Datenbank über eine .csv Datei	264
Listing 16.17	Eingabeparameter zur Ansprache der Funktion FX_DAILY.	264
Listing 16.18	Eingabeparameter zur Ansprache der Funktion OVERVIEW.	265
Listing 16.19	Abfrage an die Datenbank und Speicherung in einem DataFrame.	266
Listing 16.20	Abfrage aller Security ID, zu denen noch keine Statics angelegt wurden.	268
Listing 16.21	Anfrage fehlender Ticker und Speicherung in der Datenbank	268
Listing 16.22	Anfrage der gesamten historischen Marktdaten und Upload in die Datenbank	269
Listing 16.23	Abfrage an die Datenbank zur Bestimmung fehlender Marktdaten.	271
Listing 16.24	Verarbeitung der angefragten Marktdaten und Upload in die Datenbank	272
Listing 16.25	Abfrage und Speicherung der ersten Zeile der Tabelle marketdata_fx_daily	273
Listing 16.26	Initialisierung notwendiger Bibliotheken und Komponenten	273
Listing 16.27	Darstellung des aktuellen Portfolios in MySQL	275

Listing 16.28	Wert des gesamten Portfolios zum letzten Handelstag	275
Listing 16.29	Darstellung der Portfolioallokation nach Land	276
Listing 16.30	Zusammenfassen übriger Länder mit dem entsprechenden Volumen	277
Listing 16.31	Bereitstellung des kumulierten Nominals in einem DataFrame.	277
Listing 16.32	Ausgabe aller jemals gehandelten Wertpapiere	278
Listing 16.33	Definition der Listen var und initial_date	278
Listing 16.34	Initialisierung der Berechnung der historischen Portfolioentwicklung	279
Listing 16.35	Initialisierung der Berechnung des kumulierten Portfoliowerts zu jedem Handelstag	280
Listing 16.36	Speicherung des kumulierten Handelspreises	280
Listing 16.37	Bereitstellung der Marktdaten der zu iterierenden Security ID ab dem ersten Handelstag.	280
Listing 16.38	Initialisierung und Berechnung des Portfoliowerts, vom ersten bis zum nächsten Handelstag	281
Listing 16.39	Initialisierung und Berechnung des letzten Depotwerts, vom ersten bis zum nächsten Handelstag	281
Listing 16.40	Berechnung des kumulierten Handelspreises im Fall eines Zukaufs des Wertpapiers	282
Listing 16.41	Berechnung des Portfoliowerts und des letzten Depotwerts im Fall des letzten Handelstags des Wertpapiers.	282
Listing 16.42	Speicherung des DataFrame df_portfolio	283
Listing 16.43	Initialisierung und Speicherung der berechneten Portfolio- und letzten Depotwerte	283
Listing 16.44	Datengruppierung im DataFrame df_dash.	284
Listing 16.45	Vorbereitung der Daten zur grafischen Darstellung innerhalb der App	284
Listing 16.46	App in dash – Header	286
Listing 16.47	App in dash – Grafik Portfolioentwicklung.	287
Listing 16.48	App in dash – Tabelle Portfolio	287
Listing 16.49	App in dash - Portfolio nach Land/Industrie	288
Listing 16.50	Initialisierung des app.callback	289
Listing 16.51	Darstellung des Gesamtportfolios innerhalb des app.callback	290
Listing 16.52	Initialisierung der Darstellung einzelner, selektierter Wertpapiere innerhalb des app.callback	291
Listing 16.53	Konsolenausgabe: print(df_portfolio.loc[df_portfolio ['security_id'] == i, 'Ticker'])	292
Listing 16.54	Fortsetzung der Darstellung einzelner, ausgewählter Wertpapiere innerhalb des app.callback	292

Teil I

Grundlagen der Programmierung



Einleitung

1

Benjamin M. Abdel-Karim

Dieses Kapitel gibt einen Überblick über den Inhalt, den Aufbau und die Zielsetzung des Buchs. Mit dem wachsenden Angebot an Datenquellen wächst das Interesse, diese entsprechend zu monetarisieren. Dies gilt nicht nur für digitale Geschäftsprozesse, sondern auch für die Forschung und Entwicklung. Damit sind das Verständnis und die Fähigkeit zur Datenanalyse gewinnbringende Vermögenswerte, um die gemeinschaftliche Wohlfahrt des Staats voranzubringen. Im ersten Teil des Kapitels wird der Gegenstand dieses Buchs beschrieben. Daran knüpft der Aufbau des Buchs an und es wird eine Erklärung geboten, weshalb sich Python besonders für Einsteiger eignet.

1.1 Gegenstandsbereich dieses Buchs

Die stets zunehmende Menge an Daten und die implizit entstehende Komplexität dieser Daten führen dazu, dass neue Geschäftsbereiche und Stellenbeschreibungen entstehen, sodass die Fähigkeit, diese Daten zu verarbeiten und geeignet zu analysieren, immer gefragter wird. Das Anwendungsfeld der Datenanalyse war bis vor wenigen Jahren nur im Bereich Forschung und Entwicklung zu finden. Allerdings entdecken zunehmend Start-ups, der Mittelstand und die großen Technologiekonzerne diesen Bereich für sich, um neue Geschäftsmodelle zu erschließen. Vor dem Hintergrund der Wettbewerbssicherung führt dieser Umstand dazu, dass Studierende, Berufsanfänger und Unternehmer sich frühzeitig mit den Grundlagen der Datenanalyse befassen sollten. Damit richtet sich dieses Buch an eben jene interessierten Leser, die sich mit den Grundlagen der Datenanalyse befassen

B. M. Abdel-Karim (✉)
Frankfurt am Main, Hessen, Deutschland
E-mail: BenjaminM.Abdel-Karim@gmx.de

möchten, insbesondere an Leser, die sich frühzeitig den neuen Wettbewerbsanforderungen stellen wollen.

Dieses Werk ist nicht als reines Einführungswerk zu verstehen, sondern als Nachschlagewerk zum Lösen der klassischen Problemstellungen in der Datenanalyse, die in der Regel zu Beginn der Datenanalyse auftreten. Basierend auf den Erkenntnissen zahlreicher Vorlesungen, Übungen, Mentorings und Speaker Events fasst dieses Buch einige Quellcodeauschnitte in strukturierter Art und Weise zusammen. Das Buch beginnt mit den essenziellen Grundlagen der Programmierung in Python. Diese werden durch praktische Implementierungen von Anwendungsbeispielen vertieft.

1.2 Aufbau und Zielsetzung

Als Grundlagen- und Nachschlagewerk hat dieses Buch den primären Anspruch, die Umsetzung von wesentlichen Methoden im Umgang der Datenanalyse im Kontext der Programmiersprache Python darzulegen. Der erste Teil des Buchs vermittelt die essenziellen Grundkenntnisse, um die Data-Science-Projekte im zweiten Teil des Buchs nachvollziehen zu können. Die Besonderheit in diesem Buch sind eben jene Projekte (Case Studies). Jede Case Study behandelt eine Data-Science-Herausforderung aus der Perspektive einer meiner Co-Autoren, sodass Sie als Leser in diesem Buch die Gelegenheit erhalten werden, unterschiedliche Programmierstile und -strategien kennenzulernen. Daraus ergibt sich der folgende Aufbau dieses Buchs:

- Grundlagen
 - Python Installation und Einstieg
 - Grundlegende Datentypen
 - Datenstrukturen und Indexierung
 - Kontrollstrukturen
- Data-Science-Projekte
 - Data Science und Data-Science-Prozess
 - Überblick bedeutender Bibliotheken
 - Datenvorverarbeitung
 - Datenvisualisierungen
 - Entwicklung eigener Modelle

Die Zielsetzung des Buchs ist damit, die gesammelten Erkenntnisse und Erfahrungen zu archivieren und weiterzugeben, damit der Leser aus vorangegangenen Erfahrungen der Autoren und dem entstandenen Lernprozess profitieren kann. Im Allgemeinen richtet sich das Buch mit dieser Zielsetzung an jeden interessierten und motivierten Leser, sich die Grundlagen des Programmierens in Python und von Data Science anzueignen. Außerdem ist das Buch für Studierende im Bachelor- und Masterstudium konzipiert, die ihren

Wissenshorizont erweitern möchten und durch die erworbenen Fähigkeiten imstande sind, die klassischen Aufgaben des wissenschaftlichen Arbeitens im Rahmen der Datenanalyse zu bewerkstelligen. Darüber hinaus ist das Buch für Berufspraktiker geschrieben, die im Rahmen ihrer Berufstätigkeit eine flexible und moderne Programmiersprache nutzen wollen, um die anfallenden Daten zu analysieren. In vielen Firmen werden bisher einfache Tabellenverarbeitungsprogramme eingesetzt, die allerdings schnell an ihre Verarbeitungskapazitäten stoßen. Hierzu können die vorgestellten Anwendungsfälle einen Eindruck zu den möglichen Alternativen mithilfe der Programmiersprache Python aufzeigen. Zusätzlich richtet sich dieses Buch an junge Unternehmer, die den Wert ihrer Daten erkannt haben und nun auf der Suche nach einer Inspiration sind, diese Daten geeignet auszuwerten.

Demnach möchte dieses Buch einen Beitrag zur Wissensdiffusion moderner Technologien liefern, wie beispielsweise Hardware und Programmiersprachen, und durch anschauliche und praxisnahe Beispiele zum selbstständigen Ausprobieren anregen. Vor dem Hintergrund des exponentiellen Wissenszuwachses der menschlichen Gesellschaft und der damit verbundenen Entwicklung einer hochspezialisierten Wissensgesellschaft (Stehr, 2006; Bittlingmayer & Bauer, 2006), ist die private Weiterbildung ein zentraler Bestandteil der fachlichen Weiterentwicklung.

1.3 Warum Python?

In der Welt der Programmierung existieren zahlreiche Programmiersprachen. Jede einzelne von ihnen besitzt Stärken und Schwächen. Für Einsteiger bietet sich besonders die Programmiersprache Python an. An dieser Stelle möchte ich kurz auf einige gute Gründe zur Nutzung von Python eingehen, ohne den Details aus den kommenden Kapiteln vorzugreifen:

- Python ist schnell zu erlernen, da diese Programmiersprache im Vergleich zu anderen Programmiersprachen auf wenige Syntaxsymbole setzt.
- Zudem werden Variablen dynamisch deklariert, sodass das Schreiben des Codes zu Beginn vereinfacht wird.
- Python wird im Bereich Forschung und Entwicklung sowie in vielen Praxiskontexten eingesetzt, sodass ihr eine hohe praktische Relevanz beizumessen ist.
- Python wird von einem großen Nutzerkreis durch umfassende Module (Bibliotheken) unterstützt, sodass viele Programmcode-teile schon verwendet werden können. Außerdem existieren zahlreiche Onlinekurse und Literaturbeiträge.
- Python lässt sich mit anderen Programmiersprachen wie C++ oder HTML verknüpfen, sodass komplexere Programme geschrieben werden können.

Die folgenden Kapitel gehen auf die Grundlagen der Programmiersprache ein.