

jonas FREIKNECHT
stefan PAPP

BIG DATA in der Praxis

Lösungen mit **Hadoop, Spark, HBase** und **Hive**
Daten speichern, aufbereiten, visualisieren

2. Auflage

HANSER



Im Internet: [Github-Repository](#)
zum Buch

Bleiben Sie auf dem Laufenden!



Unser **Computerbuch-Newsletter** informiert Sie monatlich über neue Bücher und Termine. Profitieren Sie auch von Gewinnspielen und exklusiven Leseproben. Gleich anmelden unter



www.hanser-fachbuch.de/newsletter



Hanser Update ist der IT-Blog des Hanser Verlags mit Beiträgen und Praxistipps von unseren Autoren rund um die Themen Online Marketing, Webentwicklung, Programmierung, Softwareentwicklung sowie IT- und Projektmanagement. Lesen Sie mit und abonnieren Sie unsere News unter



www.hanser-fachbuch.de/update



Jonas Freiknecht
Stefan Papp

Big Data in der Praxis

Lösungen mit Hadoop, Spark, HBase
und Hive

Daten speichern, aufbereiten,
visualisieren

2., erweiterte Auflage

HANSER

Die Autoren:

Jonas Freiknecht, Karlsruhe, www.jofre.de

Stefan Papp, Gafrenz, stefan.papp@data-wizard.net

Alle in diesem Buch enthaltenen Informationen, Verfahren und Darstellungen wurden nach bestem Wissen zusammengestellt und mit Sorgfalt getestet. Dennoch sind Fehler nicht ganz auszuschließen. Aus diesem Grund sind die im vorliegenden Buch enthaltenen Informationen mit keiner Verpflichtung oder Garantie irgendeiner Art verbunden. Autoren und Verlag übernehmen infolgedessen keine juristische Verantwortung und werden keine daraus folgende oder sonstige Haftung übernehmen, die auf irgendeine Art aus der Benutzung dieser Informationen – oder Teilen davon – entsteht.

Ebenso übernehmen Autoren und Verlag keine Gewähr dafür, dass beschriebene Verfahren usw. frei von Schutzrechten Dritter sind. Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Buch berechtigt deshalb auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.

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.

Dieses Werk ist urheberrechtlich geschützt.

Alle Rechte, auch die der Übersetzung, des Nachdruckes und der Vervielfältigung des Buches, oder Teilen daraus, vorbehalten. Kein Teil des Werkes darf ohne schriftliche Genehmigung des Verlages in irgendeiner Form (Fotokopie, Mikrofilm oder ein anderes Verfahren) – auch nicht für Zwecke der Unterrichtsgestaltung – reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

© 2018 Carl Hanser Verlag München, www.hanser-fachbuch.de

Lektorat: Sylvia Hasselbach

Copy editing: Jürgen Dubau, Freiburg/Elbe

Umschlagdesign: Marc Müller-Bremer, München, www.rebranding.de

Umschlagrealisation: Stephan Rönigk

Gesamtherstellung: Kösel, Krugzell

Ausstattung patentrechtlich geschützt. Kösel FD 351, Patent-Nr. 0748702

Printed in Germany

Print-ISBN: 978-3-446-45396-8

E-Book-ISBN: 978-3-446-45601-3

*„Wenn Du schnell gehen willst, geh allein.
Wenn Du weit gehen willst, geh mit anderen.“*

Afrikanisches Sprichwort

Inhalt

1	Einleitung	1
2	Big Data	7
2.1	Historische Entstehung	9
2.2	Big Data – ein passender Begriff?	10
2.2.1	Die drei V	11
2.2.2	Weitere Vs	14
2.2.3	Der Verarbeitungsaufwand ist big	14
2.2.4	Sicht der Industrie auf Big Data	15
2.3	Eingliederung in BI und Data Mining	16
3	Hadoop	21
3.1	Hadoop kurz vorgestellt	21
3.2	HDFS – das Hadoop Distributed File System	23
3.3	Hadoop 2.x und YARN	28
3.4	Hadoop als Single-Node-Cluster aufsetzen	30
3.4.1	Falls etwas nicht funktioniert	44
3.5	Map Reduce	46
3.6	Aufsetzen einer Entwicklungsumgebung	49
3.7	Implementierung eines Map-Reduce-Jobs	56
3.8	Ausführen eines Jobs über Kommandozeile	68
3.9	Verarbeitung im Cluster	72
3.10	Aufsetzen eines Hadoop-Clusters	74
3.11	Starten eines Jobs via Hadoop-API	86
3.12	Verketteten von Map-Reduce-Jobs	99
3.13	Verarbeitung anderer Dateitypen	115
3.14	YARN-Anwendungen	130
3.14.1	Logging und Log-Aggregation in YARN	131
3.14.2	Eine einfache YARN-Anwendung	134
3.15	Vor- und Nachteile der verteilten Verarbeitung	159

3.16 Die Hadoop Java-API	160
3.16.1 Ein einfacher HDFS-Explorer	161
3.16.2 Cluster-Monitor	173
3.16.3 Überwachen der Anwendungen im Cluster	175
3.17 Gegenüberstellung zur traditionellen Verarbeitung	177
3.18 Big Data aufbereiten	178
3.18.1 Optimieren der Algorithmen zur Datenauswertung	178
3.18.2 Ausdünnung und Gruppierung	180
3.19 Ausblick auf Apache Spark	182
3.20 Markt der Big-Data-Lösungen	184
4 Das Hadoop-Ecosystem	187
4.1 Ambari	188
4.2 Sqoop	189
4.3 Flume	189
4.4 HBase	190
4.5 Hive	191
4.6 Pig	191
4.7 ZooKeeper	191
4.8 Oozie	192
4.9 Mahout	193
4.10 Data Analytics und das Reporting	193
5 NoSQL und HBase	195
5.1 Historische Entstehung	195
5.2 Das CAP-Theorem	196
5.3 ACID und BASE	197
5.4 Typen von Datenbanken	198
5.5 Umstieg von SQL und Dateisystemen auf NoSQL oder HDFS	201
5.5.1 Methoden der Datenmigration	201
5.6 HBase	203
5.6.1 Das Datenmodell von HBase	203
5.6.2 Aufbau von HBase	206
5.6.3 Installation als Stand-alone	207
5.6.4 Arbeiten mit der HBase Shell	209
5.6.5 Verteilte Installation auf dem HDFS	211
5.6.6 Laden von Daten	214
5.6.7 HBase Java-API	226
5.6.8 Der Umstieg von einem RDBMS auf HBase	249
6 Data Warehousing mit Hive	253
6.1 Installation von Hive	254

6.2	Architektur von Hive	256
6.3	Das Command Line Interface (CLI)	257
6.4	HiveQL als Abfragesprache	259
6.4.1	Anlegen von Datenbanken	259
6.4.2	Primitive Datentypen	260
6.4.3	Komplexe Datentypen	260
6.4.4	Anlegen von Tabellen	261
6.4.5	Partitionierung von Tabellen	262
6.4.6	Externe und interne Tabellen	262
6.4.7	Löschen und Leeren von Tabellen	263
6.4.8	Importieren von Daten	264
6.4.9	Zählen von Zeilen via count	265
6.4.10	Das SELECT-Statement	265
6.4.11	Beschränken von SELECT über DISTINCT	269
6.4.12	SELECT auf partitionierte Tabellen	269
6.4.13	SELECT sortieren mit SORT BY und ORDER BY	270
6.4.14	Partitionieren von Daten durch Bucketing	271
6.4.15	Gruppieren von Daten mittels GROUP BY	272
6.4.16	Subqueries – verschachtelte Abfragen	273
6.4.17	Ergebnismengen vereinigen mit UNION ALL	273
6.4.18	Mathematische Funktionen	274
6.4.19	String-Funktionen	276
6.4.20	Aggregatfunktionen	276
6.4.21	User-Defined Functions	277
6.4.22	HAVING	285
6.4.23	Datenstruktur im HDFS	286
6.4.24	Verändern von Tabellen	286
6.4.25	Erstellen von Views	289
6.4.26	Löschen einer View	289
6.4.27	Verändern einer View	289
6.4.28	Tabellen zusammenführen mit JOINS	290
6.5	Hive Security	292
6.5.1	Implementieren eines Authentication-Providers	298
6.5.2	Authentication-Provider für HiveServer2	303
6.5.3	Verwenden von PAM zur Benutzerauthentifizierung	303
6.6	Hive und JDBC	304
6.7	Datenimport mit Sqoop	322
6.8	Datenexport mit Sqoop	324
6.9	Hive und Impala	325
6.10	Unterschied zu Pig	326
6.11	Zusammenfassung	327

7	Big-Data-Visualisierung	329
7.1	Theorie der Datenvisualisierung	329
7.2	Diagrammauswahl gemäß Datenstruktur	335
7.3	Visualisieren von Big Data erfordert ein Umdenken	336
7.3.1	Aufmerksamkeit lenken	337
7.3.2	Kontextsensitive Diagramme	339
7.3.3	3D-Diagramme	341
7.3.4	Ansätze, um Big-Data zu visualisieren	342
7.4	Neue Diagrammart	344
7.5	Werkzeuge zur Datenvisualisierung	348
7.6	Entwicklung einer einfachen Visualisierungskomponente	352
8	Auf dem Weg zu neuem Wissen – Aufbereiten, Anreichern und Empfehlen	365
8.1	Eine Big-Data-Table als zentrale Datenstruktur	368
8.2	Anreichern von Daten	370
8.2.1	Anlegen einer Wissensdatenbank	371
8.2.2	Passende Zuordnung von Daten	372
8.3	Diagrammempfehlungen über Datentypanalyse	376
8.3.1	Diagrammempfehlungen in der BDataTable	378
8.4	Textanalyse – Verarbeitung unstrukturierter Daten	384
8.4.1	Erkennung von Sprachen	385
8.4.2	Natural Language Processing	386
8.4.3	Mustererkennung mit Apache UIMA	394
9	Infrastruktur	415
9.1	Hardware	416
9.2	Betriebssystem	417
9.2.1	Paketmanager	417
9.2.2	Git	418
9.2.3	VIM	419
9.2.4	Terminalumgebung	419
9.3	Virtualisierung	420
9.4	Container	420
9.4.1	Docker-Crashkurs	421
9.4.2	Infrastructure as Code	424
9.5	Distributionen	424
9.6	Reproduzierbarkeit	425
9.7	Zusammenfassung	425
10	Programmiersprachen	427
10.1	Merkmale	428
10.1.1	Funktionale Paradigmen	428

10.2	Big-Data-Programmiersprachen	429
10.2.1	Java	429
10.2.2	Scala	430
10.2.3	Python	433
10.2.4	R	436
10.2.5	Weitere Programmiersprachen	437
10.3	Zusammenfassung	438
11	Polyglot Persistence	439
11.1	Praxis	440
11.1.1	Redis	440
11.1.2	MongoDB	443
11.1.3	Neo4j	443
11.1.4	S3	444
11.1.5	Apache Kudu	447
11.2	Zusammenfassung	447
12	Apache Kafka	449
12.1	Der Kern	450
12.2	Erste Schritte	450
12.3	Dockerfile	454
12.4	Clients	454
12.5	Python Chat Client	454
12.6	Zusammenfassung	456
13	Data Processing Engines	457
13.1	Von Map Reduce zu GPPEs	457
13.1.1	Herausforderungen	458
13.1.2	Verfahren zur Verbesserung	459
13.1.3	Von Batch und Streaming zu Lambda	461
13.1.4	Frameworks in a Nutshell	462
13.2	Apache Spark	462
13.2.1	Datasets	462
13.2.2	Von RDDs zu Data Frames	463
13.2.3	Hands On Apache Spark	463
13.2.4	Client-Programme schreiben	465
13.2.5	Das Spark-Ecosystem	470
13.3	Zusammenfassung	474
14	Streaming	475
14.1	Kernparadigmen	475
14.2	Spark Streaming	478
14.2.1	Beispiel	479

14.3	Apache Flink	480
14.4	Zusammenfassung	483
15	Data Governance	485
15.1	Begriffsdschungel	486
15.2	Governance-Pfeiler	487
15.2.1	Transparenz	487
15.2.2	Verantwortung	488
15.2.3	Standardisierung	489
15.3	Fokusthemen von Data Governance	489
15.3.1	Policies	489
15.3.2	Quality	490
15.3.3	Compliance	490
15.3.4	Business Intelligence	490
15.4	Datenschutz	491
15.4.1	Werkzeuge	492
15.5	Sicherheit im Hadoop-Ecosystem	497
15.6	Metadatenmanagement	498
15.6.1	Open-Source-Werkzeuge	499
15.6.2	Kommerzielle Datenkataloge	500
15.7	Organisatorische Themen	500
15.7.1	Privacy by Design	501
15.7.2	k-Anonymity	501
15.7.3	Standards	503
15.8	Zusammenfassung	503
16	Zusammenfassung und Ausblick	505
16.1	Zur zweiten Auflage 2018	505
16.2	Zur ersten Auflage 2014	507
17	Häufige Fehler	511
18	Anleitungen	517
18.1	Installation und Verwendung von Sqoop2	517
18.2	Hadoop für Windows 7 kompilieren	523
19	Literaturverzeichnis	527
Index	531

1

Einleitung

Der Begriff Big Data ist seit der ersten Auflage im Jahr 2014 zu einem zentralen Thema in der IT-Landschaft von Unternehmen geworden. Kaum jemand spricht noch von einer kurzfristigen Hype. Big-Data-Technologien bestimmen auch den Inhalt vieler Fachzeitschriften und -bücher.

Mit Apache Hadoop und NoSQL-Datenbanken können große, meist unstrukturierte Datenmengen effizient verteilt verarbeitet werden. Darüber hinaus haben in der letzten Zeit Data Processing Engines wie Apache Spark und Messaging-Systemen wie Kafka an Popularität gewonnen, da sie neue Möglichkeiten bieten, wie noch schneller Wert aus Daten extrahiert werden kann. Machine-Learning- und Deep-Learning-Werkzeuge sind mittlerweile ebenfalls ein wesentlicher Teil einer Big-Data-Systemlandschaft.

In den Medien wird häufig der generelle Mehrwert, der durch den Einsatz von Big-Data-Technologien für Banken, Automobilhersteller, Forschungseinrichtungen, Versicherungen etc. entsteht, hervorgehoben. Es wird die Notwendigkeit betont, sich mit den firmeninternen und öffentlichen Datenmengen zu beschäftigen, um dem eigenen Unternehmen einen Wettbewerbsvorteil zu verschaffen. Durch die Auswertung großer Datenmengen sollen neue, geschäftskritische Informationen gewonnen werden, welche die Grundlage für Unternehmensentscheidungen sein können. Daten gelten als das Öl des 21. Jahrhunderts.

Die sogenannte „Value Proposition“ für Unternehmen ist somit klar: Mit Big Data können Firmen Kosten sparen, schnellere Entscheidungen treffen und neue Märkte generieren. Zahlreiche Berater nutzen Domänensprache, also eine spezialisierte Businesssprache, und erstellen für ihre Kunden neue Churn-Modelle¹, Customer Experience Solutions² und Data Monetisation Use Cases³.

Die Frage nach dem *Wie* bleibt in vielen Büchern unbeantwortet. Wie funktionieren die Lösungen im Detail, um Informationen aus Daten zu extrahieren? Wie integrieren sich Hadoop und Co. in bisherige Business-Intelligence-Architekturen und wie ist das Zusammenspiel mit dem Data Warehouse? Welche Bedeutung haben Frameworks wie Spark und Kafka in einer modernen Big-Data-Architektur? Und wie schaffen es Data Scientists, große

¹ Modelle, in denen untersucht wird, warum Kunden kündigen und wie man das verhindern kann.

² Lösungen, mit denen man untersucht, wie sich Kunden verhalten, um ihr Verhalten abzuschätzen und zu lernen, wie man Geschäftsprozesse optimieren kann, um Kundenanliegen besser zu erfüllen.

³ Data Monetisation ist die Disziplin, wie Firmen Geld aus Daten machen können. Beispielsweise können Telekommunikationsunternehmen gesammelte Geolokationsdaten auch an Dritte verkaufen

Datenmengen aufzubereiten, zu visualisieren und den Fachabteilungen zugänglich zu machen? Mit welchen Tools wird eine Big-Data-Architektur geschaffen, um für neue, bisher unbekannte Herausforderungen optimal aufgestellt zu sein?

Dieser Fragenkatalog ließe sich beliebig fortführen und erweitern, denn die *technischen* Aspekte des Big-Data-Trends werden nur selten in ausreichendem Detailgrad diskutiert. Und wenn, dann nicht in Form von Gesamtlösungen, sondern in kleinen, gut verdaulichen Häppchen. Gründe dafür gibt es viele. Zum einen sind Big Data-Experten rar, und nur wenige haben die Zeit, ihr Wissen in Büchern oder Fachartikeln weiterzugeben. Darüber hinaus steht hinter Big Data nicht nur eine einzelne neue Technologie. Big-Data-Architekturen basieren oft auf vielen verschiedenen Komponenten, die erst im Zusammenspiel ihr ganzes Potenzial entfalten.

Das Thema ist unter technischen Gesichtspunkten umfangreich und bietet zahlreiche Bereiche, in denen es sich neues Wissen anzueignen gilt. Wichtig ist daher, dass Sie eine hohe Affinität zum Forschen und zum Experimentieren mitbringen. Viele Big-Data-Technologien werden laufend weiterentwickelt und andere verschwinden wieder. Dazu kommen die unterschiedlichsten Anforderungen von Unternehmen an Big Data, die wiederum unterschiedlich aufgebaute Big-Data-Architekturen erfordern.

Was den Leser in diesem Buch erwartet

Big Data wird in diesem Buch aus einer Engineering-Sicht betrachtet. Einzelne fachliche Anwendungsgebiete werden natürlich berücksichtigt, aber das Primärziel ist, Lesern, die an der Technologie interessiert sind, einen praktischen Einstieg in moderne Lösungsansätze zu bieten, die für die Verarbeitung von Daten z. B. aus sozialen Netzwerken, unstrukturierten Webseiten, umfangreichen Fließtextdokumenten und geografischen Daten nötig sind. Dabei wird nicht nur gezeigt, wie große Datenmengen in einem Cluster verarbeitet, sondern auch über ein Data Warehouse bereitgestellt oder mit neuen, innovativen Diagrammen visualisiert werden können. Themen wie *NoSQL* werden besprochen und im praktischen Teil HBase als Vertreter dieser Kategorie aktiv eingesetzt. Apache Hive wird als Data-Warehouse-Software vorgestellt, um zu zeigen, inwiefern auf Big Data mit Abfragesprachen ähnlich SQL zugegriffen werden kann. Darüber hinaus wird Hive dann auch neueren SQL-Trends wie dem Framework Apache Spark gegenübergestellt. Sie lernen, welche neuen Diagrammarten dabei unterstützen, große Datenmengen mit komplexen Beziehungen untereinander zu visualisieren und zu verstehen. Diese versprochenen Erläuterungen werden nicht nur in Textform gegeben, Sie werden auch aktiv in den Entwicklungsprozess miteinbezogen und wo möglich, werden die theoretischen Hintergründe nähergebracht. Neben den bekannten Apache-Projekten wie Hadoop, Hive und HBase werden auch einige weniger bekannte Frameworks wie Apache UIMA oder Apache OpenNLP besprochen, um gezielt die Verarbeitung unstrukturierter Daten zu behandeln. Dazu wird in dem Buch gezeigt, wie Sie kleinere Projekte entwickeln, um die Kniffe bezüglich der Nutzung der neuen Software kennenzulernen und zu verstehen. Das Ziel ist es, Sie auf den Effekt und den Mehrwert der neuen Möglichkeiten aufmerksam zu machen, sodass Sie diese konstruktiv in Ihr Unternehmen tragen und für sich und Ihre Kollegen ein Bewusstsein für den Wert Ihrer Daten schaffen.

Voraussetzungen

Wie im Vorwort erwähnt, ist die wichtigste Voraussetzung sicherlich die Experimentierfreude und die Bereitschaft, Neues zu erlernen und alte Gewohnheiten und Denkweisen zu hinterfragen. Die technischen Vorkenntnisse, die Sie mitbringen sollten, um dieses Buch flüssig lesen und nachvollziehen zu können, lassen sich in die drei Bereiche *Entwicklungsumgebungen*, *Entwicklung* und *Betrieb* unterteilen. Sie sollten sich mit Java und IntelliJ oder Eclipse als Entwicklungsumgebung auskennen. Da Hadoop, HBase, Hive, Sqoop etc. auf Ubuntu, in virtuellen Umgebungen und Dockercontainern installiert wird, ist es ebenfalls hilfreich, sich in einer Unix-Umgebung bewegen zu können und die grundlegenden Befehle zu kennen, um etwa Verzeichnisse zu wechseln, anzulegen oder zu löschen. Zudem ist eine etwas stärkere Systemumgebung vonnöten, um alle Szenarien aus dem Buch zu Hause nachvollziehen zu können. Wesentlich ist dabei RAM. 8 GB RAM sollten schon vorhanden sein. Die gute Nachricht ist, dass Sie für die verwendete Software kein Geld bezahlen müssen, denn alle in diesem Buch verwendeten Komponenten stehen in vollem Umfang kostenlos im Internet zur Verfügung.

Alle Beispiele in diesem Buch werden Schritt für Schritt aufgebaut und führen Sie somit hin zu einer fertigen und funktionstüchtigen Implementierung.

Die verwendeten Softwareversionen

In der ersten Auflage dieses Buches wurde Hadoop in der Version 2.2.0 verwendet, ergänzt um die weiteren Komponenten wie Hive und HBase in der jeweils zum damaligen Zeitpunkt aktuellen Version. Diese Versionierung wurde in der 2. Auflage beibehalten.

Die vorgestellten Komponenten wie Hadoop, Hive, HBase werden natürlich regelmäßig um neue Features erweitert, bei der Kernarchitektur wird jedoch auf eine hohe Abwärtskompatibilität geachtet. Aus diesem Grund sind alle Beispiele aus dem Buch weiterhin lauffähig und die Software kann entsprechend der Anleitungen im Buch installiert werden. Wenn Sie mit einer neueren Version arbeiten möchten, muss lediglich die Versionsnummer angepasst werden.

Steht also im Buch beispielsweise

```
sudo wget http://mirror.lwnetwork.org.uk/APACHE/hadoop/common/hadoop-2.2.0/hadoop-2.2.0.tar.gz
```

dann können Sie z. B. `hadoop-2.2.0` durch `hadoop-3.1.0` ersetzen und für die Java-Installation die Version 8 verwenden, wobei Sie anstelle von `sudo apt-get install openjdk-7-jdk` den Befehl `sudo apt-get install openjdk-8-jdk` ausführen.

```

hduser@stefanpapp-VirtualBox: ~
File Edit View Search Terminal Help
hduser@stefanpapp-VirtualBox:~$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu Bionic Beaver (development branch)
Release:       18.04
Codename:      bionic
hduser@stefanpapp-VirtualBox:~$ hadoop version
Hadoop 3.1.0
Source code repository https://github.com/apache/hadoop -r 16b70619a24cdf5d3b0f
cf4b58ca77238ccb6d
Compiled by centos on 2018-03-30T00:00Z
Compiled with protoc 2.5.0
From source with checksum 14182d20c972b3e2105580a1ad6990
This command was run using /usr/local/hadoop/share/hadoop/common/hadoop-common-3
.1.0.jar
hduser@stefanpapp-VirtualBox:~$

```

Bild 1.1 Beispiel mit Hadoop 3.1.0 und Ubuntu 18.04

Sie können also problemlos mit den aktuellen Versionen arbeiten, indem Sie beim Ausführen der Praxisbeispiele die Versionsnummern ersetzen. Darüber hinaus können Sie die Beispiele natürlich auch auf Basis der in diesem Buch verwendeten Versionen nachvollziehen.

Für wen ist dieses Buch geschrieben?

Dieses Buch ist für Menschen konzipiert, die sich in ihrem Beruf praktisch mit dem Thema Big Data auseinandersetzen. Dieses Buch soll aber nicht nur Informatiker ansprechen, sondern auch solche, die aus anderen Disziplinen kommen und vielleicht eine tolle Idee (und jede Menge Daten) haben und damit planen, z. B. ein Start-up zu gründen. Ebenso sollen Studenten und Auszubildende mit dieser Lektüre begleitet werden, die das Buch als Ergänzung zu einer Vorlesung oder zum Schreiben einer Seminararbeit verwenden können. Wenn Sie sich also zu einer Gruppe der in Bild 1.1 genannten Tätigkeitsfelder zählen oder mit einem der genannten Themen beschäftigen, dann lohnt sich ein Blick ins Buch besonders.

Informatiker	Studenten und Auszubildende	Start-ups	IT-Interessierte
<ul style="list-style-type: none"> • Analysten • BI-Verantwortliche • Data-Scientists • Consultants 	<ul style="list-style-type: none"> • Informatiker • Mathematiker • Physiker • Chemiker • Biologen 	<ul style="list-style-type: none"> • Daten auf mobilen Geräten • Visualisierung • Smarter-Cities 	<ul style="list-style-type: none"> • <i>Datenbewusstsein</i> entwickeln • Neuerungen entdecken • Fachsimpeln

Bild 1.2 Gruppen und Themen, die in Zusammenhang mit Big Data genannt werden

Menschen, die Big Data von einer Business-Ebene betrachten, können durchaus das eine oder andere aus dem Buch lernen, wenngleich der Fokus des Buches mehr auf Technologie als auf den Möglichkeiten einer „Data Monetisation“ liegt. Wer jedoch die Technologie versteht, wird auch besser verstehen, wie man daraus Geschäftsfelder erschließen kann.

Ein großes Thema bei Big Data sind auch gesellschaftliche Auswirkungen der Digitalisierung (bei der Big Data eine wesentliche Rolle spielt) und das Thema Datenschutz. In einer Zeit, in der Unternehmen bei Datenmissbrauch horrend Strafen drohen, kann dieser Aspekt nicht ignoriert werden. Allerdings wird sich dieses Buch mehr auf die technischen als auf die juristischen oder ethischen Herausforderungen konzentrieren.

Warum „Big Data in der Praxis“?

Jonas Freiknecht, der Autor der ersten Auflage, sieht sich als Learning-by-Doing-Typ, dem es liegt, sich Wissen anhand von praktischen Erfahrungen anzueignen. Wenn er Befehle auf der Tastatur eingibt, kann er sich diese einfach besser merken. Häufig entsteht auch dann ein tiefergehendes Interesse an der Materie, mit der er sich gerade beschäftigt, sodass er dann bereit ist, die Theorie nachzuholen.

Stefan Papp, der Autor, der die erste Auflage des Buches aktualisiert und erweitert hat, sieht das genauso: Technische Details lassen sich oft durch die praktische Anwendung am besten verstehen. Dadurch erhoffen sich beide Autoren den Effekt, dass Sie die Verbindung zwischen Hintergrundwissen und der tatsächlichen Anwendung schnell herstellen und verinnerlichen und es Ihnen beim Lesen nicht langweilig wird.

Vorgehensweise und Struktur

Dieses Buch beginnt mit einer theoretischen Einführung in alle Themen rund um Big Data. Neben der historischen Entwicklung des Begriffs und der Diskussion einiger unterschiedlicher Definitionen wird mithilfe von Studien und Umfragen gezeigt, welche Industrien welche Themen mit dem Begriff Big Data verbinden. Somit wird die Erwartungshaltung verschiedener Menschen in Bezug auf den Trend dargelegt. Eine Gegenüberstellung der Begrifflichkeiten BI, Data Mining und Big Data soll dabei helfen, Zusammenhänge, Unterschiede und gegenseitige Ergänzung der drei Begrifflichkeiten zu erkennen und Sie befähigen, diese gegeneinander abzugrenzen und an anderer Stelle Schnittpunkte zwischen ihnen zu finden.

Im Anschluss folgt ein Kapitel über Hadoop, in dem die Installation, Konfiguration und Bedienung erklärt wird. Dabei wird jeweils auf die Besonderheiten bei der Verwendung mit einem *Single-* oder *Multi-Node-Cluster* eingegangen. Es schließt sich die Entwicklung von Map-Reduce-Jobs und YARN-Anwendungen an, gefolgt von einem ausführlichen Abschnitt zur Arbeit mit der Hadoop-API, um den Zugriff auf das HDFS, den Resource Manager etc. zu erklären.

Nachdem die Funktionsweise und die Idee hinter Hadoop bekannt sind, wird in Kapitel 4 kurz vorgestellt, welche Projekte um Hadoop herum in dessen Eco-System existieren und welche Aufgaben diese haben. In Kapitel 5 wird das Thema NoSQL aufgegriffen, theoretisch erläutert und praktisch unter Zuhilfenahme von HBase umgesetzt. Dabei wird nicht nur gezeigt, wie HBase installiert und eingerichtet wird, sondern auch, wie auf dessen Daten, entweder über das Terminal oder die Java-API, zugegriffen werden kann.

Analog dazu wird in Kapitel 6 das Thema Data Warehousing mit Apache Hive besprochen und gezeigt, wie sich Hive in das Big-Data-Umfeld integrieren lässt. Elementarer Bestandteil dieses Kapitels ist die Abfragesprache *HiveQL* mit all ihren Ausprägungen und die Verwendung von Hive über einen herkömmlichen JDBC-Adapter, um aus einer Java-Anwendung Abfragen absetzen und auswerten zu können.

Das Thema Datenvisualisierung wird in Kapitel 7 behandelt, in dem zuerst einige Visualisierungsframeworks vorgestellt und verglichen werden. Danach wird mit *D3.js* ein Set von Visualisierungskomponenten erarbeitet, mit denen in einer Beispieldatenanwendung ein paar ansehnliche Diagramme gezeichnet werden. Im theoretischen Teil dieses Kapitels geht es darum, was man beachten muss, wenn jemand plant, große Datenmengen auf kleinem Raum unterzubringen, und welche Trends und Möglichkeiten es dabei gibt.

In Kapitel 8 soll das Thema Informationsgewinnung nähergebracht werden, das zum einen einen Zusammenschritt aller bisher kennengelernten Techniken in einer schicken Gesamtlösung vereint und des Weiteren auf die Besonderheiten bei der Verarbeitung von unstrukturierten Daten mit aktuellen Text-Mining-Frameworks eingeht, darunter Apache UIMA und Apache OpenNLP. Diese werden ebenfalls als Bestandteil in das hier zu entwickelnde Programm einfließen.

Erweitert wurde die zweite Auflage des Buches um sechs neue Kapitel zu Hadoop-unabhängigen Frameworks. In Kapitel 9 wird das Konzept *Infrastructure as Code* vorgestellt und gezeigt, wie man mit Docker-Containern arbeitet. Kapitel 10 widmet sich den funktionalen Programmierparadigmen und stellt verschiedene Programmiersprachen vor, die häufig in Big-Data-Szenarien Anwendung finden. Kapitel 11 beschreibt das Konzept von *Polyglot Persistence*, dem Weg, Daten in mehrere unterschiedliche Datenbanken und Container zwischenzulagern. In Kapitel 12 wird Apache Kafka als Messaging-System vorgestellt, bevor es in Kapitel 13 um die Data Processing Machine Apache Spark geht und in Kapitel 14 Daten-Streaming mit Apache Flink vorgestellt wird. Kapitel 15 beschäftigt sich mit dem Thema *Data Governance* und der effektiven Verwaltung von Daten.

Das Buch schließt ab mit einem Ausblick und zwei weiteren Kapiteln, die Ihnen Lösungen zu häufigen Fehlern bei der Arbeit mit Hadoop, Hive und HBase und ergänzende Anleitungen bieten.



Die Beispieldaten zum Buch

Die Beispieldaten zum Buch können unter https://github.com/StefanPapp/big-data_buch heruntergeladen werden und enthalten u. a. die fertigen Projekte, die im Buch erarbeitet werden. Nutzen Sie diese gerne als Nachschlagewerk, um Vorgehensweisen und Verwendung der entsprechenden APIs im Detail zu verstehen. Last, but not least sind online diverse Testdatensätze zu finden, die gerne während der Entwicklung und Erprobung der Anwendungen genutzt werden dürfen. Für die Daten in der Wissensdatenbank liegen im Ordner *Lizenzdateien* die Quellen der Daten vor. Die generierten Beispieldatensätze sind zufällig gewählt bzw. generiert, sodass Übereinstimmungen von Namen, Adressen, Berufen oder anderen Eigenschaften mit denen von realen Personen nur zufällig sind.

Wenn Sie Fragen oder Anregungen bezüglich dieses Buchs haben, würden sich beide Autoren freuen, wenn Sie sie kontaktieren. Jonas Freiknecht ist über www.jofre.de kontaktierbar. Stefan Papp erreichen Sie am besten über LinkedIn oder Xing. Beide Autoren wünschen viel Spaß bei der vorliegenden Lektüre.

2

Big Data

Zu Beginn haben Sie bereits eine kurze Einführung in das Thema Big Data erhalten, die in diesem Kapitel weiter vertieft wird. Bild 2.1 zeigt eine Statistik, die eine Umfrage von Gartner (Kart, 2012) aus dem Jahre 2012 verbildlicht. Darin wurde evaluiert, welche Industrien sich mit dem Thema Big Data in welchem Ausmaß auseinandersetzen.

Has your organization already invested in technology specifically designed to address the big data challenge?

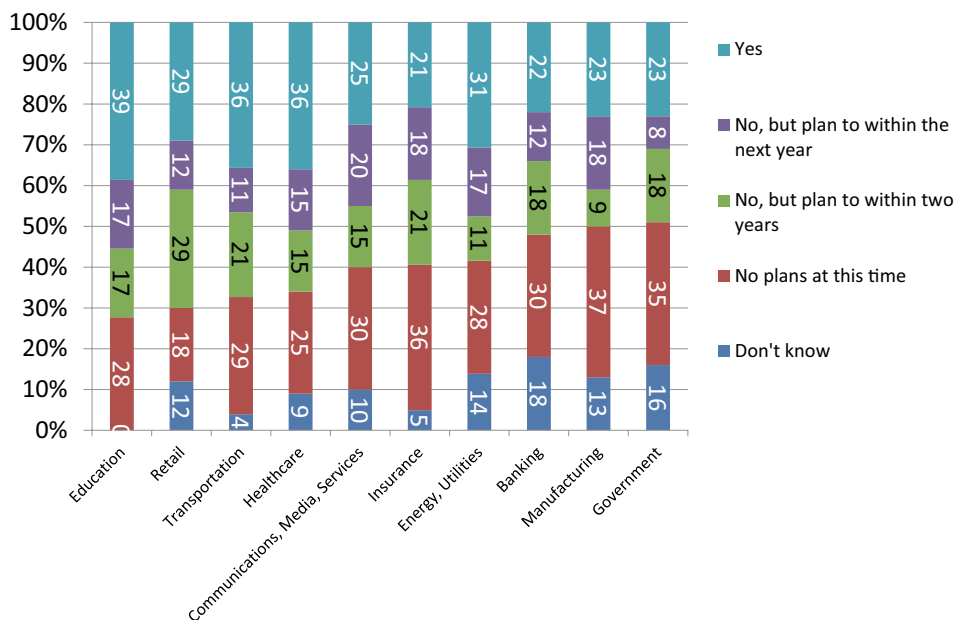


Bild 2.1 Big Data regte in 2012 das Interesse sämtlicher Industrien an.

Das wenig verblüffende Ergebnis war, dass sich alle bereits 2012 damit beschäftigt haben, allen voran die Sparte *Education*. Es gibt den berühmten Ausspruch: „Big data is like teenage sex: everyone talks about it, nobody really knows how to do it, everyone thinks everyone else is doing it, so everyone claims they are doing it.“

Es ist davon auszugehen, dass dieser Spruch 2012 stärker zutraf als heute. Viele Firmen wussten damals nicht, wie Apache Hadoop in ihren Unternehmen genutzt werden konnte. Viele Firmen erhofften sich auch eine kostenlose Datenbank, die ihr teures Data Warehouse ablösen sollte, was sich für manche später als kostspieliger Irrtum herausstellte. Es ist jedoch eine gewisse Grundeinstellung zu erkennen, die aussagt, dass Big Data schon 2012 als wichtige Neuerung wahr- und ernst genommen wurde.

2017 schaut die Lage schon anders aus. Niemand spricht mehr von einem Hype. Zahlreiche Unternehmen investieren Millionen Euro in Big Data. Einige davon gehen in die Cloud, andere investieren in eine sogenannte On-Premise-Lösung, was nichts anderes als ein eigener Cluster im eigenen Datencenter ist. In zahlreichen Big-Data-Programmen arbeiten Firmen daran, sich neue Märkte und Geschäftsfelder aufzubauen. Manche gehen sogar so weit zu behaupten, dass diverse Industrien nur über Investitionen in Big Data langfristig überleben können.

Banken, die seit Jahrzehnten mit Mainframe-Architekturen arbeiten und hohe Personalkosten durch Filialen haben, werden durch FinTechs unter Druck gesetzt, die schlanker und agiler aufgebaut sind und alle ihre Dienste online anbieten. Bessere Einsichten in Daten soll diesen Banken helfen, konkurrenzfähig zu bleiben.

Amazon weitet seine Lebensmittelparte auch in Europa aus, und mittels Machine-Learning-Algorithmen und anderen Diensten ist es möglich, den Einkauf nahezu zu automatisieren. Das ist ein gutes Beispiel dafür, wie der Erfolg von Big Data auch von gesellschaftlichen Faktoren abhängt. Bestehende Generationen sind es noch gewohnt, im Supermarkt einzukaufen. Jedoch wächst eine neue Generation heran, die mit Online Stores aufgewachsen ist. Passt das Angebot und die Zustelllogistik, ist es durchaus denkbar, dass die sogenannten Digital Natives in ein paar Jahren auch Lebensmittel und Haushaltsprodukte fast ausschließlich im Internet einkaufen könnten und das Filialnetz der Supermarktketten ähnlich wie bei Banken zu schrumpfen beginnt.

Der Wert von Telekommunikationsdaten ist hoch, und diese können für zahlreiche Dienste verwendet werden. Irgendwann könnten die Einnahmen über die Bewegungsdaten der Kunden mehr Geld einbringen als die Einnahmen durch Mobilfunkverträge.

Wer kann sich noch eine Zukunft ohne autonomes Fahren vorstellen? Die zahlreichen Sensoren, die in Autos verbaut sind, können zudem dazu verwendet werden, Positionsdaten auszuwerten.

Es gibt de facto kaum eine Industrie, die nicht mit Big Data in Berührung ist. Manche Anwendung wirkt für viele bedrohlich, andere freuen sich darauf. Aber Big Data ist nicht mehr aufzuhalten.

Die Gartner Reports beschreiben nun oft andere Probleme als in 2012. Jeder nutzt Big Data, aber scheinbar haben viele Probleme damit, Big-Data-Systeme vom Labstatus in die Produktion zu bringen (Gardner, 2016). Auch fühlen sich viele Firmen von der Komplexität erschlagen. Früher wurde nur von Hadoop gesprochen, mittlerweile drängen neue Technologien und Verfahren auf den Markt, die teils von Hadoop unabhängig sind und deren Wert für viele Firmen erst erschlossen werden muss.

Auch wissen viele Firmen nicht, wie sie die volle Bandbreite von der Analyse der Daten bis zum Betrieb bewerkstelligen sollen. Zahlreiche Berater sprechen vom Data Swamp und der

Gefahr, dass Big-Data-Systeme ohne Governance undurchschaubar werden. Auch Datenschutz, speziell neue Gesetzesnovellen wie die GDPR, machen Firmen zu schaffen.

Bevor es in den folgenden Abschnitten an die Begriffsdefinitionen von Big Data geht, soll ein kurzer Einblick in dessen historische Entstehung gegeben und skizziert werden, wo die Big-Data-Bewegung ihren Ursprung hat.

■ 2.1 Historische Entstehung

Einen konkreten Ursprung für den Begriff Big Data gibt es nicht. Zwar ist bekannt (Press, 2012), dass der Begriff selbst von Michael Cox und David Ellsworth verhältnismäßig früh (im Jahr 1997) öffentlich in einem Paper genannt wurde (Cox, et al., 1997), jedoch wird auf *Wikipedia* der Ursprung des Begriffs kontrovers diskutiert, und andere Quellen werden genannt (Mashey), die behaupten, dass John Mashey den Terminus *Big* bereits 1994 im Zusammenhang mit Datenmengen verwendete. Dies geschah jedoch vorwiegend als stilistisches Mittel, um die Komplexität einiger Teilbereiche der IT hervorzuheben. Darunter waren auch *Big Bandwidth*, *Big Physics*, *Big Latency* und eben auch *Big Data*, sodass der Begriff zwar gleich dem heutigen verwendet wurde, die Bedeutung jedoch nicht oder nur in Teilen übereinstimmte.



Bild 2.2 Frühe Nennung des Begriffs Big Data auf der IEEE Supercomputing Conference 1996 in Pittsburgh; Foto: Michael Woodacre

Im November 2009 entstand der erste Wikipedia-Artikel zu Big Data und wurde vom Benutzer John Blackburn prompt wieder gelöscht. Die Begründung war folgende:

„Delete as per nom – it is simply a combination of big and data, dictionary words which have no place here. I’m not even sure it’s a neologism, and even if it was it doesn’t need an article.“
(John Blackburne)

Nach einigen Diskussionen, die besagten, dass *Big Band* oder *Big Bang* ebenso gelöscht werden müssten, wenn Big Data doch auch nur ein zusammengesetzter Begriff aus einem Adjektiv und einem Substantiv sei, wurde der Artikel letztendlich zugelassen und sogar später (2011) in den *Gartner Hype Cycle* aufgenommen (McBurney, 2012).

Besonders interessant an dieser Betrachtung ist, dass Doug Cutting bereits 2006 unter der Schirmherrschaft von Yahoo an Hadoop arbeitete. Die Definition des Begriffs Big Data hat also von der ersten Open-Source-Implementierung von Hadoop in 2006 bis hin zum Wikipedia-Artikel fünf Jahre benötigt, um zu reifen und um Big Data als eigenes Teilgebiet der Informatik anzuerkennen.

In den letzten Jahren hat es sich durchgesetzt, Big Data als neue Disziplin zu verstehen, die sich in erster Linie von traditionellen Business-Intelligence-Ansätzen dadurch abgrenzt, dass in Big Data auch Datenquellen für Analysen ausgewertet werden, die nicht in die traditionelle BI-Welt passen. Wir reden von Datenquellen, die kontinuierlich vorwiegend unstrukturierte (oder schemalose) Daten bereitstellen. Also von Social Networks wie Facebook bis hin zu Sensordaten, die von Maschinen in einer Produktion geliefert werden. Wie aber wird nun Big Data eigentlich definiert?

■ 2.2 Big Data – ein passender Begriff?

In diesem Abschnitt werden zunächst verschiedene Quellen herangezogen, die Big Data aus verschiedenen Perspektiven betrachten. Ohne in die Tiefe zu gehen, lässt sich Big Data so definieren: Big Data sind Datenmengen, die zu groß für traditionelle Datenbanksysteme sind, eine hohe Halbwertszeit besitzen und in ihrer Form nicht den Richtlinien herkömmlicher Datenbanksysteme entsprechen (Dumbill, 2012). Ziel ist es nun, diese Daten dennoch zu speichern und zu verarbeiten, sodass aus ihnen zeitnah wertvolle, neue Informationen gewonnen werden können. Diese neu gewonnenen Informationen können etwa passende Produktempfehlungen in E-Commerce-Lösungen sein, empfohlene Kontakte in sozialen Netzwerken oder Artikelvorschläge auf Nachrichtenseiten. Sind diese Daten nun wirklich *big* gemäß der oben gegebenen Definition?

Auf den ersten Blick wirkt es tatsächlich nicht so, denn Freundschaften können als klassische n:m-Relation gespeichert werden, Artikelvorschläge werden anhand der Tags des gelesenen Artikels erstellt, und Produktempfehlungen entstehen auf Basis der Produktkategorie der vorher betrachteten Artikel. Zieht man aber nun vom Benutzer generierte Inhalte (*User-Generated Content*) hinzu, trifft man auf Inhalte wie:

- Rezensionen und Bewertungsschreiben für Güter und Dienstleistungen
- Foren-Posts und Kommentare
- Pinnwandeinträge

- Blogbeiträge
- (Wissenschaftliche) Artikel
- Tweets

Diese Daten enthalten oft subjektive Meinungen von Benutzern und Konsumenten und sind für Unternehmen, deren Geschäftserfolg vom Verstehen von Wünschen abhängt, dementsprechend wertvoll. Allerdings müssen sie vorher analysiert werden, um sie für Maschinen lesbar zu machen und somit dem Datenhalter einen Mehrwert zu liefern. Diese Datenbeschaffenheit geht mit einem Datenumfang einher, der die traditionelle Datenverarbeitung vor eine scheinbar unlösbare Aufgabe stellt. So sammelt Twitter beispielsweise 8 Terabyte Daten am Tag, Facebook 500 Terabyte, und Google verarbeitet pro Tag etwa 20 Petabyte an User-Generated Content.

Neben den von Menschen generierten Daten gibt es auch maschinengenerierten Daten. Maschinenbauer rüsten ihre Maschinen mit Sensoren aus. Eine Flugzeugturbine liefert 20 Terabytes an Daten pro Stunde (gigaom, 2010). In Produktionslinien, die kontinuierlich Rohstoffe in mehreren Schritten zu Produkten fertigen, enthält jeder Abschnitt eine hohe Zahl an Sensoren, die Druck, Temperatur und weitere Werte messen. Videodaten werden oft für die Auswertung der Qualität verwendet.

2.2.1 Die drei V

Die Frage nach der am meisten verbreiteten Definition von Big Data lässt sich wohl am ehesten durch die drei V beantworten, die der Anbieter von Marktanalysen Gartner 2001 einführte (Lancy, 2001). Zwar wurde hier noch kein Zusammenhang zu Big Data hergestellt, jedoch erkannte man bereits die zukünftigen Herausforderungen der Datenverarbeitung (hier im E-Commerce-Sektor), die den Big-Data-Begriff später prägen sollten.

Volume

Volume (deutsch: Volumen) bezieht sich auf Datenmengen. In den Neunzigern waren die ersten Homecomputer zu Beginn mit 20 Megabyte-Festplatten ausgestattet, heute gibt es mehrere Gigabyte auf USB-Sticks in Fingernagelgröße. Dieses einfache Beispiel spiegelt exponentiell wachsende Datenmengen wider. Dieses Wachstum fasst eine Studie von IDC (Gantz, et al., 2011) passend in Worte:

„Like our physical universe, the digital universe is something to behold – 1.8 trillion gigabytes in 500 quadrillion “files” – and more than doubling every two years.“

Datenmengen wachsen weit schneller als Datenzugriffsraten. Gleichzeitig müssen riesige Datenmengen in möglichst kurzer Zeit durchforstet werden. Festplatten haben oft durchschnittliche Datendurchsatzraten von 100 Mbit/s. Ein komplettes Durchforsten von einem Terabyte auf einem einzigen Speichermedium kann mehrere Stunden dauern. Selbst die schnellsten SSD-Laufwerke können daran nicht viel ändern.

Um die Datenmengen effizient zu verarbeiten, ist es demnach notwendig, die Ausführung zu parallelisieren. Es ist also ersichtlich, dass bei großen Datenmengen die Verarbeitung das *Big* ausmacht (Baron, 2013), nicht der Speicherbedarf (siehe Abschnitt 2.2.3).

Velocity

Das zweite V beschreibt die Geschwindigkeit, in der Daten verarbeitet werden müssen. Ein passendes Beispiel sind etwa Twitter-Meldungen oder Blog-Posts, die zu Zeiten von Wahlkämpfen verfasst werden. Dabei stellt das Internet ein starkes, aber schwer zu kontrollierendes Werkzeug dar. Bekommt ein Politiker beispielsweise eine schlechte Presse, so muss das bereitstehende Marketing-Team entsprechend zügig reagieren und diese durch korrigierende Meldungen relativieren. Die Betonung liegt auf *zügig*, denn ein einziges Gerücht zur falschen Zeit kann zu ungewollten Auswirkungen führen. So sorgte ein am 23. April 2013 geschriebener Tweet über einen Bombenanschlag auf das Weiße Haus für einen Börsencrash an der Wall Street. Bleibt man dem Beispiel der Börse treu, finden sich schnell noch weitere Szenarien, in denen die Geschwindigkeit der Erfassung von Daten eine große Rolle spielt. So sind zum Beispiel Ankündigungen über Firmenübernahmen Gold wert, wenn diese direkt nach der Bekanntgabe verifiziert werden können. Für beide Szenarien gilt: Je schneller die Auswertung stattfindet, desto höher ist der Wert der Information.

Variety

Hier kommt nun die bereits häufig angesprochene Abwesenheit von festen Strukturen und Normalisierungen zur Sprache. Das sicherlich beste Beispiel, um die Datenvielfalt zu beschreiben, mit der wir es zu tun haben, ist das Internet, das – außer vielleicht den Wiki-Seiten – keine feste Struktur vorweisen kann, aber doch einige Ähnlichkeiten aufweist, die eine maschinelle Verarbeitung ermöglichen. So können etwa bei der Analyse des HTML-Codes die Titel-Tags, z. B. `<h1>`, durchsucht werden, um eine thematische Einordnung des Inhalts vorzunehmen. Jedoch sind HTML-Seiten nicht die einzigen Daten, die verarbeitet werden. IBM befragte Mitte 2012 einige Unternehmen, die an einer Big-Data-Initiative teilnahmen, welche Quellen sie für ihre Analysen verwenden (Schroeck, et al., 2012).

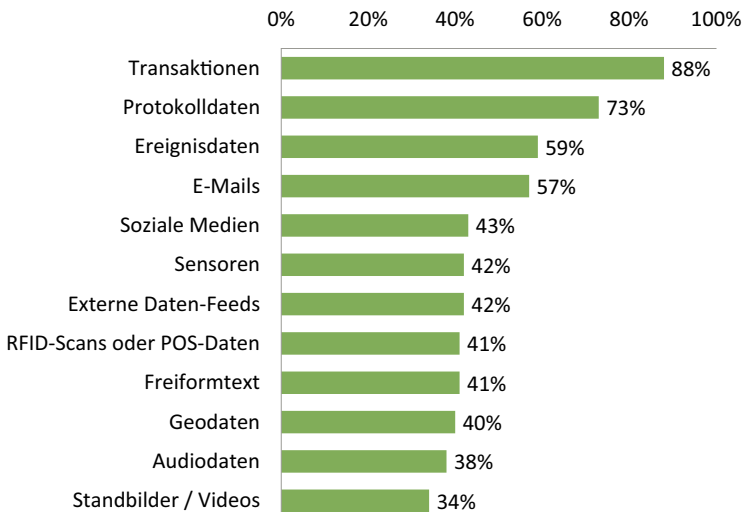


Bild 2.3 Quellen für Big-Data-Analysen auf Basis einer Umfrage von IBM im Jahr 2012

Transaktionsdaten beziehen sich dabei beispielsweise auf den klassischen Börsenhandel, in dem jeder Ver- und Einkauf gespeichert wird. Protokolldaten sind unter anderem Server-

logs, die entsprechend dem Log-Level der jeweiligen Architektur sehr groß ausfallen können und häufig dazu genutzt werden, um Klickpfade durch komplexere Anwendungen zu ermitteln und Benutzer möglichst lange im System zu halten bzw. zu einer bestimmten Aktion (Kauf, Registrierung, Empfehlung) zu bewegen. Ereignisdaten werden etwa in der Automobilindustrie protokolliert, in der Fahrzeugteile produziert, an die Logistik übergeben und verfrachtet werden. Das Feld dieser Daten ist jedoch weder auf die Automobilindustrie noch auf die Produktherstellung im Allgemeinen beschränkt, sondern beschäftigt sich in der IT mit allen Systemen, die bestimmte Ereignisse aufzeichnen und zur Auswertung bereitstellen. So können etwa die Ausschussquote einer Produktreihe überprüft und gegebenenfalls fehlerhafte Teile im gesamten Produktionszyklus, über mehrere Hersteller hinweg, ausfindig gemacht werden. Platz vier auf der Liste belegen E-Mails, die von Mail-Service-Anbietern gescannt, auf Muster von Malware oder Spam durchsucht und für gezielte Produktvorschläge für den jeweiligen Empfänger hin untersucht werden. Ob und wie dieses Vorgehen mit den Datenschutzbestimmungen des jeweiligen Landes und der Moral der Betreiber vereinbar ist, sei einmal dahingestellt. Dass soziale Medien, externe Datenfeeds und Freitextformen noch keine so starke Beachtung finden, mag daran liegen, dass Restriktionen für die Sichtbarkeit von Daten, etwa in Facebook, die Akquise erschweren oder aber, dass die Szenarien für eine Nutzung der Daten noch nicht gefunden sind, um einen produktiven Mehrwert daraus zu ziehen.

Da nun einige Quellen für Big Data vorgestellt wurden, lässt sich auch gleich auf die Vielgestalt der Formatierung eingehen. Neben klassischen, unformatierten Texten kommen häufig JSON (*JavaScript Object Notation*), XML (*Extensible Markup Language*), HTML- oder sogar Byte-Code vor. Gerade wenn man an den Aspekt der Visualisierung denkt, ist es wichtig, Relationen zwischen einzelnen Datensätzen herzustellen, um diese in Abhängigkeit voneinander zu präsentieren. Was in relationalen Datenbanken über simple Queries erreicht werden kann, bedarf bei Plain-Text-Analysen eines erheblichen Aufwands. Viel früher trifft man jedoch bei der Analyse auf die Herausforderung, die gewünschte Information aus jedem einzelnen der vielen Formate herauszufiltern. Des Weiteren gilt zu bedenken, dass sich Formate im Laufe der Zeit auch ändern können. Gerade bei der Auswertung fremder, externer Datenquellen erfolgt meist keine Benachrichtigung über eine Anpassung der Datenstruktur seitens des Datenhalters. Hier ist es wichtig, Auswertungen entsprechend zu überwachen, um Abweichungen frühzeitig festzustellen.

Ist es denn nun gerechtfertigt, von unstrukturierten Daten zu reden? Schließlich weisen ja viele Datensätze eine Struktur auf, nur eben keine feste, einheitliche. Als besserer Begriff wäre hier vielleicht polystrukturiert anzuführen, wie es der Analyst Mike Ferguson in seinem Blog beschreibt (Ferguson). Im späteren Verlauf des Buches, wenn wir zu den Eingabeformaten für die diversen Diagramme kommen, wird sich zeigen, dass diese Vielgestalt einen großen Teil der Arbeit eines Datenanalysten ausmacht, denn die Interpretation der Eingangsdaten einer Analyse variiert nur allzu häufig. Zu diesem Umstand gesellen sich ebenso Fehler in Daten, die schon vor dem Big-Data-Hype bekannt waren. Nathan Yau (Yau, 2010) benennt sechs davon wie folgt:

- Fehlende Werte
- Falsche Beschriftung
- Inkonsistenz
- Tippfehler

- Werte ohne Kontext
- Verteilte Datensätze (über mehrere Quellen hinweg)

Die Komplexität einer Verarbeitung von Daten, die in mehreren, gegebenenfalls unbekannt-ten Strukturen vorliegen und diesen sechs Umständen unterliegen, erfordert also einen erheblichen Mehraufwand gegenüber der Aufbereitung von normalisierten Daten, z. B. aus einer relationalen Datenbank.

2.2.2 Weitere Vs

Es wurde zum Sport von Beratern, weitere Vs auf Präsentationen zu finden, um eine eigene Version der Big-Data-Vs zu generieren.

IBM z. B. führte ein viertes V (Veracity) ein, das die Richtigkeit und die Echtheit von Daten beschreibt (Zikopoulos, et al., 2013). Zwar steigt die Menge an zur Verfügung stehenden Daten nachweislich an, jedoch werden diese häufig durch generierte Inhalte verfälscht, die da sein können:

- Werbung und Spam, die eine einseitige Sicht auf Personen, Produkte oder Vorkommnisse wiedergeben.
- Per Automatismus übersetzte Texte, die häufig grammatikalische, sprachliche und inhaltliche Fehler aufweisen.
- Veraltete oder falsch kategorisierte Suchergebnisse oder Forenindizes.
- Gezielte Falschaussagen oder Fehlinformationen.

Das beste Beispiel sind die klassischen Falschmeldungen, die sich im Internet manchmal in wenigen Minuten verbreiten. So streute etwa ein junger Brite am 26. Februar 2012 das Gerücht, dass der Schauspieler Rowan Atkinson gestorben sei (Gardner, 2012). In nur drei Stunden wurde das Gerücht so schnell verteilt, dass sogar auf Wikipedia der Todestag des Schauspielers eingetragen wurde. Einmal mehr zeigt sich hier die Notwendigkeit, die gesammelten Daten vor der Nutzung zu verifizieren und auszusortieren.

Häufiger als von Veracity wird mittlerweile von Value gesprochen, um zu unterstreichen, dass Big Data erst dann Sinn macht, wenn Machine-Learning-Applikationen Zusammenhänge in den Daten erkennen können.

2.2.3 Der Verarbeitungsaufwand ist big

Ein weiterer interessanter Ansatz, den Aufwand der Verarbeitung großer Datenmengen als *big* zu sehen, liefert der Autor Pavlo Baron:

„Ich hatte z. B. einen Fall, bei dem es um lächerliche Datenmengen ging, die problemlos auf einen USB-Stick gepasst hätten. Man erwartete allerdings simultane Zugriffszahlen im zweifachen Millionenbereich pro Sekunde. [...] das ist definitiv big [...].“ (Baron, 2013)

Es ist also aus dieser Perspektive nicht die bloße Größe der Daten, sondern die Komplexität der Aufbereitung und der Informationsgewinnung. Ein gutes Beispiel sind dafür etwa Video-Streams in Kaufhäusern, die das Kaufverhalten von Kunden auswerten sollen. Auch

wenn eine einstündige Aufnahme lediglich ein paar Hundert Megabyte groß ist, ist die Schwierigkeit der Implementierung und des Trainierens von situationserkennenden Algorithmen sehr hoch und steht beispielsweise im Gegensatz zu einem einfachen Algorithmus, der lediglich alle *<h1>-Tags* aus einigen Millionen HTML-Seiten auslesen muss. Sind die notwendigen Daten extrahiert, müssen gegebenenfalls noch Beziehungen zu anderen Datensätzen hergestellt werden, etwa über einen übereinstimmenden Datumswert, Quellenübereinstimmungen oder, im Optimalfall, über vorliegende IDs. Relationale Daten hingegen verfügen über Schlüsselattribute, die eine Zuordnung von Datensätzen erheblich vereinfachen.

Roger Magoulas von O'Reilly Media gibt eine weitere sehr schöne Definition für Big Data:

„*Big Data ist, wenn die Daten selbst Teil des Problems werden.*“

Dieses Zitat passt besonders gut, da Magoulas gar nicht erst versucht, eine Größendefinition zu geben, sondern einfach sagt, dass man, wenn die Datenmenge für *aktuelle* Verarbeitungsmethoden zu umfangreich wird, von Big Data spricht.

2.2.4 Sicht der Industrie auf Big Data

IBM befragte in der bereits in Abschnitt 2.2.1 erwähnten Studie (Schroeck, et al., 2012) mehrere Unternehmen nach deren Definition von Big Data. Eine Auswertung nach Schlagworten bestätigte weitestgehend die in den vorigen Abschnitten gegebene Sicht auf den neuen Trend. Auffällig ist, dass der Größenbegriff nicht immer im Sinne einer Datengröße verwendet wird. Stattdessen wird etwa in dieser Studie das Schlagwort *Größere Bandbreite an Informationen* verwendet, das sowohl auf große Datenmengen als auch auf mehr oder vielfältigere Informationsquellen hindeutet. 16 Prozent der befragten Unternehmen stellen neue Datenarten und Analysemethoden in den Vordergrund, was wieder für das *big* im Sinne des Verarbeitungsaufwands hindeutet (siehe Abschnitt 2.2.3).

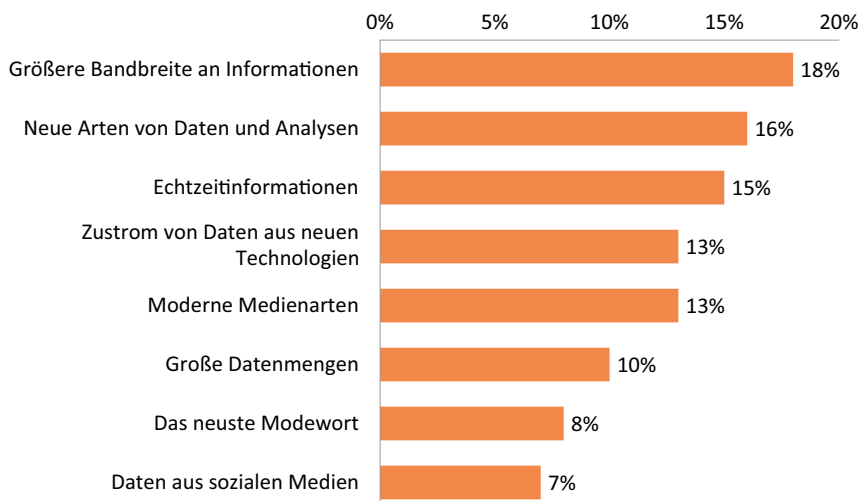


Bild 2.4 Definition von Big Data (Schroeck, et al., 2012)

■ 2.3 Eingliederung in BI und Data Mining

Um die Begriffe BI und Data Mining in Relation zu Big Data setzen zu können, gilt es, diese im Vorfeld zu definieren. Kemper, Mehanna & Unger bezeichnen BI als Filter, der Daten in strukturierte Information umwandle (Kemper, et al., 2010). Gartner hingegen konstatiert etwas ausführlicher, dass BI ein Überbegriff für Anwendungen, Infrastruktur, Werkzeuge und Best Practices sei, die den Zugriff auf und die Analyse von Informationen ermöglichen, um Entscheidungsfindung und Performance zu erhöhen (Gartner). Hält man sich nun strikt an die Definitionen, besteht der Unterschied zwischen BI und Big Data darin, dass BI sich auf bereits vorliegende Informationen bezieht, die dazu noch strukturiert sind und sich auf einen eindeutigen Kontext beziehen.

Das Ziel von BI und der Big-Data-Explorationen ist jedoch dasselbe, nämlich aus vorhandenen Daten neue Erkenntnisse zu gewinnen, die der Entscheidungsfindung bei vorher definierten Fragestellungen dienen. Der Trend bei Big Data geht dabei auch oft in die Richtung zu lernen, was man noch nicht weiß.

BI ist jedoch mittlerweile mehr als diese einfache Begriffsdefinition. Es hat sich in den letzten Jahren zu einem festen Prozess samt einem Set aus technischen Werkzeugen entwickelt, um das Berichtswesen in Unternehmen zu automatisieren. Dazu gehören die Datenaufbereitung, die Datenspeicherung in DWHs sowie deren Darstellung aus verschiedenen Perspektiven.

Welche Techniken, Methoden und Arbeitsschritte werden aber nun angewandt, um Informationen aus vorliegenden Daten zu extrahieren? Die Antwort darauf gibt der sogenannte KDD-Prozess (*Knowledge Discovery in Databases*).

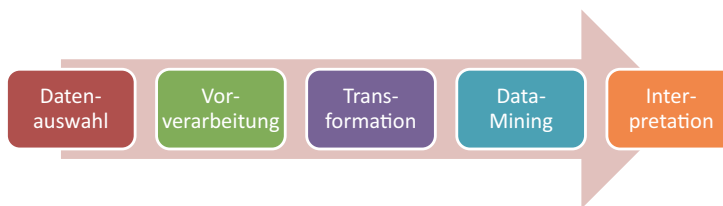


Bild 2.5 Der KDD-Prozess nach (Kononenko, et al., 2007)

Der (iterative und interaktive) KDD-Prozess hat das Ziel, gültige, neue, nützliche und verständliche Muster in vorhandenen Daten zu erkennen (Fayyad, et al., 1996). Wirft man nun einen Blick auf den vierten Schritt des in Bild 2.5 illustrierten Ablaufs, so ist zu erkennen, dass Data Mining einen Teil des KDD-Prozesses darstellt. Dieser nimmt gesäuberte, korrigierte und transformierte Daten entgegen, extrahiert daraus Muster und Zusammenhänge und gibt diese zur Interpretation frei. Quellen müssen, anders als der Begriff KDD vermuten lässt, nicht zwingend Datenbanken sein, sondern können auch als simple Datensätze gesehen werden, z.B. als Flat Files, CSV, XML oder Dateisystemstrukturen. Wichtig ist, dass diese bereits im Vorfeld aufbereitet wurden. Zu dieser Aufbereitung (*Preprocessing*) gehören: