Institut für Parallele und Verteilte Systeme

Universität Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Bachelorarbeit

Anforderungen von Data-Science-Anwendungsfällen im Zonenreferenzmodell

Marc Arthur Altvater

Studiengang:	Data Science B.Sc.
Prüfer/in:	PD Dr. rer. nat. habil. Holger Schwarz
Betreuer/in:	Corinna Giebler, M.Sc., Dipl.-Inf. Michael Behringer
Beginn am:	15. Oktober 2020
Beendet am:	15. April 2021

Kurzfassung

Die Menge an weltweit generierten Daten, sowohl im privaten, als auch vor allem im wirtschaftlichen Umfeld wächst stetig. Daraus entsteht einerseits eine große Nachfrage nach Methoden, um das Volumen der Daten zu verarbeiten und schließlich daraus Wissen zu gewinnen. Andererseits benötigt es auch technologische Konzepte, um diese Daten schnell und kostengünstig speichern zu können. In diesem Bereich hat sich *Data Science* als Wissenschaftszweig etabliert, mit dem Ziel, Methoden aus verschiedenen Bereichen der Mathematik und Informatik zu nutzen, um fundiertes Wissen aus den zugrundeliegenden Daten zu extrahieren. Im Umfeld von *Big Data* gewinnen dagegen *Data Lakes* an Bedeutung, da sie es ermöglichen, (unternehmensweite) Daten zentral zu sammeln und in ihrem rohen, unverarbeiteten Zustand zu speichern. Durch die Erhaltung des ursprünglichen Zustands der Daten werden keine Informationen eingebüßt. Jedoch ist diese Art der Datenhaltung ineffizient.

Aus diesem Grund wurden Zonenmodelle entwickelt, um Daten innerhalb eines Data Lakes in unterschiedliche Bereiche nach ihrem jeweiligen Verarbeitungsgrad zu unterteilen. Data Scientisten werden hierbei oftmals als Experten im Bereich der Datenanalyse einbezogen. Viele der vorhandenen Zonenmodelle stellen ihnen daher isolierte Bereiche für die Erprobung ihrer Methoden zur Verfügung. Jedoch gibt es wenig Forschung, welche die praktischen Anforderungen von Data Scientisten bei der Arbeit mit zonenbasierten Data Lakes betrachtet.

Diese Arbeit befasst sich daher mit der Umsetzbarkeit von Data-Science-Anwendungsfällen in zonenbasierten Data Lakes. Hierfür werden sowohl allgemeine Anforderungen für Data Science, als auch charakteristische Beispiele anhand des Produktlebenszyklus, definiert. Im Anschluss wird daraus ein konzeptionelles Vorgehen für den Prozess der Datenanalyse abgeleitet, welches mittels einer prototypischen Implementierung erprobt wird. Bei der Evaluation der gewonnenen Erkenntnisse wird zudem zwischen der Nutzung einer zonenbasierten und einer zonenlosen Data-Lake-Architektur unterschieden.

Es stellt sich heraus, dass durch die Einführung einer klaren Struktur und der konsequenten Speicherung der Daten in ihren jeweiligen Verarbeitungsgraden zusätzlicher Speicherplatz erforderlich wird. Jedoch profitiert die explorative Datenanalyse signifikant von der Nutzung der vorab berechneten Ergebnisse, was in messbar kürzeren Rechenzeiten resultiert. Dieser Effekt verstärkt sich mit zunehmendem Umfang des Data Lakes und wirkt sich somit auch positiv auf zukünftige Anwendungsfälle aus.

Zudem wird auf die Rollen der unterschiedlichen Data-Science-Spezialisten in der Praxis eingegangen und ein Konzept zur Erweiterung der Zonenmodelle hinsichtlich eines optimierten, kooperativen Ansatzes vorgeschlagen.

Inhaltsverzeichnis

1 Einleitung **13**

2 Grundlagen **15**
2.1 Big Data 15
2.2 Apache Hadoop und Apache Spark 16
2.3 Data Science 17
2.4 Data Lakes 18
2.5 Twitterdaten-Analyse mit Python 21

3 Verwandte Arbeiten **25**

4 Anwendungsfälle **27**
4.1 Anforderungen an Data-Science-Anwendungsfälle 27
4.2 Beispiel Produktlebenszyklus 29

5 Allgemeine Konzepte **33**
5.1 Knowledge Discovery in Databases 33
5.2 KDD-Prozess in Data-Warehouse- und Data Lake-Architekturen 35
5.3 Data-Science-Spezialisten im Zonenreferenzmodell 36
5.4 Leitfaden für die Implementierung von Data-Science-Anwendungsfällen 38

6 Implementierung **41**
6.1 Aufbau der Testarchitektur 41
6.2 Anwendungsfall 1: Marktforschung 43
6.3 Anwendungsfall 2: Marktbeobachtung 49
6.4 Anwendungsfall 3: Prognose zukünftiger Entwicklungen 51

7 Evaluation **53**

8 Zusammenfassung **57**

Literaturverzeichnis **61**

Abbildungsverzeichnis

1 Charakteristiken von Big Data (*„Big V's“*) 16
2 Zonenreferenzmodell nach Giebler et. al. [4] 19

3 Produktlebenszyklus nach Li et. al. [31] mit Zuordnung der betrachteten Anwendungsfälle 30

4 KDD-Prozess nach Fayyad et. al. [33] mit Erweiterung der Data-Science-Fachbereiche 34

5 Verzeichnisstruktur des HDFS zu Beginn der Implementierung (vereinfacht) . . . 42
6 Anzahl Tweets über „Apple“ und deren Polarität 47
7 Anzahl Tweets zu „Macbook“ und deren Polarität 47
8 Verzeichnisstruktur des HDFS nach Abschluss der Marktforschung (vereinfacht) . 48
9 Anzahl Tweets zu den „Top-Companies“ des NASDAQ 50

10 Zonenreferenzmodell mit Transfer-Bereich in der explorativen Zone 56
11 Zonenreferenzmodell mit zusätzlicher Transferzone 56

Tabellenverzeichnis

I Übersicht der Zonen des Zonenreferenzmodells nach Giebler et al. [4] 21

II Übersicht der betrachteten Data-Science-Anwendungsfälle (inkl. praktischer Beispiele) 32

III KDD-Prozess in Data-Warehouse-Systemen und zonenlosen/-basierten Data Lakes 36
IV Arbeitsgebiete der Data-Science-Spezialisten im Zonenreferenzmodell 37
V Anforderungen der Data-Science-Spezialisten an ihre jeweiligen Anwendungsfälle 38

VI Schema der Inputdaten 43
VII Gegenüberstellung zwischen Leitfaden und konkreter Implementierung der Anwendungsfälle 52

VIII Laufzeitanalyse der betrachteten Anwendungsfälle nach Architekturstil des Data Lakes 54
IX Übersicht der aufgewendeten Ressourcen der betrachteten Anwendungsfälle . . . 55

Verzeichnis der Quelltexte

1 Anonymisierung der Nachrichtenverfasser 43
2 Export der anonymisierten Daten in die Rohdatenzone 43
3 Unterteilung des Datensatzes durch Selektion 44
4 Konvertierung der Datumsangabe in menschenlesbares Format 44
5 Extraktion der Nachrichten ohne Verfasser 44
6 Erzeugen eines zusammengesetzten Datensatzes und anschließende Aggregation der monatlichen Tweetzahlen 45
7 Meinungsanalyse der Twitter-Nachrichtentexte 46
8 Extraktion der Nachrichten über „Apple“ sowie mit Inhalt „Macbook“ 46
9 Aggregation der Meinungsanalyse über „Apple“-Nachrichten 46
10 Aufbereitung der Ergebnisse zur Erstellung einer Visualisierung 47
11 Filterung der Nachrichten über „Amazon“ 49

1 Einleitung

In den letzten Jahren ist speziell im industriellen und wirtschaftlichen Umfeld die Menge der laufend generierten Daten signifikant angestiegen [1]. Daraus entstand sowohl die Suche nach Methoden zur effektiven Verarbeitung von großen Datenmengen, als auch ein Bedarf an Konzepten, um diese Daten dauerhaft und kostengünstig in großem Umfang speichern zu können.

Im Bereich der Methoden etablierte sich *Data Science* als Wissenschaftszweig, der sich mit der Datenanalyse und der Extraktion von Wissen beschäftigt. Durch die zunehmende Signifikanz in Wissenschaft und Praxis rückt Data Science aktuell vermehrt in den Fokus, sodass Data Scientist inzwischen oftmals auch als „sexiest job of the 21st century“ [2] bezeichnet wird. Trotz der steigenden Popularität besteht das Ziel von Data Science dennoch im Kern in der Gewinnung von nicht-trivialem Wissen aus der Menge an (heterogenen) Informationen.

Im Bereich der Datenhaltung werden klassischerweise Data-Warehouse-Systeme eingesetzt. Jedoch stoßen diese durch Volumen, Geschwindigkeit und der oftmals fehlenden Struktur der anfallenden Daten heutzutage an ihre Grenzen. Daher gewinnen *Data Lakes* im Umfeld von „Big Data“ seit ihrer Entwicklung im Jahr 2010 [3] zunehmend an Popularität. Sie ermöglichen die Speicherung aller Daten - unabhängig von Format und Struktur - in ihrem rohen, unveränderten Zustand. Dadurch bieten sie die Möglichkeit, jederzeit auf den Ursprungszustand der Daten zuzugreifen, ohne (Meta-)Informationen zu verlieren. Diese Art der Datenhaltung ist jedoch ineffizient. Infolgedessen entstanden Zonenmodelle, welche Daten innerhalb eines Data Lakes in unterschiedliche Bereiche nach ihrem jeweiligen Verarbeitungsgrad unterteilen. Durch die zusätzliche Speicherung von vorverarbeiteten Daten kann eine signifikante Steigerung der Effizienz erreicht werden.

Die Relevanz von Data Scientisten im Umfeld von Big Data wird hierbei hervorgehoben, da sie von den verschiedenen Modellen als Experten für Datenanalyse einbezogen werden. Darüber hinaus stellen ihnen die bestehenden Zonenmodelle oftmals isolierte Bereiche zur Erprobung ihrer Methoden und zur Durchführung aufwändiger Analysen bereit.

Durch das gesteigerte Interesse aus Wirtschaft und Industrie nach derartigen Technologien wurden sie vor allem in Anlehnung an die praktischen Bedürfnisse vorangetrieben und dahingehend entwickelt. Dies führt dazu, dass viele vereinzelte Modelle existieren, jedoch nur wenige Ansätze für ein allgemeines Konzept erarbeitet wurden.

Im Verlauf dieser Ausarbeitung wird daher auf das Zonenreferenzmodell [4] zurückgegriffen, welches durch Giebler et al. systematisch aus existierenden zonenbasierten Modellen hergeleitet wurde. Die beschriebenen Konzepte gelten ebenfalls für eine Vielzahl von weiteren Zonenmodellen und fungieren folglich an verschiedenen Stellen als Bezugspunkt für zonenbasierte Data-Lake-Architekturen.

Die Datenexploration und die Anwendung von Data-Science-Methoden zählen zu den Hauptanwendungsbereichen eines Data Lakes. Daher wird im Rahmen dieser Arbeit untersucht, inwiefern ein zonenbasierter Data Lake in Verbindung mit Data Science verwendet werden kann und welche Vor-/Nachteile sich gegenüber der Nutzung eines zonenlosen Konzepts ergeben.

Der weitere Aufbau der Ausarbeitung setzt sich wie folgt zusammen. Kapitel 2 stellt einen Überblick über die thematischen Grundlagen dar. Dieser wird durch eine Beschreibung verwandter Arbeiten und deren Forschungslücken in Bezug auf Data Science in Kapitel 3 ergänzt. Anschließend beschreibt Kapitel 4 eine Definition allgemeiner Anforderungen an Data-Science-Anwendungsfälle, sowie charakteristischer Praxisbeispiele anhand des Produktlebenszyklus. In Kapitel 5 werden daraufhin allgemeine Konzepte erörtert, welche die Grundlage für die folgende prototypische Implementierung in Kapitel 6 und deren Evaluation in Kapitel 7 bilden. Dieser Abschnitt wird durch eine allgemeine Bewertung des Zonenreferenzmodells hinsichtlich der Umsetzbarkeit von Data-Science-Anwendungsfälle ergänzt. Abschließend fasst Kapitel 8 die Ergebnisse der Arbeit zusammen und bietet einen Ausblick für die Möglichkeiten zukünftiger Forschung.

2 Grundlagen

Dieses Kapitel beschreibt verschiedene Konzepte, deren Verständnis die Grundlage für die folgende Ausarbeitung bildet. Zunächst wird einleitend der Begriff *Big Data* (Abschnitt 2.1), anschließend das verbreitete *Apache Hadoop Framework* und dessen Analyse-Werkzeug *Apache Spark* (Abschnitt 2.2) erläutert. Darauf aufbauend skizziert Abschnitt 2.3 das Berufsbild von *Data Scientisten* in der modernen Wirtschaft.

Abschnitt 2.4 stellt das Konzept von *Data Lakes* vor. Dabei wird auf den Grundgedanken in der Entstehung von Data Lakes eingegangen, sowie deren Potentiale im Vergleich zu klassischen Data-Warehouse-Systemen kurz diskutiert. Nach der Beschreibung des Aufbaus und des *allgemeinen Konzepts* (Abschnitt 2.4.1) von Data Lakes, folgt eine Erläuterung des *Zonenreferenzmodells für Data Lakes* (Abschnitt 2.4.2). Dieses Modell erweitert die bestehenden Data-Lake-Architekturen um ein einheitliches Konzept zur zonenbasierten Datenhaltung.

Abgeschlossen werden die Grundlagen durch eine kurze Einführung in die *Analyse von Twitterdaten mithilfe von Python* in Abschnitt 2.5.

2.1 Big Data

Big Data wird heutzutage in vielen Zusammenhängen als eine Art universeller Begriff genutzt. Ziel dieses Abschnittes ist es deshalb, *Big Data* zu charakterisieren und eine allgemeine Beschreibung zu liefern.

Ausgehend von der Namensgebung liegt es nahe, dass es sich um überaus große Mengen von Daten handelt. Laut Miloslavskaya und Tolstoy [5] zeichnet sich *Big Data* vor allem dadurch aus, dass riesige Datenmengen, oder auch unendliche Datenströme, in beliebiger Form vorliegen. „Beliebige Form“ umfasst hierbei sowohl beliebig strukturierte Daten (strukturierte, semi-strukturierte und unstrukturierte Form), als auch Daten beliebiger Formate (beispielsweise JSON, CSV, PDF, etc.). Diese Fülle übersteigt häufig die Möglichkeiten der traditionellen Datenspeicherung und -verarbeitung, da trotz kontinuierlichem Zufluss an Daten stets eine schnelle Antwortzeit des Systems (oft annähernd Echtzeit) erwartet wird.

McAfee und Brynjolfsson [6] stellen das Potential der Nutzung von Big Data im wirtschaftlichen Umfeld in den Vordergrund. Sie betonen, dass (Management-)Entscheidungen auf Basis einer fundierten Datengrundlage schlicht zu besseren Entscheidungen führen, da Entschlüsse nicht mehr ausschließlich auf der Intuition der jeweiligen Entscheider beruhen.

Sowohl in der Wissenschaft, als auch in der Wirtschaft werden die charakteristischen Eigenschaften von Big Data oft als *„Big V's“* [6] beschrieben. Gemeint sind hiermit prägnante Merkmale, welche in Englisch mit „V“ beginnen. Die Anzahl der *„V's“* kann je nach Fokus der Ausarbeitung variieren, dennoch gibt es einige wiederkehrende, zentrale Kriterien, die regelmäßig in Zusammenhang mit Big Data genannt werden [5, 6, 7, 8]: Volumen (*volume*), Geschwindigkeit (*velocity*) und Vielfalt (*variety*).

McAfee und Brynjolfsson [6] betonen, dass trotz der „Macht von Big Data“ das menschliche Verständnis nötig ist, um die vorliegenden Daten effektiv nutzen und daraus das erhoffte Wissen gewinnen zu können. In diesem Umfeld besteht eine fundamentale Aufgabe von Data Scientisten darin, die Wertigkeit der vorliegenden Daten auszuschöpfen. Diese Wertigkeit der Daten (engl. *„value“*) wird daher häufig als ein weiteres *„V“* von Big Data definiert [5, 6, 7].

Daraus ergibt sich die in Abbildung 1 skizzierte Zusammenstellung der zentralen Charakteristiken von Big Data.

Abbildung 1: Charakteristiken von Big Data (*„Big V's“*)

2.2 Apache Hadoop und Apache Spark

Wie bereits in Abschnitt 2.1 beschrieben, übersteigt das Volumen von Big Data oftmals die Möglichkeiten von klassischen Modellen der Datenverarbeitung. Für diesen Fall wurden spezialisierte Systeme zur Handhabung von Daten dieses Umfangs entwickelt. Ein Beispiel hierfür ist das weit verbreitete Open Source *Hadoop Project*[1] der Apache Foundation. Ein zentraler Vorteil von Hadoop besteht darin, dass auf die Nutzung kostspieliger und spezialisierter (Hochleistungs-)Technik verzichtet werden kann. Stattdessen ermöglicht es die Nutzung beliebiger (Standard-)Hardware zur verteilten Speicherung und Verarbeitung von Daten. Dies senkt die initialen Kosten ungemein und erlaubt somit, in Kombination mit der stufenlosen Skalierbarkeit des Systems, die Arbeit mit Big Data für eine Vielzahl an Unternehmen [9].

Das Hadoop Framework setzt dabei auf ein eigenes Dateisystem zur verteilten Datenspeicherung auf Clustern, das *Hadoop Distributed File System*[2] (HDFS). Eine zentrale Bedeutung nimmt ebenfalls das *MapReduce*-Verfahren [10] ein, da es effiziente, nebenläufige Berechnungen im HDFS ermöglicht. Hierbei werden die benötigten Daten partitioniert und über das Cluster verteilt berechnet. Das HDFS stellt sowohl die Grundlage für den gemeinsamen Datenaustausch dar, als auch die Basis, um die Teilergebnisse im *Reduce*-Schritt [10] wieder zusammenzuführen.

[1] https://hadoop.apache.org/
[2] https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html

HDFS und MapReduce bilden den Kern des Hadoop-Projekts. Zusätzlich existieren weitere Komponenten, welche den Funktionsumfang ergänzen. Hier sind unter anderem HBase, Pig, Hive oder Spark zu nennen [11]. Im Zuge dieser Arbeit wird ausschließlich auf Spark detaillierter eingegangen, da es eine einfache und leistungsfähige Grundlage für die späteren Berechnungen bietet.

Apache Spark[3] verbindet einfach zugängliche Programmierbibliotheken mit mächtigen Datenanalysewerkzeugen. Damit schafft es eine universelle Umgebung, um effiziente Berechnungen auf den umfangreichen Datenmengen in verteilten Hadoop-Clustern zu ermöglichen. Spark erweitert den Funktionsumfang von Hadoop um Möglichkeiten, sowohl große Mengen an ruhenden Daten („data at rest“), als auch den kontinuierlichen Zufluss an Daten einfach und schnell zu verarbeiten. Hierbei nutzt Spark eigene Funktionsbibliotheken, wodurch der aufwändige Entwurf individueller MapReduce-Algorithmen entfällt. Weiterhin ermöglicht Spark die Nutzung von SQL-Funktionen, DataFrame- & Graph-Strukturen, sowie die Anwendung von Machine-Learning-Algorithmen im Big-Data-Umfeld. Da Spark für die parallele Ausführung auf einem (Hadoop-)Cluster spezialisiert und optimiert ist, ergibt sich eine höhere Leistungsfähigkeit bei gleichzeitig geringerer Laufzeit der Algorithmen im Vergleich zur alleinigen Nutzung des MapReduce-Paradigmas.

2.3 Data Science

Durch das stetige Wachstum der weltweit anfallenden Datenmenge [12] steigt ebenfalls das Interesse an Experten, die den versteckten Wert in der Menge an oftmals rohen, heterogen strukturierten Daten herausarbeiten können [2].

Um aus der Masse an Daten gewinnbringende Erkenntnisse ziehen zu können, bedarf es einiger spezifischer Fähigkeiten. Davenport und Patil sehen beispielsweise in Data Scientisten eine seltene Mischung aus Hackern, Analysten, guten Kommunikatoren und vertrauenswürdigen Beratern [2]. Als Ergebnis einer systematischen Auswertung der Fachliteratur beschreiben Schumann et al. [13] ein sehr breites Spektrum an möglichen Fähigkeiten von Data Scientisten, sodass sie ebenfalls als begehrte „Allrounder“ angesehen werden können. Um die gefundenen Kompetenzen zu einem Profil zu bündeln, gruppieren sie diese in soziale, fachliche und selbstständige Bereiche. Sie betonen hierbei jedoch auch, dass alle erforderlichen Kompetenzen praktisch nicht durch eine Person zu verkörpern sind und empfehlen die Entwicklung von individuellen Kompetenzprofilen anhand der konkreten Aufgabengebiete. In der Praxis kann diese Entwicklung bereits beobachtet werden, da sich beispielsweise Abgrenzungen in den Anforderungen zwischen Data Scientists, Data Engineers, Data Analysts, etc. [14, 15] ausprägen.

Data Science definiert sich allerdings nicht ausschließlich über die Fähigkeiten von Data Scientisten im Umgang mit Daten, sondern es schließt auch die dazugehörigen Technologien und Anwendungen ein. Erst dadurch wird es ihnen möglich die Menge der Daten zu „zähmen“ [2] und daraus Erkenntnisse zu gewinnen.

[3] https://spark.apache.org/

Hierzu zählen beispielsweise Programmiersprachen für die einfache und effiziente Verarbeitung von Daten, wie Python[4] und R[5], oder auch die bereits beschriebenen Big-Data-„Werkzeuge“ Hadoop und Spark (Abschnitt 2.2). Eine zunehmende Bedeutung erhalten ebenfalls *Data Lakes*, da sie die immensen Mengen an Daten unterschiedlicher Formate unkompliziert zur Verfügung stellen. Das Konzept von *Data Lakes* wird im folgenden Abschnitt 2.4 erläutert.

2.4 Data Lakes

In Abschnitt 2.2 zu Hadoop und Spark wird beschrieben, wie traditionelle Systeme zur Verwaltung und Verarbeitung von Daten im Umfeld von Big Data an ihre Grenzen stoßen können. Eine Limitierung dieser Systeme besteht darin, dass sie typischerweise ausschließlich mit strukturierten Daten arbeiten. Jedoch sind die anfallenden Daten im Umfeld von Big Data häufig heterogen und besitzen kein einheitliches, vordefiniertes Format. *Data Lakes* hingegen ermöglichen die Speicherung von Daten unterschiedlicher Typen und Formate, da sie in ihrem „natürlichen Zustand“, also im Rohformat, abgelegt werden. Dies war der zentrale Punkt von James Dixon, als er im Jahr 2010 zum ersten Mal das Konzept eines Data Lakes in seinem Blog vorstellte [3]. Durch dieses Konzept findet vorab keine Manipulation statt und die „unterste Ebene der Sichtbarkeit“ bleibt erhalten [3]. Das Prinzip der *„data at rest“* ist von elementarer Bedeutung für das Verständnis von Data Lakes, weshalb Daten stets in ihrem Rohformat erhalten bleiben müssen [16]. Daher dürfen sie nicht verschoben werden, nachdem sie endgültig abgelegt wurden [5]. Dixon wählt den Terminus *Data Lake* hierbei bewusst, da er sich die Menge an verfügbaren Daten als See vorstellt, in den verschiedene Nutzer „eintauchen“ oder auch Proben entnehmen können.

Das Konzept von Data Lakes findet speziell in Bereichen mit hohem Datenaufkommen und erheblichem Informationsgehalt innerhalb der Daten steigendes Interesse. Daher ist es aus wirtschaftlicher Sicht von Vorteil, die unternehmensweiten Daten an einem zentralen Ort sammeln zu können. Traditionell wird dies über große Datenbanksysteme oder Data Warehouses realisiert. Analog zu strukturierten Data Warehouses bündeln Data Lakes viele unternehmensweite Quellen und ermöglichen es dadurch befugten Nutzern, zu jeder Zeit ihre Analysen auf dem gesamten verfügbaren Bestand der Daten ausführen zu können [17].

Die Lagerung der unveränderten Rohdaten bietet zusätzliche Perspektiven für die Datenanalyse. Beispielsweise können somit unkompliziert Analysen für differenzierte Ansichten eines Problems betrachtet werden, indem neu gewonnene Erkenntnisse in Retrospektive mit den historischen Datensätzen verknüpft werden. Diese Möglichkeiten entfallen bei der Nutzung eines klassischen Data-Warehouse-Systems, da hierbei ein anwendungsfallspezifisches Schema bereits beim Import der Datenbasis fest vorgegeben wird. Die dabei verlorenen Informationen können nicht nachträglich wiederhergestellt werden und stehen somit auch für spätere Fragestellungen nicht zur Auswahl.

Im folgenden Abschnitt wird der konzeptionelle Aufbau von Data Lakes geschildert, gefolgt von einer Beschreibung zonenbasierter Data-Lake-Architekturen am Beispiel des Zonenreferenzmodells von Giebler et al. [4]. Dieses Modell stellt die thematische Grundlage für die weitere Ausarbeitung dar.

[4] https://www.python.org/
[5] https://www.r-project.org/

2.4.1 Konzeptioneller Aufbau

Da sie eine große, stetig wachsende Menge an Daten umfassen, sollen Data Lakes konzeptionell analog zu Hadoop durch kostengünstigen Speicher aufgebaut werden. Jedoch stellt Dixon in einer späteren Ausführung klar, dass sie nicht als „Data Warehouses in Hadoop“ zu verstehen sind [18]. Giebler et al. [16] empfehlen deshalb, in Abhängigkeit des jeweiligen Anwendungsfalls und der zugrundeliegenden Daten, angemessene Konzepte beim Aufbau des Data Lakes einzusetzen (beispielsweise *MongoDB*[6] als dokumentenorientierte Datenbank oder *neo4j*[7] als Graphdatenbank). In der Literatur wird darauf hingewiesen, dass für gewisse Anwendungsfälle ebenfalls mehrere solcher Konzepte kombiniert werden können [5]. In der Praxis ist es oftmals unumgänglich verschiedene Konzepte einzusetzen, da sie jeweils spezifische Zwecke erfüllen. Dies gilt beispielsweise auch für die gleichzeitige Nutzung von Data Lakes und Data Warehouses [19, 20].

Durch die praxisorientierte Entwicklung des Data-Lake-Konzepts existiert heutzutage kein verbreitetes, allgemeingültiges Schema zum Aufbau von Data Lakes mit wissenschaftlicher Grundlage. Dies zeigen Giebler et al. nach umfassender Literaturrecherche [16] auf. In ihrer Ausarbeitung sammeln und bewerten sie bisherige Konzepte und praktische Ansätze. Daraus resultieren vier zentrale Aspekte für die Realisierung von Data Lakes:

- Data Lake Architekturen
- Data Lake Modellierung
- Metadaten Management
- Data Lake Verwaltung

Unter Data-Lake-Architekturen fallen grundsätzlich zwei verschiedene Konzepte der Datenhaltung und -organisation innerhalb eines Data Lakes: *Zonenbasierte Architekturen* und *Teich-Architekturen* (engl. *„pond“*). Im Rahmen dieser Arbeit werden ausschließlich zonenbasierte Architekturen betrachtet. Diese werden im nachfolgenden Abschnitt 2.4.2 anhand des *Zonenreferenzmodells für Data-Lake-Management* [4] detailliert besprochen.

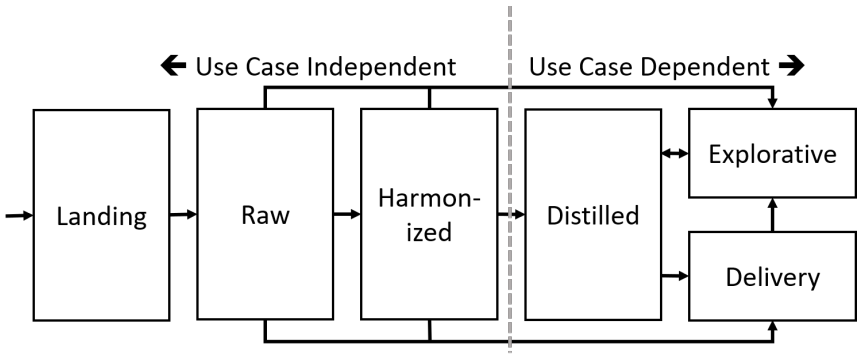

Abbildung 2: Zonenreferenzmodell nach Giebler et. al. [4]

[6] https://www.mongodb.com/de
[7] https://neo4j.com/

2.4.2 Zonenreferenzmodell

Zonenbasierte Modelle zählen zu den zentralen Konzepten der Architektur von Data Lakes. Die Idee, alle eingehenden Daten in ihrer Rohform zu speichern, bietet den Vorteil keine Informationen zu verlieren und zu jeder Zeit auf ihren ursprünglichen Zustand zugreifen zu können. Durch die Nutzung von kostengünstigem Speicher ist dies kein negativer wirtschaftlicher Faktor, die resultierende Ineffizienz durch Wiederholung regelmäßiger Verarbeitungsschritte auf denselben Rohdaten jedoch schon [4].

Das Konzept von zonenbasierten Architekturen behebt diesen Nachteil und kombiniert die positiven Effekte, indem die vorliegenden Daten basierend auf ihren verschiedenen Verarbeitungsstufen gruppiert und strukturiert abgelegt werden. Verschiedene Ansätze hierzu werden beispielsweise von Gorelik [21] oder Ravat und Zhao [22] beschrieben. Somit bleiben die abgelegten Daten dauerhaft und unverändert erhalten, während bereits transformierte oder anderweitig verarbeitete Datensätze ebenso in ihrer neuen Form separat gesichert werden können. Benötigte Daten können dadurch in jedem ihrer Zustände schnell und ohne weitere Berechnungen abgerufen werden - falls sie zu einem früheren Zeitpunkt bereits präpariert wurden. Dementsprechend wird beim Abruf dieser vorab berechneten Datensätze der andernfalls anfallende Rechenaufwand eingespart. Diese Ersparnis summiert sich mit jeder Nutzung weiter auf und demonstriert den Vorteil in der Anwendung von zonenbasierten Architekturen.

Das steigende Interesse für Data Lakes in der Wirtschaft führt zu einer Vielzahl an existierenden individuellen Modellen. Jedoch fehlt es an einem systematischen, allgemeingültigen Konzept für (zonenbasierte) Data-Lake-Architekturen. Giebler et al. [4] widmen sich dieser Thematik und stellen die Eigenschaften verschiedener Konzepte gegenüber, mit dem Ziel ein universelles *Zonenreferenzmodell* für Data-Lake-Architekturen zu entwickeln. Ihr resultierendes Modell ist schematisch in Abbildung 2 skizziert.

Dabei kombinieren sie die Anforderungen an den Funktionsumfang eines Data Lakes (siehe [4]) mit den Erkenntnissen der bestehenden Konzepte (u.a. [21, 22]), um ein allgemeingültiges Modell als praxistaugliche Grundlage zu entwickeln. Das Zonenreferenzmodell umfasst sechs eigenständige Zonen, wovon ausschließlich die *Rohdatenzone* nicht optional ist. Eine Übersicht über die Charakteristiken der einzelnen Zonen ist in Tabelle I dargestellt.

	Landezone	**Rohdatenzone**	**Harmonisierte Zone**	**Reduktions-zone**	**Explorative Zone**	**Auslieferungs-zone**
Original-bezeichnung	Landing Zone	Raw Zone	Harmonized Zone	Distilled Zone	Explorative Zone	Delivery Zone
Eigenschaften	Zwischenstation für eintreffende Daten, Löschung nach Weitergabe an Rohdatenzone, Input-Schema bleibt erhalten	Datengrundlage des Data Lakes, verschiedene Speicher-systeme können kombiniert werden, anonymisierte Personendaten	Kopie/Sicht einer Teilmenge der Rohdaten, umfasst eine Struktur mehrerer (Teil-) Schemas, nachfrage-orientiertes Konzept	Aggregierte und anwendungsfall-spezifische Daten	Testsystem für Data Scientisten, hohe Flexibilität, wenig Vorgaben an die Datencharakteristiken, Löschung von nicht benötigten Daten erlaubt	Schnittstelle zwischen Data Lake und externen Anwendungen, hochgradig anwendungs-fallspezifische Daten, verschiedene Darstellungen eines Datensatzes möglich
Funktionen	Verarbeitung eintreffender Daten(-Ströme), automatisierte Weiterleitung der Daten an die Rohdatenzone, simple syntaktische Anpassungen möglich	Dauerhafte Speicherung aller eingehenden Rohdaten, Löschung generell nur durch rechtliche Vorgaben, Historisierung zur Nachverfolgung von Änderungen	Erstellen einer harmonisierten und konsolidierten Sicht auf die Daten	Datenpräparation zur Effizienzsteigerung, komplexere Änderungen an Semantik/ Syntax möglich	Test beliebiger Analysen und Modelle, Suche nach neuen und gewinnbringenden Erkenntnissen, Zugriff auf Daten aller Zonen für Analysen möglich	Bereitstellung von anwendungsspezifischen Daten für externe Anwendungen
Zugriff	System- und Prozessebene	System- und Prozessebene, Data Scientisten	System- und Prozessebene, Data Scientisten	System- und Prozessebene, Data Scientisten, Domänen-experten	Data Scientisten	System- und Prozessebene, beliebige Nutzer

Tabelle I: Übersicht der Zonen des Zonenreferenzmodells nach Giebler et al. [4]

2.5 Twitterdaten-Analyse mit Python

Angesichts ihrer starken Präsenz im Alltag und der dauerhaft wachsenden Menge an weltweiten generierten Daten stoßen soziale Netzwerke und Mikroblogging-Dienste, wie Twitter[8], in den Bereichen Data Science und Data Analytics auf großes Interesse. Diese Plattformen stellen eine große Anzahl heterogener Daten zur Verfügung, welche sowohl in der Forschung, als auch aus wirtschaftlicher Sicht von Interesse sind (Evaluation von Kundenzufriedenheit & Marketingerfolgen, etc.). In der Literatur beschäftigen sich deshalb verschiedene Ausarbeitungen mit der Extraktion und Auswertung solcher Daten. In diesem Kapitel wird eine kurze Einführung in die Analyse von Twitterdaten gegeben und beschrieben wie deren Verarbeitung mithilfe von Python realisiert werden kann.

[8] https://twitter.com/home

Hierbei wird beispielhaft auf die Ausführung von Wisdom und Gupta [23] eingegangen. Die vorgestellten Vorgehensweisen können jedoch auch allgemeingültig, zum Beispiel durch Nutzung unterschiedlicher Programmbibliotheken, angewendet werden.

Nach Wisdom und Gupta [23] kann die Verarbeitung von Twitterdaten in vier wesentliche Schritte eingeteilt werden:

- Datenbeschaffung
- Datenaufbereitung
- Datenanalyse
- Visualisierung der Ergebnisse

Sie empfehlen zur Realisierung der einzelnen Schritte die Nutzung von vorhandenen Python-Bibliotheken, wodurch viele der benötigten Funktionalitäten unkompliziert abgebildet werden können. Für die Beschaffung der Daten bietet sich beispielsweise die Python Twitter-API *Tweepy* an. Diese ermöglicht die automatisierte Abfrage von Twitterdaten, um sie im Anschluss wahlweise direkt weiterzuverarbeiten oder zwischenzuspeichern.

Zur Vorbereitung der späteren Analysen ist es vorteilhaft den Textanteil der gewonnenen Daten entsprechend aufzubereiten. Dazu bieten sich Methoden aus dem Bereich des Information Retrieval & Text Mining (IRTM) an, beispielsweise die Anwendung von *Tokenization* und die Entfernung häufiger *Stop-Words* [23]. Dieses Vorgehen wird in der Praxis genutzt, um die Effektivität der Analyse von textuellen Daten zu steigern. Dabei werden Zeichenfolgen in Wortblöcke aufgetrennt, um anschließend häufig auftretende Worte mit geringer Aussagekraft zu entfernen. Durch die Reduzierung einer Zeichenfolge auf ihre einzigartigen, definierenden Worte wird der extrahierte Informationsgehalt präzisiert. Für die praktische Anwendung existieren hierfür spezialisierte Funktionen, beispielhaft empfehlen Wisdom und Gupta die *NLTK*-Bibliothek für Python.

Im Bereich der Datenanalyse raten Wisdom und Gupta [23], in Bezug auf die vorliegenden Textdaten, zu einer Kombination aus IRTM-Methoden und weiterführenden Analysefunktionen. So können grundsätzliche Schritte, wie beispielsweise die Häufigkeit bestimmter Terme oder die Generierung von *bigramm*-Termen, ausgeführt werden. Die Generierung von *bigramm*-Termen ist im Bereich von IRTM ein gängiges Verfahren, welches die Zerlegung von Termen innerhalb eines Textes in jeweils zwei aufeinanderfolgende Fragmente beschreibt. Die Untersuchung der Beziehungen zwischen verschiedenen Fragmenten ermöglicht folglich die Analyse ganzer Sätze oder Texte. Auf dieser Basis können daraufhin Analysen mithilfe von Machine Learning und Advanced-Analytics-Methoden erstellt werden.

Je nach betrachtetem Anwendungsfall bieten sich verschiedene Arten zur Visualisierungen der Analyseergebnisse an. Bei der Verarbeitung von geographischen Daten liegt die Nutzung einer Karte als visuelle Darstellung nahe. Wisdom und Gupta [23] wenden in ihrer Vorlage die *geoJSON*-Bibliothek in Verbindung mit der *Leaflet*-Kartenbibliothek an.

Neben der allgemeinen Analyse von Twitterdaten können weitere, ausführlichere Analysen durchgeführt werden. Exemplarisch hierfür wird die Meinungsanalyse von Social-Media-Daten am Beispiel von Twitterdaten kurz erläutert.

Meinungsanalyse von Twitterdaten

Die Meinungsanalyse stellt ein gängiges Mittel in der Auswertung von Social-Media-Daten dar. Sie beschäftigt sich mit der Frage, wie die Ansicht des Verfassers einer Nachricht bezüglich bestimmter Themen kategorisiert werden kann. Gängige Klassen zur Einordnung von Mitteilungen sind beispielsweise [24, 25]:

- positiv,
- negativ, oder
- neutral.

Einige Untersuchungen nutzen darüber hinaus weitere Verfeinerungen der genannten Klassen [26]. Im Umfang dieser Arbeit wird darauf jedoch verzichtet.

In der Literatur thematisieren verschiedene Ausarbeitungen die Kombination der Meinungsanalyse mit zusätzlichen Konzepten. Als Beispiel kann die Ausführung von Mane et al. [24] genannt werden, welche das Hadoop-Framework zur Meinungsanalyse von Twitterdaten nutzt. Dadurch ermöglichen sie die Echtzeitverarbeitung einer großen Menge an Daten. Die Studie von Go et al. [25] kann ebenfalls als Beispiel betrachtet werden, da sie die Klassifizierung von Tweets mithilfe von Machine-Learning-Methoden aus dem Bereich des *supervised learning* darstellen.

Mit den Beschreibungen der Themenbereiche *Big Data*, *Apache Hadoop*, *Data Science*, *Data Lakes* - insbesondere des *Zonenreferenzmodells* - sowie einer kurzen Einführung in die *Analyse von Twitterdaten mit Python* endet das Grundlagenkapitel. Im Folgenden werden verwandte wissenschaftliche Konzepte diskutiert, um dabei aktuelle Forschungslücken zu identifizieren.

3 Verwandte Arbeiten

Im Rahmen dieser Arbeit werden verschiedene Data-Science-Anwendungsfälle in Kombination mit dem Zonenreferenzmodell von Giebler et al. [4] beschrieben, um daraus Anforderungen an das Modell abzuleiten. In diesem Kapitel wird deshalb untersucht, wie zonenbasierte Data Lakes und Data Science in vorhandener Literatur zusammenspielen und welche Möglichkeiten in der bisherigen Architektur ungenutzt bleiben.

Data Lakes sind ein recht junges Konzept, die darauf aufbauenden Zonenmodelle somit ebenfalls. Daher existieren in der wissenschaftlichen Literatur nicht viele Informationen, wie und in welchem Ausmaß Data-Science-Anwendungen effektiv angewandt werden können. Giebler et al. [4] beschreiben bei der Entwicklung ihres Modells an einigen Stellen, welche Funktionalitäten und Möglichkeiten für Data Scientisten als Anwender gegeben sein sollen. Jedoch wird dies im Rahmen der ursprünglichen Definition nicht weiter vertieft.

Bei der Beschreibung ihres Zonenreferenzmodells [4] betonen Giebler et al. etwa die Notwendigkeit für Data Scientisten Daten zurück in den Data Lake schreiben zu können. Dies kann beispielsweise genutzt werden, um gespeicherte historische (Teil-)Ergebnisse für zukünftige Auswertungen (wieder) zu verwenden. Giebler et al. beschreiben dazu anhand ihrer prototypischen Implementierung des Zonenreferenzmodells die Speicherung von aggregierten Werten in der *Reduktionszone*, sowie von berechneten Ergebnissen in der *Auslieferungszone*. Dabei weisen sie auf die Wiederverwendbarkeit dieser Daten bei zukünftigen Auswertungen hin.

Viele existierende Zonenmodelle definieren eine isolierte Zone exklusiv für Data Scientisten. Beispiele hierfür sind die *Transient Zone* bei Madsen [27], die *Sandbox* bei Sharma [28] oder die *Explorative Zone* bei Giebler et al. [4]. In dieser Zone können Analysen frei und flexibel auf beliebigen Daten durchgeführt werden, wodurch eine uneingeschränkte Funktionalität sichergestellt wird. Die vorangehende Beschaffung der Daten, sowie die anschließende Rückführung der gewonnenen Ergebnisse, wird dabei in Verbindung mit den weiteren Zonen realisiert. Der prototypische Anwendungsfall von Giebler et al. verzichtet jedoch auf die Implementierung einer *explorativen Zone*, weshalb die Optionen dieser Zone für Data Scientisten nicht praktisch untersucht werden konnten. Insgesamt gibt es wenig Details zur praktischen Umsetzung in allen Arbeiten. Insbesondere gibt es daher auch keine Informationen über die verschiedenen Nutzergruppen und deren Umgang mit einer solchen Zone.

Ein weiterer Aspekt bei der Anwendung von Data Science in zonenbasierten Data Lakes liegt daher in der Zugriffssteuerung der einzelnen Nutzergruppen auf Zonenebene. Der Zugriff auf die verschiedenen Bereiche des Zonenreferenzmodells wird nach Giebler et al. [4] auf System- und Prozessebene, sowie nach Anwendergruppen aufgeteilt. Bei den Anwendergruppen unterteilen sie hauptsächlich nach Data Scientisten, Domänenexperten und beliebigen Nutzern. Für die Erstellung umfangreicher Auswertungen benötigen Data Scientisten entsprechend erweiterte Zugriffsmöglichkeiten, um die zonenübergreifende Beschaffung einer umfangreichen Datengrundlage zu ermöglichen.

Die Definition der Zugriffsberechtigungen einzelner Nutzergruppen wurde in den Ausführungen der verschiedenen Zonenmodelle jedoch nicht detailliert in Bezug auf Data Science betrachtet. Daher existiert bisher keine Unterteilung der Zugriffs- und Nutzungsmöglichkeiten aus Sicht der spezialisierten *Data Engineers*, *Data Analysts* oder *Data Scientists* (siehe Abschnitt 2.3).

Zusammenfassend lässt sich festhalten, dass existierende Arbeiten das Thema *Data Science* in zonenbasierten Data Lakes aufgreifen, jedoch die vorgestellten Eigenschaften und Ansätze nicht den benötigten Detailgrad zur Verfügung stellen.

Ziel dieser Arbeit ist es daher, an die existierenden Ansätze anzuknüpfen, daraus in Abschnitt 5.4 eine konzeptionelle Vorgehensweise für die Realisierung von verschiedenen Data-Science-Anwendungsfällen zu definieren und diese anhand einer prototypischen Implementierung in Kapitel 7 zu evaluieren. Im Fokus stehen hierbei die Auswirkungen durch die Wiederverwendung von vorab aufbereiteten Daten, sowie von Erkenntnissen aus historischen Analysen. Dies stellt eine zentrale Motivation in der Entwicklung von zonenbasierten Data Lakes dar.

Hiermit schließt der Abschnitt über literarische Grundlagen und offene Fragestellungen existierender Forschung ab. Im Anschluss folgt eine Definition allgemeiner Anforderungen an Data-Science-Anwendungsfälle und die Beschreibung von konkreten, praxisnahen Anwendungsfällen, welche die Grundlage für die anschließende Implementierung bilden.

4 Anwendungsfälle

Das Ziel dieses Kapitels besteht darin, Anforderungen verschiedener Data-Science-Anwendungsfälle im Zonenreferenzmodell zu untersuchen und einen Leitfaden für deren Umsetzung zu schaffen. Hierzu werden zunächst in Abschnitt 4.1 allgemeine Anforderungen an Data-Science-Anwendungsfälle erhoben, um die Vorgehensweise bei der Datenanalyse darzustellen. Anschließend werden in Abschnitt 4.2 charakteristische Anwendungsfälle definiert, welche im Zonenreferenzmodell praktisch umgesetzt werden können. Diese Beispiele sind an eine praxisnahe Anwendung angelehnt und stellen typische Stationen der Produktentwicklung und Marktforschung dar. Ebenso bilden sie die Grundlage für die prototypische Implementierung in Kapitel 6.

4.1 Anforderungen an Data-Science-Anwendungsfälle

Der Bereich Data Science ist stark durch wirtschaftliches Interesse vorangetrieben und geprägt. Infolgedessen wurden bei der Definition allgemeiner Anforderungen und Charakteristiken ebenfalls Einflüsse aus Wissenschaft ([29]) und Wirtschaft ([30]) miteinbezogen.

Durch Bündelung der verschiedenen Einflüsse wurde eine Gruppierung in vier übergeordnete Bereiche mit einzelnen, feineren Unterteilungen erstellt. Diese werden in den folgenden Abschnitten jeweils kurz beschrieben und anschließend in einer Aufzählung zusammengefasst.

1 - Definition von Frage- und Problemstellungen

Zu Beginn einer Data-Sciene-Aufgabenstellung ist es essentiell, die zu untersuchende *Fragestellung des Problems klar zu definieren*. Hieraus gilt es anschließend eindeutige Statements und Hypothesen abzuleiten, an welchen sich der weitere Ablauf des jeweiligen Anwendungsfalls orientieren soll. Da dieser Schritt jedoch nicht durch die Wahl der zugrundeliegenden Architektur beeinflusst wird, fließt diese erste Anforderung nicht in die Bewertung der Implementierung in Bezug auf die Nutzung zonenbasierter Data-Lake-Architekturen ein.

Dennoch kann hierbei erwähnt werden, dass Data Lakes die Möglichkeit bieten, im Verlauf des Projekts die Fragestellung für die jeweiligen Datensätze zu modifizieren. Dies wird durch die Unabhängigkeit ihres konkreten Schemas vom jeweiligen Anwendungsfall und der Speicherung der Daten in ihrem rohen, unveränderten Zustand ermöglicht - im Gegensatz zu den starren, vorab definiterten Strukturen von Data Warehouses.

2 - Bereitstellung qualitativ hochwertiger Daten

Der zweite, große Bereich behandelt die *Bereitstellung von qualitativ hochwertigen Daten* (nach Hardin et al. [29]), um eine fundierte Grundlage für die anschließenden Prozesse zur Analyse und Wissensgewinnung bereitzustellen. Zur weiteren Abstufung wird zwischen der *Datenakquise*, *Datenstrukturierung* und *Daten(vor-)verarbeitung* differenziert.

Die *Datenakquise* umfasst hierbei sowohl die Beschaffung, als auch die sichere und massenhafte Speicherung von Daten (laut Tiedemann [30]). Diese gesicherten Daten gilt es anschließend in einem einheitlichen und gut nutzbaren Schema zu organisieren, was die zentrale Herausforderung bei der *Datenstrukturierung* darstellt. Durch die anschließende *(Vor-)Verarbeitung der vorliegenden Daten* soll ein Verarbeitungsgrad erreicht werden, auf Basis dessen die anschließenden Analysen durchgeführt werden können.

3 - Explorative Datenanalyse

Die *explorative Datenanalyse*, einschließlich der zugehörigen Methoden, fällt in den dritten Anforderungsbereich. In diesem Bereich wird das Ziel verfolgt, in den Daten Muster und Zusammenhänge zu erkennen. Dabei wird das gesammelte Domänenwissen über die Problemstellung in den Analyseprozess eingebracht [29], um zur Studie der Daten Techniken aus dem Bereich der *Advanced Analytics* anzuwenden oder *Machine-Learning-Modelle* zu trainieren.

4 - Wissensgewinnung und Visualisierung

Mit den Ergebnissen der durchgeführten Analysen können im letzten Bereich, der *Wissensgewinnung und Visualisierung*, Antworten auf die anfänglichen Fragestellungen und Hypothesen gegeben werden. Data Science schließt hierbei ebenso die Aufarbeitung und Darstellung des extrahierten Wissens ein. Damit kann auf der abstrahierten Domänenebene (ohne das notwendige technische Verständnis), mithilfe der gewonnenen Erkenntnisse, der (wirtschaftliche) Entscheidungsprozess unterstützt werden.

Die Inhalte der Diskussion zu den allgemeinen Anforderungen und Charakteristiken von typischen Data-Science-Anwendungsfällen können somit übersichtlich zusammengefasst werden:

1 - Definition von Frage- und Problemstellungen

2 - Bereitstellung qualitativ hochwertiger Daten

- **2.1** - Datenakquise
- **2.2** - Datenstrukturierung
- **2.3** - Daten(vor-)verarbeitung

3 - Explorative Datenanalyse

- **3.1** - Anwendung des Domänenwissens
- **3.2** - Advanced Analytics & Machine Learning

4 - Wissensgewinnung und Visualisierung

- **4.1** - Aufarbeitung der Analyseergebnisse
- **4.2** - Unterstützung des Entscheidungsprozesses

In den folgenden Abschnitten werden nun nacheinander die betrachteten Anwendungsfälle skizziert. Diese sind als verständliche und praxisnahe Beispiele entlang des Produktlebenszyklus gewählt.

4.2 Beispiel Produktlebenszyklus

Für die Untersuchung der Anwendbarkeit von Data Science im Zonenreferenzmodell werden drei Anwendungsfälle aus dem Verlauf des Produktlebenszyklus (PLZ) genutzt. Dabei wird Wert auf die Abbildung von praxisnahen Szenarien gelegt. Die hierfür gewählte, einfache Darstellung unterscheidet den PLZ in drei wesentliche Sektoren:

- Beginn der Lebensdauer (BLD)
- Mitte der Lebensdauer (MLD)
- Ende der Lebensdauer (ELD)

Diese Definition wenden ebenfalls Li et al. [31] in ihrer Arbeit zum Umgang mit Big Data im PLZ an. Analog dazu werden Beispiele für Anwendungsfälle von Data Science aus den Sektoren von Beginn und Mitte der Lebensdauer gewählt. Das Ende der Lebensdauer wird von Li et al. [31] überwiegend durch Aufgaben im Bereich Produktrecycling dargestellt. Daher ist der letzte Abschnitt des PLZ im Umfang dieser Arbeit (in Verbindung mit Social-Media-Daten) nicht von Relevanz.

Die folgenden Abschnitte beschreiben drei exemplarische Anwendungsfälle, welche ein breites Spektrum von typischen Praxisbeispielen repräsentieren sollen. Hierbei werden die Bereiche Marktbeobachtung (Abschnitt 4.2.1), Marktforschung (Abschnitt 4.2.2) und die Prognose zukünftiger Entwicklungen (Abschnitt 4.2.3) thematisiert, um zentrale Schritte im Lebenszyklus eines Produktes mit Bezug zur Datenanalyse abzubilden. Die Zuordnung der exemplarischen Anwendungsfälle innerhalb des vereinfachten Produktlebenszyklus nach Li et al. ist in Abbildung 3 dargestellt.

Abbildung 3: Produktlebenszyklus nach Li et. al. [31] mit Zuordnung der betrachteten Anwendungsfälle

4.2.1 Marktbeobachtung

Einer der initialen Schritte des Produktlebenszyklus besteht in der *Marktbeobachtung* während des Beginns der Lebensdauer. Dieser beinhaltet eine Sondierung des Marktes hinsichtlich aktueller Trends, sowie die Identifizierung von Marktlücken mit Erfolg versprechendem wirtschaftlichem Potential. Die Nutzung von Social-Media-Daten ermöglicht dabei eine anhaltende Ansammlung und strukturierte Auswertung aktueller „Trending Topics“. Ein Beispiel hierfür sind die tagesaktuellen *Twitter Trends*[1].

Data Scientisten können im Sinne der Marktbeobachtung beispielsweise die Menge an Nachrichten zu bestimmten Marken analysieren, um daraus den Verlauf aktueller Trends am Markt abzuleiten. Darüber hinaus könnte die Reichweite dieses Verfahrens zusätzlich durch Hinzunahme und Auswertung weiterer Quellen (z.B. News-Feeds) bezüglich ihrer Trends zu den betrachteten Marken erweitert werden. Diese Möglichkeiten werden jedoch aus Gründen des Umfangs im weiteren Verlauf nicht berücksichtigt, da der Fokus auf den Synergien zwischen den verschiedenen Anwendungsfällen bei Nutzung einer zonenbasierten Architektur liegt.

Data Lakes eröffnen dabei die Möglichkeit, die eingehenden Twitter-Nachrichten als Datengrundlage für die Analyse anzusammeln und dauerhaft in ihrem Rohzustand zu speichern, ohne Informationen zu verlieren. Daneben bieten zonenbasierte Data-Lake-Architekturen mit einer *Reduktionszone* für aggregierte Daten einen idealen Bereich zur regelmäßigen Gruppierung der gewonnenen Daten, beispielsweise innerhalb fester zeitlicher Intervalle. Diese konzentrierten Informationen können als Zwischenergebnisse berechnet und abgelegt werden, um eine weitere Ebene der Abstraktion im Verarbeitungsgrad hinzuzugewinnen.

[1] https://twitter.com/i/trends

4.2.2 Marktforschung

Ein charakteristischer Anwendungsfall in der Mitte der Lebensdauer liegt im Bereich der *Marktforschung* und behandelt die Auswertung von Kundenzufriedenheit und Produktperformance [32]. Dies kann gleichzeitig über mehrere Faktoren zum Unternehmenserfolg beitragen, da sowohl Trends bezüglich der Reputation einer Marke festgestellt, als auch der Erfolg und die Qualität einer Produktneuerung gemessen werden können.

Data Scientisten gewinnen hierbei durch Auswertung von Social-Media-Daten ein direktes, unverschleiertes Bild der Kundenmeinung bezüglich eines Produktes oder einer Marke. In Kombination mit der bereits in Abschnitt 2.5 vorgestellten Meinungsanalyse ist es möglich, eine Methode zu entwickeln, um die aktuelle Kundenzufriedenheit eines Produkts am Markt auszuwerten. Der Einsatz eines zonenbasierten Data Lakes kann hier ebenfalls helfen, die generierten Daten strukturiert abzulegen und somit spätere Prozessschritte zu vereinfachen.

4.2.3 Prognose zukünftiger Entwicklungen

Ein weiterer, bedeutender Bereich in der Mitte der Lebensdauer besteht in der Prognose zukünftiger Entwicklungen, basierend auf den historischen Daten und Erkenntnissen vorheriger Produkte. Dieser Lern- und Vorhersageprozess wird heutzutage oftmals über Training und Optimierung von Machine-Learning-Modellen abgebildet und kann für ein breites Spektrum an Themengebieten angewandt werden.

Gängige Anwendungsbeispiele für Data Science sind die Erkennung von Produktfehlern bzw. die Vorhersage der Restlebenszeit eines Produkts im Bereich der Zuverlässigkeitsanalyse. Dabei wird anhand von automatisch erkannten („gelernten“) Mustern in der jeweils vorliegenden Datengrundlage die zukünftige Entwicklung des aktuellen Verlaufs prognostiziert. Dies kann durch die Auswertung von Nutzerberichten hinsichtlich Produktfehlern realisiert werden, um Rückschlüsse über das spezifische Ausfallgeschehen eines Produkts zu ziehen.

Hierbei ermöglichen zonenbasierte Data-Lake-Architekturen verschiedene Vorteile. Zum einen bietet die separierte *explorative Zone* Spielraum für Tests und Analysen, sodass Data Scientisten nicht durch vorgegebene Strukturen oder die Limitationen einzelner Zonen beschränkt sind. Die daraus resultierenden Modelle, sowie das neu gewonnene Wissen über die Daten können anschließend zurück in die *Reduktionszone* innerhalb der Data-Lake-Struktur geschrieben werden. Dadurch wird sichergestellt, dass diese Ergebnisse für zukünftige Analysen wiederverwendet werden können. Ebenfalls ist es möglich, die generierten Resultate über die *Auslieferungszone* bereitzustellen, sodass Domänenexperten direkt in ihren Entscheidungsprozessen auf diese Informationen zugreifen können.

Die Beschreibung der betrachteten Anwendungsfälle wird in Tabelle II übersichtlich zusammengefasst. Hiermit schließt das Kapitel zum Thema Data-Science-Anwendungsfälle und deren Anforderungen anhand praktischer Beispiele ab. Nachfolgend werden allgemeine Konzepte erörtert, welche als Basis für die anschließende Implementierung sowie deren Evaluation dienen.

	Marktbeobachtung	**Marktforschung**	**Prognose zukünftiger Entwicklungen**
Beispielhafter Anwendungsfall	Analyse von Nachrichten über bestimmte Marken, um aktuelle Trends abzuleiten	Meinungsanalyse von Social-Media-Daten zur Auswertung von Kundenzufriedenheit und Produktperformance	Erkennung von Mustern in Produktfehlern, um das spezifische Ausfallgeschehen zu bestimmen
Stellung im Produktlebenszyklus	Beginn der Lebensdauer	Mitte der Lebensdauer	Mitte der Lebensdauer
Beispieldaten aus Social Media	Nachrichten zu bestimmten Themen/Marken, Twitter Trends, News-Feeds	Meinungsanalyse von Nachrichten zu Marken & Produkten	Auswertung von Nutzerberichten über Produktfehler
Vorteile zonenbasierter Data Lakes	Regelmäßige Aggregation in Reduktionszone, Nutzung vorhandener historischer Daten	Regelmäßige Aggregation in Reduktionszone, Nutzung vorhandener historischer Daten	Nutzung vorhandener historischer Daten, unabhängige Analyse in explorativer Zone, direkte Bereitstellung der Resultate über Auslieferungszone

Tabelle II: Übersicht der betrachteten Data-Science-Anwendungsfälle (inkl. praktischer Beispiele)

5 Allgemeine Konzepte

In diesem Kapitel werden allgemeine Hintergründe und Konzepte vorgestellt, welche im Rahmen der Ausarbeitung von Relevanz sind. Hierfür wird in Abschnitt 5.1 der KDD-Prozess [33] als schematische Vorgehensweise zur Wissensgewinnung aus Datenquellen eingeführt und die Verbindung zu Data Science und deren Fachbereiche betrachtet. Im Abschnitt 5.2 folgt eine Gegenüberstellung von Data Lakes und Data Warehouses als Architekturstile für zentralisierte Datenhaltung hinsichtlich ihrer Möglichkeiten zur Datenanalyse. Anschließend befasst sich Abschnitt 5.3 mit den individuellen Anforderungen von Data-Science-Spezialisten im Zonenreferenzmodell und beschreibt einzeln deren unterschiedliche Arbeitsweisen in ihrem jeweiligen Umfeld. Abgeschlossen werden die allgemeinen Konzepte in Abschnitt 5.4 mit der Beschreibung eines Leitfadens für die Implementierung von Data-Science-Anwendungsfällen in zonenbasierten Data Lakes. Dieser bildet die konzeptionelle Grundlage für die anschließende prototypische Implementierung in Kapitel 6.

5.1 Knowledge Discovery in Databases

Das allgemeine Schema zur Wissensgewinnung (in Datenbanken), woran sich auch typische Data-Science-Anwendungsfälle orientieren, wird als *KDD-Prozess* (engl. *„Knowledge Discovery in Databases“*) beschrieben [33]. Es bildet über verschiedenen Stationen die Extraktion von Wissen aus einer zentralen Datenquelle ab. Durch Selektion, Aufbereitung und Transformation der vorliegenden Daten wird die Erkennung von Mustern mittels Data Mining ermöglicht. Die Interpretation dieser entdeckten Muster führt zu neuen Erkenntnissen und weiterführenden Informationen über die zugrundeliegenden Daten. Dieser Prozess von Quelldaten hin zu extrahiertem Wissen wird im oberen Teil in Abbildung 4 abstrakt abgebildet.

Die bereits definierten, typischen Anforderungen an Data-Science-Anwendungsfälle (Abschnitt 4.1) lassen sich ebenfalls anhand dieses Schemas abbilden. Die *Bereitstellung qualitativ hochwertiger Daten* deckt hierbei einen großen Anteil des *KDD-Prozesses* ab. So stellen die Zwischenschritte Datenakquise, Datenstrukturierung und Daten(vor-)verarbeitung die Stationen der Selektion, Aufbereitung und Transformation der Daten dar. Die explorative Datenanalyse wird dabei analog durch die Station des Data Minings im KDD-Prozess beschrieben. Die abschließende Interpretation mit dem Ziel der Wissensgewinnung entspricht schematisch der gleichnamigen Anforderung. Die definierte Anforderung aus Abschnitt 4.1 thematisiert darüber hinaus ebenfalls Schritte zur Visualisierung und Unterstützung von wirtschaftlichen Entscheidungsprozessen.

In Kapitel 3 wurde die Frage eröffnet, inwiefern sich die Vorgehensweise bei der Datenanalyse unterscheidet, wenn zwischen den verschiedenen Data-Science-Spezialisten differenziert wird. Dabei kann es von Interesse sein, das schematische Konzept des KDD-Prozesses hinsichtlich der Aufgabengebiete von Data-Science-Spezialisten zu untersuchen (siehe Abschnitt 2.3). Somit lässt sich bestimmen, ob gegebenenfalls eine Unterteilung des Schemas in mehrere Teilabschnitte anhand der jeweiligen Fachbereiche möglich ist.

Durch ihren starken Fokus auf die Beschaffung und Organisation von Datenquellen wirken Data Engineers überwiegend im Bereich der Selektion und Aufbereitung von benötigten Daten. Darüber hinaus sind sie oftmals für die Bereitstellung von bereinigten oder aggregierten Daten zuständig, welche für die weiterführenden Analysen der spezialisierten Data Scientisten herangezogen werden können. Die Zuordnung der spezialisierten Data Scientisten ist weniger trivial, da ihre Aufgabengebiete über die gesamte Kette der Datenverarbeitung zu finden sein können. In Abhängigkeit von den jeweiligen Anforderungen konzentrieren sie sich in der Regel auf die detaillierte Analyse versteckter Muster innerhalb der vorliegenden Daten. Die Interpretation der gefundenen Muster wird dabei oftmals durch Data Analysts durchgeführt, da sie eine stärkere Fokussierung auf die wirtschaftlichen Zusammenhänge in der zugehörigen Domäne besitzen.

Eine mögliche Unterteilung der Stationen des KDD-Prozesses wird anhand der beschriebenen, typischen Aufgabengebiete der einzelnen Spezialisten im unteren Bereich von Abbildung 4 dargestellt.

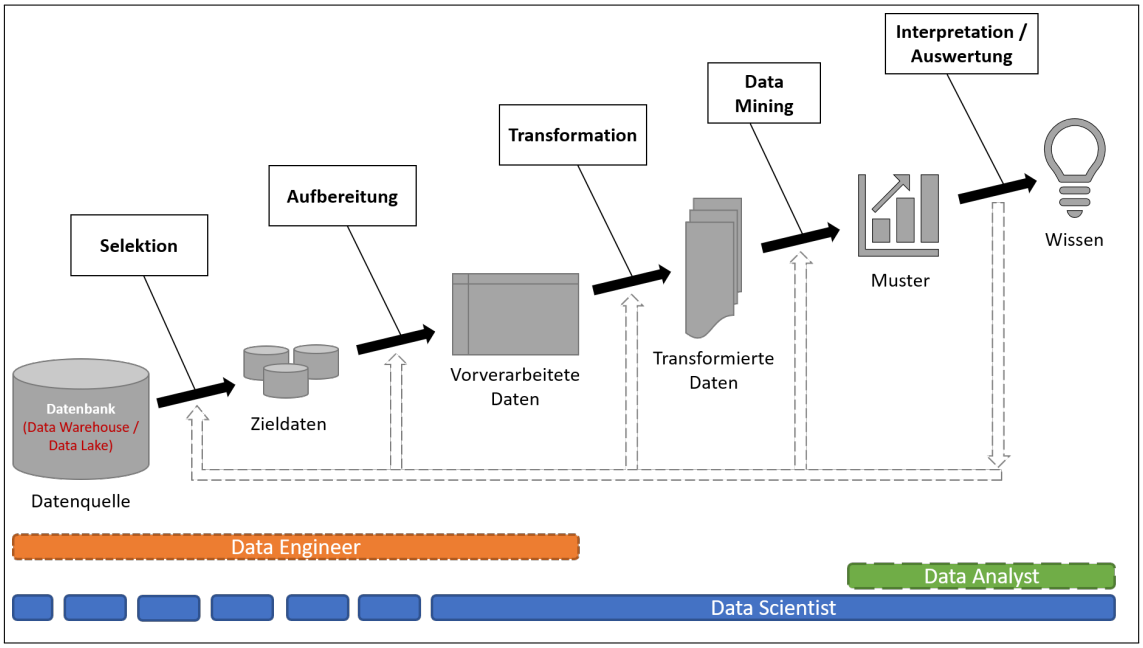

Abbildung 4: KDD-Prozess nach Fayyad et. al. [33] mit Erweiterung der Data-Science-Fachbereiche

5.2 KDD-Prozess in Data-Warehouse- und Data Lake-Architekturen

Um die Möglichkeiten von Data-Science-Anwendungsfällen im Zonenreferenzmodell fundiert bewerten zu können, wird im folgenden Abschnitt diskutiert, in welchem Umfang die Stationen des KDD-Prozesses in den verschiedenen Architekturen (Data-Warehouse-Systeme, allgemeine Data Lakes, zonenbasierte Data Lakes) realisiert werden können und inwiefern diese durch die jeweiligen Konzepte profitieren.

Sowohl Data Warehouses, als auch Data Lakes zeichnen sich dadurch aus, dass sie als unternehmensweite, zentrale Datenbasis für die Datenanalyse dienen. Dennoch unterscheiden sie sich bereits fundamental in ihrem Einsatzzweck: Data Lakes sammeln alle Arten von Daten in ihrem rohem Zustand, während Data Warehouses ein vorab definiertes Zielschema besitzen. Die *Selektion* und Extraktion von Daten steht bei beiden Architekturen im Mittelpunkt, jedoch unterscheiden sie sich bereits im Schritt der *Datenaufbereitung*.

Im Data Warehouse wird schon während des Imports neuer Datenquellen das vorgegebene Zielschema angewandt. Somit ist eine nachgelagerte Änderung der Struktur der benötigten Daten nicht ohne aufwändige Anpassungen der Extraktionsschritte möglich. Data Lakes hingegen erfordern durch die Lagerung von Rohdaten (ohne einheitliches Format oder Struktur) eine wiederholte Aufbereitung der Datensätze für jede anschließende Nutzung. Bei der Unterscheidung zwischen Data-Lake-Architekturen mit und ohne Zonen-Strukturen, lässt sich jedoch anmerken, dass durch die Anwendung von zonenbasierten Architekturen (wie beschrieben durch das Zonenreferenzmodell in Abschnitt 2.4.2) redundante Bereinigungsschritte mittels separater Speicherung der Daten in ihren einzelnen Verarbeitungsschritten ersetzt werden können. Weitere Vorteile dieser Methodik sind im Grundlagenkapitel in Abschnitt 2.4 erläutert.

Die Schritte zur *Datentransformation* können sowohl durch Data-Warehouse-Systeme, als auch durch Data Lakes abgebildet werden, da sie essentielle Vorgänge im Datenanalyseprozess darstellen. *Data Marts* im Bereich der Core-Data-Warehouses beinhalten transformierte Daten und fungieren als Schnittstelle für die weiterführenden Prozesse. In einer strikt zonenlosen Data-Lake-Struktur existiert wiederum kein separater Bereich für die Lagerung aufbereiteter Daten. Deshalb können diese im Verlauf der Datenanalyse entweder nicht zwischengespeichert werden, oder durch unstrukturierte Speicherung innerhalb des Data Lakes zu einer chaotischen Datenhaltung führen. Mittels einer zonenbasierten Data-Lake-Architektur wird diesen Problemen entgegengewirkt, indem isolierte Bereiche zur strukturierten Speicherung von aufbereiteten Daten eingeführt werden (beispielsweise die *Reduktionszone* des Zonenreferenzmodells). Diese bilden analog zu Data Marts die Grundlage für die folgenden Prozessschritte.

Die Schritte zu *Data Mining* und der *Interpretation* der gefundenen Ergebnisse stellen im KDD-Prozess die zentralen Schritte der Datenanalyse dar. Sie können ebenfalls in jeder der genannten Architekturen abgebildet werden. Hierfür bieten Data-Warehouse-Systeme die Unterstützung für OLAP- & Data-Mining-Methoden an, welche auf den bereitgestellten Daten der Data Marts aufbauen und die anschließende Interpretation der Resultate durch die jeweiligen Nutzer ermöglichen. In zonenlosen Data Lakes wird auf den Zwischenergebnissen der Datentransformation aufgebaut. Jedoch kann die Beschaffung dieser Daten durch die fehlende Struktur in der Datenhaltung erschwert oder eingeschränkt sein. Hier eröffnet ein zonenbasierter Data Lake weitere Vorteile, da die benötigten Zwischenergebnisse strukturiert nach ihrem Verarbeitungsgrad abgelegt und damit leicht zu beschaffen sind.

Zusätzlich bieten viele Zonenmodelle einen isolierten Bereich für die Datenanalyse durch Data Scientisten an (wie in Kapitel 3 beschrieben). Die resultierenden Ergebnisse können den jeweiligen Nutzern außerhalb des Data Lakes über eine Schnittstelle (beispielsweise die *Auslieferungszone*) zur Verfügung gestellt werden, um eine Interpretation im jeweiligen Umfeld zu ermöglichen.

Die Diskussion hinsichtlich der Umsetzbarkeit des KDD-Prozesses anhand der unterschiedlichen zugrundeliegenden Architekturen wird in Tabelle III zusammengefasst. Volle Funktionalität eines Teilprozesses wird hierbei durch „✓“ abgebildet, eingeschränkte Funktionalität durch „(✓)“.

KDD-Prozess	Data Warehouse	Data Lake (ohne Zonen)	Zonenbasierter Data Lake
Selektion	✓	✓	✓
Aufbereitung	(✓)	(✓)	✓
Transformation	✓	(✓)	✓
Data Mining	✓	(✓)	✓
Interpretation	✓	✓	✓

Tabelle III: KDD-Prozess in Data-Warehouse-Systemen und zonenlosen/-basierten Data Lakes
volle Funktionalität ≙ „✓“, eingeschränkte Funktionalität ≙ „(✓)“, fehlende Funktionalität ≙ „ - “

5.3 Data-Science-Spezialisten im Zonenreferenzmodell

Um ein besseres Verständnis für die verschiedenen Tätigkeiten von Data Scientisten im Umfeld von zonenbasierten Data Lakes zu erhalten, werden im folgenden Abschnitt die Aufgabengebiete der einzelnen Data-Science-Spezialisten betrachtet. Hierfür werden die Überlegungen aus Abschnitt 5.1 und Abschnitt 5.2 mit der Definition des Zonenreferenzmodells hinsichtlich der Zugriffsverwaltung einzelner Nutzergruppen aus Abschnitt 2.4.2 kombiniert. Auf dieser Basis kann das Zonenmodell in Bezug auf die Arbeitsgebiete von Data-Science-Spezialisten untersucht werden. Dies wird abschließend in Tabelle IV zusammengefasst.

Aus den unterschiedlichen Herangehensweisen und Aufgabenstellungen lassen sich ebenfalls individuelle Anforderungen für die einzelnen Gruppen ableiten, was in Tabelle V dargestellt wird. In den folgenden Abschnitten wird auf die einzelnen Experten im Detail eingegangen, um daraus unterschiedlichen Anforderungen hinsichtlich ihrer jeweiligen Rolle im Umgang mit einem zonenbasierten Data Lake abzuleiten.

Data Scientisten, welche sich auf Datentransformation und Data Mining konzentrieren, verfolgen den wissenschaftlichsten Ansatz unter den verschiedenen Ausprägungen. Sie befassen sich hauptsächlich mit der Datenanalyse und der Suche nach nicht-trivialen Mustern in Datensätzen. Zu diesem Zweck existiert im Zonenreferenzmodell ein separater Bereich - die *explorative Zone* - exklusiv für Data Scientisten. Darin können sie ohne Einschränkungen Machine-Learning-Modelle testen, verschiedenste (über den Data Lake verteilte) Datensätze miteinander verknüpfen oder weitere unterschiedliche Arten von Auswertungen erstellen. Zur Sicherstellung bestmöglicher, umfassender Analysen werden Data Scientisten daher erweiterte Freiheiten und Zugriffsmöglichkeiten eingeräumt, wodurch sie über fast alle Zonen des Data Lakes verfügen können.

Zonenreferenzmodell	Data Scientist	Data Engineer	Data Analyst
Landezone	-	(✓)	-
Rohdatenzone	(✓)	✓	-
Harmonisierte Zone	✓	✓	-
Reduktionszone	✓	(✓)	✓
Explorative Zone	✓	-	-
Auslieferungszone	✓	(✓)	✓

Tabelle IV: Arbeitsgebiete der Data-Science-Spezialisten im Zonenreferenzmodell
Hauptarbeitsgebiet ≙ „✓“, vereinzelte Tätigkeiten ≙ „(✓)“, keine Tätigkeiten ≙ „ - “

Data Scientisten kombinieren diese umfassenden Zugriffsmöglichkeiten mit dem Ziel der detaillierten Datenanalyse. Dadurch können sie ein unverfälschtes Resultat sicherstellen, sowie alle definierten Anforderungen aus Abschnitt 4.1 bei der Bearbeitung ihrer jeweiligen Anwendungsfälle beachten. Dies variiert jedoch bei Betrachtung der beiden weiteren Expertenrollen.

Die Rolle von *Data Engineers* kann hierbei in zwei Fälle unterteilt werden: Einen systemseitigen und einen datenfokussierten Einsatzzweck. Data Engineers können einerseits durch ihre Erfahrungen auf Systemebene, sowie der Tätigkeit bei der Beschaffung von Datenströmen, beim Aufbau eines Data Lakes involviert sein und somit auch Zugriff auf die Lande- und Auslieferungszone besitzen. Hierbei kann es beispielsweise in ihrem Aufgabenbereich liegen, sicherzustellen, dass das System zur automatisierten Verarbeitung eintreffender Daten(-ströme) korrekt arbeitet und die weitergeleiteten Daten an die Rohdatenzone liefert. Dafür benötigen sie jedoch unabhängig vom jeweiligen Anwendungsfall weitere Anforderungen auf Sicherheits- und Systemebene.

Andererseits kann die Tätigkeit von Data Engineers auch darin bestehen, vorverarbeitete Daten für die Analysen von Data Scientisten bereitzustellen. Hierfür arbeiten sie hauptsächlich in der Rohdatenzone (zur Sammlung der benötigten Daten), der harmonisierten Zone (zur Erstellungen einer einheitlichen Struktur der Daten) und - je nach Aufgabe - auch in der Reduktionszone (zur Bereitstellung von vorab aggregierten Daten). Daher liegt bei der Auswahl der nötigen Anforderungen der Fokus auf den Bereichen zum Verständnis der aktuellen Fragestellung und der Bereitstellung qualitativ hochwertiger Daten (einschließlich der Datenakquise, Datenstrukturierung und der Datenverarbeitung).

Wie in Abschnitt 5.1 beschrieben konzentrieren sich *Data Analysts* auf die Interpretation der gefundenen Muster und Resultate von Data Scientisten. Sie greifen auf ihr ausgeprägtes Domänenwissen zurück, um die Erkenntnisse der vorliegenden Auswertungen mit den wirtschaftlichen Vorgängen zu verbinden. Dabei erstellen sie verständliche Handlungsempfehlungen und Entscheidungsgrundlagen, welche sie anschließend der Managementebene präsentieren. Hierfür arbeiten sie hauptsächlich mit den vorliegenden Ergebnissen in der Auslieferungszone oder bei Bedarf auch mit aggregierten Daten aus der Reduktionszone. Ein tieferes Verständnis für die Vorgänge während der Analysen oder dem Aufbau der Datenstruktur des Data Lakes werden daher nicht benötigt. Somit leiten sich hauptsächlich Anforderungen bezüglich der Frage- & Problemstellung, der Anwendung des Domänenwissens und der Aufarbeitung der Analyseergebnisse zur Unterstützung von Entscheidungsprozessen ab.

Anforderungen	Data Scientist	Data Engineer	Data Analyst
Definition von Frage- & Problemstellungen	✓	✓	✓
Bereitstellung qualitativ hochwertiger Daten	(✓)	✓	-
- Datenakquise	(✓)	✓	-
- Datenstrukturierung	(✓)	✓	-
- Daten(vor-)verarbeitung	(✓)	✓	-
Explorative Datenanalyse	✓	-	-
- Anwendung des Domänenwissens	✓	-	✓
- Advanced Analytics & Machine Learning	✓	-	-
Wissensgewinnung und Visualisierung	✓	-	✓
- Aufarbeitung der Analyseergebnisse	✓	-	✓
- Unterstützung des Entscheidungsprozesses	(✓)	-	✓

Tabelle V: Anforderungen der Data-Science-Spezialisten an ihre jeweiligen Anwendungsfälle
Anforderung: relevant ≙ „✓“, teilweise relevant ≙ „(✓)“, nicht relevant ≙ „ - “

5.4 Leitfaden für die Implementierung von Data-Science-Anwendungsfällen

Bei genauerer Betrachtung der verschiedenen Zonenmodelle für Data Lakes ([4], [21], [22], etc.) orientiert sich die Aufteilung der einzelnen Zonen aus funktionaler Sicht an den allgemeinen Anforderungen an Data-Science-Anwendungsfälle aus Abschnitt 4.1. Dies kann durch den Einfluss der praktischen wirtschaftlichen Abläufe bei der Entstehung der Modelle begründet werden.

In diesem Kapitel wird daher ein Konzept diskutiert, welches als Leitfaden für die Umsetzung von Data-Science-Anwendungsfällen in zonenbasierten Data Lakes herangezogen werden kann. Dabei wird aufgezeigt, wie die gestellten Anforderungen durch Anwendung des Zonenreferenzmodells von Giebler et al. aus Abschnitt 2.4.2 adressiert werden können. Die nächsten Abschnitte diskutieren daher die Anforderungen aus Abschnitt 4.1 und beschreiben daraufhin Vorgehensweisen für mögliche Umsetzungen.

Die *Definition von Frage- und Problemstellungen* (Anforderungsbereich 1) besitzt keinen direkten Einfluss auf die Wahl der zugrundeliegenden Big-Data-Architektur. Jedoch können die notwendigen Schritte des Datenanalyseprozesses durch Vorüberlegungen zur Frage-/Problemstellung selektiert und je nach Anforderungen auch reduziert werden. Dies kann beispielweise durch Wiederverwertung von bereits bestehenden Daten geschehen oder durch den Wegfall von nicht relevanten Prozessen, falls beispielsweise in einem konkretem Fall keine visuelle Aufbereitung der Ergebnisse benötigt wird.

Die *Bereitstellung qualitativ hochwertiger Daten* wird durch die jeweiligen Zonen abgebildet, da diese die verschiedenen Grade in der Verarbeitung der Daten abbilden. So beinhaltet der Prozess der *Datenakquise* beispielsweise den initialen Schritt der Ankunft der Daten in der Landezone. Hierbei werden direkt eventuelle syntaktische Anpassungen zur Erfüllung von rechtlichen und ethischen Vorgaben zum Datenschutz eingeschlossen. Dieser Prozessschritt wird im Anschluss durch die Weiterleitung der Daten, zur dauerhaften Speicherung in der Rohdatenzone, abgeschlossen.

Ein zentraler Aspekt bei der Datenanalyse ist die Aufbereitung der Daten, was die *Strukturierung* und *(Vor-)Verarbeitung* einschließt. Im Zuge der *Strukturierung* werden die vorliegenden Daten in der harmonisierten Zone in ein einheitliches Format überführt, um anschließend eine einfach handzuhabende Grundlage für die weiteren Schritte zu schaffen. Auf dieser Basis kann die anschließende *Weiterverarbeitung* durchgeführt werden. Sie umfasst insbesondere die *Vorverarbeitung und Aggregation*, also die Zusammenführung und Gruppierung der Daten anhand bestimmter Kriterien. Die Anforderungen für die Aggregationsschritte variieren spezifisch nach Anwendungsfall. Die resultierenden Datensätze werden nach Abschluss zur späteren Nutzung gesammelt in der Reduktionszone abgelegt.

Ein zentraler Bereich von Data Science wird durch die Anforderungen an die *explorative Datenanalyse* beschrieben. Bei der Umsetzung des jeweiligen Anwendungsfalls ist es essentiell, durch die *Anwendung des Domänenwissens* ein Verständnis der Problemstellung zu erhalten. Somit kann die Vorgehensweise für die Methoden der *Advanced Analytics und Machine Learning* koordiniert werden. Dies kann beispielsweise durch die Untersuchung von (für die Fragestellung) relevanten, aggregierten Daten geschehen, um in der explorativen Zone des Data Lakes aussagekräftige Analysen und Auswertungen erstellen zu können.

Die daraus entstandenen Resultate können anschließend entweder für zukünftige Zwecke in der Reduktionszone gesichert werden, oder auch für die folgende *Aufarbeitung der Analyseergebnisse* in die Auslieferungszone verschoben werden. Dort wird durch die Interpretation der generierten Informationen *Wissen gewonnen*. Dies stellt somit den vierten Bereich der Anforderungen an Data-Science-Anwendungsfälle dar.

Abgeschlossen wird die Aufarbeitung der Daten durch die *Unterstützung des Entscheidungsprozesses*. Häufig geschieht dies mithilfe einer geeigneten *Visualisierung* und der Übergabe von abstrahierten Daten für technik- oder domänenfremde Gremien und Kunden. Im finalen Schritt werden die Resultate über die Auslieferungszone des Data Lakes (als Schnittstelle zur Außenwelt) zur Verfügung gestellt.

Damit schließt die Diskussion über die definierten Anforderungen aus Abschnitt 4.1 und ihren konzeptionellen Möglichkeiten in der Implementierung, bei Nutzung eines zonenbasierten Data Lakes, ab. Hierbei wurde insbesondere auf die Analogien zwischen dem schematischen Aufbau des Zonenkonzepts und den allgemeinen Anforderungen an Data-Science-Anwendungsfälle eingegangen. Inwiefern sich in der Praxis bei der Umsetzung von Data-Science-Anwendungsfällen jedoch Vorteile durch zonenbasierte Data Lakes gegenüber einer zonenlosen Architektur ergeben, ist bisher noch unklar.

Um die Anwendbarkeit dieses Konzepts zu demonstrieren und eine Unterscheidung zwischen den betrachteten Architekturstilen von Data Lakes zu untersuchen, wird im folgenden Kapitel die praktische Umsetzbarkeit anhand einer prototypischen Implementierung betrachtet.

6 Implementierung

Im vorherigen Abschnitt wurde ein konzeptioneller Ansatz diskutiert, welcher die Umsetzung von Data-Science-Anwendungsfällen mithilfe von zonenbasierten Data Lakes behandelt. Die dargestellten Abläufe sollen nun im Rahmen einer prototypischen Implementierung erprobt werden, um Wissen über die praktische Anwendbarkeit zu gewinnen. Hierfür wird zuerst die genutzte Testarchitektur in Abschnitt 6.1 beschrieben, gefolgt von der konkreten Implementierung der Anwendungsfälle aus Kapitel 4.

6.1 Aufbau der Testarchitektur

Zur Simulation einer Big-Data-Umgebung wurde lokal ein HDFS (Hadoop-Version 3.2.1, siehe Abschnitt 2.2) aufgesetzt. Durch die Limitationen der eingesetzten Hardware besteht das Hadoop-Cluster aus einer lokalen Instanz mit einem einzelnen Knoten. Dieser stellt die Nutzung von 8 CPU-Kernen, 2GB Hauptspeicher und einem separatem Festplattenlaufwerk mit 256GB Speicher bereit. Im Rahmen dieser Implementierung wird stellvertretend für zonenbasierte Data Lakes das Zonenreferenzmodell von Giebler et al. [4] zugrunde gelegt. Daher wurden die Verzeichnisse im HDFS analog zu den definierten Zonen des Modells gewählt. Dies ist in Abbildung 5 dargestellt. Durch Änderungen an der Verzeichnisstruktur kann diese Herangehensweise unkompliziert auf weitere Zonenmodelle übertragen werden. Somit ist eine Allgemeingültigkeit gewährleistet. Für den Vergleich der unterschiedlichen Architekturstile von Data Lakes wurde ein Verzeichnis „XX_Zoneless“ erstellt, welches einen Data Lake mit zonenloser Architektur simuliert.

Zur Durchführung des Datenanalyseprozesses wird die Data-Science-typische Anwendung *JupyterLab*[1] in Verbindung mit *PySpark*[2] als Python-Schnittstelle zu Spark (Spark-Version 3.1.1, siehe Abschnitt 2.2) genutzt. Sie stellen verbreitete Werkzeuge bei der Softwareentwicklung im Data-Science-Umfeld dar und ermöglichen eine einfache, effiziente Verarbeitung großer Datenmengen mithilfe von Python (Version 3.7.0). Hierdurch bieten sich umfassende Möglichkeiten für Data Science im Kontext von Data Lakes und anderen Big-Data-Strukturen.

Um einen replizierbaren Ablauf sowie eine identische Datengrundlage zwischen den unterschiedlichen Architekturen sicherzustellen, wird in der Implementierung auf die Nutzung der Twitter-API verzichtet. Anstelle dessen dienen Datensätze zum Thema „Tweets über die Top-Unternehmen des NASDAQ von 2015 bis 2020“ als Basis. Diese sind frei verfügbar auf der bekannten Data-Science-Plattform *kaggle*[3] und werden initial in der Landezone des zonenbasierten Data Lakes bzw. direkt im Verzeichnis des zonenlosen Data Lakes abgelegt.

[1] https://jupyter.org
[2] https://spark.apache.org/docs/latest/api/python/index.html
[3] https://www.kaggle.com/omermetinn/tweets-about-the-top-companies-from-2015-to-2020

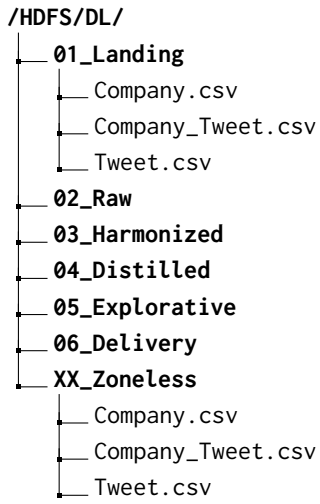

Abbildung 5: Verzeichnisstruktur des HDFS zu Beginn der Implementierung (vereinfacht)

Auf dieser Grundlage wird die praktische Umsetzung der Anwendungsfälle aus Kapitel 4 betrachtet. Aus Gründen des Umfangs werden die Beispiele aus den Bereichen *Marktbeobachtung* und *Marktforschung* konkret realisiert. Die Umsetzbarkeit der *Prognose zukünftiger Entwicklungen* wird auf Basis der gewonnenen Erkenntnisse in Abschnitt 6.4 konzeptionell betrachtet. Die Beispiele aus den Bereichen *Marktbeobachtung* und *Marktforschung* können außerdem sinnvoll auf Basis des eingesetzten Testsystems und der zugrunde liegenden Datengrundlage umgesetzt werden. Darüber hinaus versprechen sich zwischen diesen beiden Anwendungsfällen hohe Synergieeffekte durch die Wiederverwendung der jeweiligen (Zwischen-)Ergebnisse.

In den folgenden Abschnitten werden die praktischen Implementierungen unter Berücksichtigung der verschiedenen Architekturstile gegenübergestellt, um zu beschreiben, welche Vor- und Nachteile eine Realisierung mittels eines zonenbasierten/-losen Data Lakes bietet. Dabei wird mit der detaillierten Betrachtung des Anwendungsfalls der Marktforschung aus Abschnitt 4.2.2 begonnen, gefolgt von der Marktbeobachtung aus Abschnitt 4.2.1.

6.2 Anwendungsfall 1: Marktforschung

Die *Definition der Fragestellung* wurde bereits in Abschnitt 4.2.2 bei der Beschreibung des Anwendungsfalls erfüllt. Das Ziel dieses Anwendungsfalls besteht also darin, die Zufriedenheit der Kunden mit einer Marke und der Untersuchung der Performance eines speziellen Produkts zu erarbeiten. Hierfür wird die Marke *Apple* und deren Produkt, das *Macbook*, im gewählten Datensatz ausgewertet. Für die geplante Meinungsanalyse werden sowohl die harmonisierten Rohdaten, als auch aggregierte Werte benötigt. Da das Ergebnis nach der explorativen Datenanalyse an die Auslieferungszone übergeben wird, beansprucht dieses Beispiel bei der Datenanalyse alle Zonen des Data Lakes.

	Company.csv	**Company_Tweet.csv**	**Tweet.csv**
Attribute	ticker_symbol company_name	tweet_id ticker_symbol	tweet_id writer post_date body comment_num retweet_num like_num

Tabelle VI: Schema der Inputdaten

Verarbeitungsschritt 1: Beschaffung der Daten

Die *Verarbeitung der Daten in der Landezone* stellt den initialen Schritt dar. Da die benötigten Daten bereits im Vorfeld in den Data Lake geladen wurden, beginnt der Datenanalyseprozess somit mit der Untersuchung der vorliegenden Daten und deren Attribute. Die Schemas der Inputdaten sind in Tabelle VI dargestellt. Im Attribut *writer* wird der Benutzername des Verfassers im Klartext angegeben. Diese Personendaten bedürfen jedoch besonderem Schutz und werden daher durch Anwenden der „SHA-256“-Hashfunktion anonymisiert. Dieser Schritt ist in Quelltext 1 beschrieben und erhält durch die kryptografischen Eigenschaften der Hashfunktion (Kollisionsresistenz) ebenfalls die Eindeutigkeit der Zuordnungen der Verfasser zu ihren gesendeten Nachrichten.

```
1 from pyspark.sql.functions import sha2
2 anonymous_tweets = tweets.withColumn("anonymous_tweets", sha2(tweets.writer, 256))
```

Quelltext 1: Anonymisierung der Nachrichtenverfasser

Weitere Änderungen der Daten werden in diesem Fall in der Landezone nicht vorgenommen. Daher folgt die Weitergabe an die Rohdatenzone. PySpark ermöglicht durch einfache Methoden den Import und Export von Datensätzen in bzw. aus dem Data Lake. Dieser Exportschritt ist beispielhaft in Quelltext 2 dargestellt und findet in den folgenden Verarbeitungsschritten beim Export analog statt. Entsprechend der Beschreibung des Zonenreferenzmodells von Giebler et al. [4] werden die Daten nach Abschluss dieses Verarbeitungsschritts aus der Landezone gelöscht, da sie ausschließlich als eine Art Pufferzone für die Rohdatenzone fungiert.

```
1 anonymous_tweets.write.csv('hdfs://localhost:9820/DL/02_Raw/Tweet.csv', header='true')
```

Quelltext 2: Export der anonymisierten Daten in die Rohdatenzone

Verarbeitungsschritt 2: Normalisierung und Aufbau einer einheitlichen Struktur

Nach der Beschaffung der Datengrundlage werden die Daten durch *Normalisierung* in das Zielschema für die weitere Analyse gebracht und in die harmonisierte Zone exportiert. Dieser Schritt dient dem Aufbau einer Struktur und der Reduktion von Varianz zwischen den Schemas und Formaten der vorliegenden Datensätze. Hierfür wird normalerweise ein gemeinsames Zielformat gewählt (CSV, JSON, SQL-Datenbank, ...). Da im betrachteten Anwendungsfall alle Daten bereits einheitlich im CSV-Format vorliegen, ist hier keine Konvertierung notwendig. Die Datensätze „Company.csv“ und „Company_Tweet.csv“ bedürfen keiner weiteren Aufbereitung an dieser Stelle und können somit direkt an die harmonisierte Zone weitergeleitet werden. Aus der Untersuchung der bereits anonymisierten „Tweet.csv“ ergibt sich, dass ebenfalls Nachrichten ohne hinterlegten Verfasser existieren. Des Weiteren sind die gegebenen Informationen im Attribut „post_date“ nicht menschenlesbar dargestellt. Daher werden die vorliegenden Nachrichten in „Tweet.csv“ gefiltert, weiter aufbereitet und auf die benötigten Dimensionen reduziert.

Im ersten Schritt werden die vorhandenen statistischen Attribute (Anzahl an Kommentaren, Re-Tweets und Likes - siehe Tabelle VI) von der Nachricht selbst abgespalten und separat abgelegt. Hierbei bietet PySpark Werkzeuge zur Selektion von Datensätzen an (siehe Quelltext 3).

```
1 tweet = anonymous_tweets.select('tweet_id', 'writer', 'post_date', 'body')
2 tweet_stats = anonymous_tweets.select('tweet_id', 'comment_num', 'retweet_num', 'like_num')
```

Quelltext 3: Unterteilung des Datensatzes durch Selektion

Im Anschluss daran folgt die Konvertierung des nicht lesbaren Unix-Timestamps in ein menschenlesbares Format. Dies wird im Beispiel durch zeilenweise Anwendung der PySpark-SQL-Funktion „from_unixtime“ erreicht und ist in Quelltext 4 dargestellt.

```
1 from pyspark.sql.functions import from_unixtime
2 tweet = tweet.withColumn("post_date", from_unixtime(col("post_date")))
```

Quelltext 4: Konvertierung der Datumsangabe in menschenlesbares Format

Die bereits angesprochenen Nachrichten ohne Verfasser werden nun aus den vollständigen Datensätzen extrahiert und für eine mögliche zukünftige Nutzung getrennt abgelegt. Diese eindeutige Separation führt zu einer höheren Qualität der einzelnen Datensätze und reduziert deren Heterogenität. PySpark bietet hierfür die „filter“-Funktion an, welche in Quelltext 5 beispielhaft gezeigt wird.

```
1 tweet_clean = tweet.filter(tweet.writer.isNotNull())
2 tweet_null = tweet.filter(tweet.writer.isNull())
```

Quelltext 5: Extraktion der Nachrichten ohne Verfasser

Da bei der Bearbeitung von Anwendungsfällen im Umfeld eines zonenbasierten Data Lakes ein wichtiger Punkt in der zukünftigen Wiederverwendbarkeit der Datensätze liegt, werden die selektierten, konvertierten und gefilterten Daten separat in die harmonisierte Zone exportiert. Somit ist es möglich, bei zukünftigen Aufgabenstellungen wiederholte Berechnungen zu vermeiden und die bereits vorhandenen Datensätze wiederzuverwenden.

Verarbeitungsschritt 3: Aggregationen

Zur Vorbereitung der anschließenden Analysen werden nach der Normalisierung und Strukturierung der Daten *aggregierte Zwischenergebnisse* als Basis für die anschließende Datenanalyse berechnet. Hierfür wird zuerst ein Gesamtdatensatz mit allen Nachrichten und ihren zugehörigen Firmennamen generiert. Dieser wird daraufhin gruppiert und aggregiert, um konzentrierte Werte auf monatlicher Basis zu erhalten. Der Gesamtdatensatz wird in PySpark mithilfe der Funktionen „union“ und „join“ (Quelltext 6, Zeile 1-4) konstruiert und anschließend durch die PySpark-SQL-Funktionen „groupBy“ und „count“ aggregiert (Quelltext 6, Zeile 6-11). Die neu erstellten Datensätze werden zum Abschluss des Aggregationsschrittes in die Reduktionszone exportiert.

```
1  tweets = tweet_clean.union(tweet_null)
2  joinCompanyTweets = tweets.join(compTweet, 'tweet_id')
3  joinALL = joinCompanyTweets.join(comp, 'ticker_symbol').select('tweet_id', 'writer',
4            'post_date', 'body', 'ticker_symbol', 'company_name')
5
6  from pyspark.sql.functions import year, month, count
7  aggregate = joinALL.groupBy(year('post_date').alias('year'),
8              month('post_date').alias('month'), 'company_name')
9              .agg(count('company_name').alias('number_of_tweets'))
10
11 aggregate = aggregate.orderBy(["year", "month", "company_name"], ascending=True)
```

Quelltext 6: Erzeugen eines zusammengesetzten Datensatzes und anschließende Aggregation der monatlichen Tweetzahlen

Verarbeitungsschritt 4: Explorative Datenanalyse

Die explorative Datenanalyse am Beispiel der Kundenzufriedenheit und Produktperformance kann in drei wesentliche Teilschritte unterteilt werden: Die Meinungsanalyse der Tweets, die Aggregation der Ergebnisse, sowie deren Visualisierung.

Diese Teilprozesse werden jeweils unter Betrachtung der Marke „Apple“ und dem Produkt „Macbook“ durchgeführt, um aus der Meinungsanalyse Rückschlüsse auf Kundenzufriedenheit und Produktperformance ziehen zu können.

Für die Meinungsanalyse wird initial der im vorherigen Verarbeitungsschritt erstellte Gesamtdatensatz geladen. Im Anschluss wird eine benutzerdefinierte Funktion mithilfe der Bibliothek „Textblob“[4] bestimmt, welche bei zeilenweiser Anwendung die Meinungsanalyse auf dem Nachrichtentext durchführt. Die angesprochene Funktion zur Meinungsanalyse bestimmt die Polarität einer Nachricht (Positiv= 1, Neutral= 0, Negativ= -1) im Intervall $[1, -1]$. Dieser Vorgang wird in Quelltext 7 beschrieben und liefert die Gesamtdatei mit einer neuer Spalte „sentiment“ zurück. Das generierte Ergebnis wird wiederum in der explorativen Zone gesichert.

[4] https://textblob.readthedocs.io/en/dev/

```
from textblob import TextBlob
sentiment = udf(lambda x: TextBlob(x).sentiment[0])
sparkSession.udf.register("sentiment", sentiment)

sentiment = joinALL.withColumn('sentiment', sentiment('body').cast('double'))
```

Quelltext 7: Meinungsanalyse der Twitter-Nachrichtentexte

In diesem Fall wird die Aggregation der Ergebnisse der Meinungsanalyse als separater Schritt betrachtet, weshalb hierfür die exportierten Daten erneut eingelesen werden. Daraus ergibt sich bereits ein Vorteil: Da die Aggregation sowohl für die Marke „Apple“, als auch das Produkt „Macbook“ analog durchgeführt wird, kann der eingelesene Datensatz für beide Fälle unverändert verwendet werden.

Mittels der bereits beschriebenen „filter“-Funktion wird der Gesamtdatensatz in jeweils zwei Teilmengen unterteilt:

- Nachrichten über die Marke „Apple“ (Quelltext 8, Zeile 1)
- Nachrichten zu „Apple“, welche „Macbook“ im Nachrichtentext enthalten (Quelltext 8, Zeile 2).

Die gefilterten Datensätze werden ebenfalls für die Möglichkeit der zukünftigen Wiederverwendung in die explorative Zone exportiert.

```
sentiment_apple = sentiment.filter(sentiment.company_name=='apple')
sentiment_macbook = sentiment_apple.filter(lower(sentiment_apple.body).rlike('macbook'))
```

Quelltext 8: Extraktion der Nachrichten über „Apple“ sowie mit Inhalt „Macbook“

Im Folgenden wird nun die Vorgehensweise für die Visualisierung der Ergebnisse für die Marke „Apple“ beschrieben. Für das Produkt „Macbook“ gilt das Verfahren analog.

Die gefilterten Daten zu den Nachrichten über „Apple“ werden zu Beginn neu importiert und anschließend in Monatsintervallen aggregiert. Dabei wird je Intervall die Anzahl der Nachrichten aufsummiert und deren durchschnittliche Polarität ermittelt. Dies lässt sich durch PySpark ebenfalls sehr kompakt mittels der „groupBy“-SQL-Funktion durchführen (siehe Quelltext 9). Der resultierende Datensatz stellt das Ziel des Anwendungsfalls dar und wird daher sowohl als Ergebnis in der Auslieferungszone bereitgestellt, als auch zur zukünftigen Wiederverwendung in der Reduktionszone gesichert.

```
apple_agg_sentiment = sentiment_apple.groupBy(year('post_date').alias('year'),
                      month('post_date').alias('month'), 'company_name')
                      .agg(count('company_name').alias('number_of_tweets'),
                      mean('sentiment').alias('mean_sentiment'))
                      .orderBy(["year", "month"], ascending=True)
```

Quelltext 9: Aggregation der Meinungsanalyse über „Apple“-Nachrichten

Zur Unterstützung des Entscheidungsprozesses werden die berechneten Ergebnisse abschließend visualisiert. Hierfür werden die Resultate erneut eingelesen, um sie für die Darstellung mittels der gängigen Python-Bibliothek „matplotlib“ aufzubereiten. Dieser Programmablauf ist in Quelltext 10 dargestellt. Dabei wird zuerst in Zeile 1-5 das Label für die Zeitachse zusammengesetzt, um es danach in eine Python-Liste als Basis für die X-Achse des Diagramms umzuwandeln.

Im Anschluss werden in Zeile 7-8 die aggregierten Werte für die beiden Y-Achsen ebenfalls in Python-Listen überführt. Nachdem alle benötigten Daten im gewünschten Format vorliegen, kann mithilfe von „matplotlib“ unkompliziert ein Diagramm erstellt werden. Dieses Diagramm ist in Abbildung 6 dargestellt, das Ergebnis für das Produkt „Macbook“ in Abbildung 7.

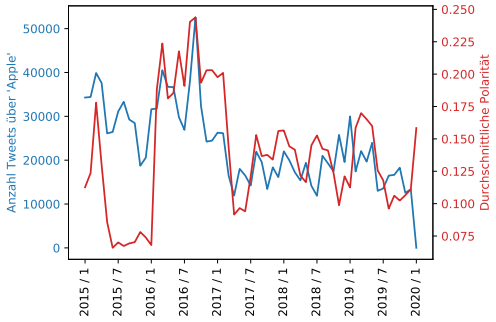

Abbildung 6: Anzahl Tweets über „Apple“ und deren Polarität

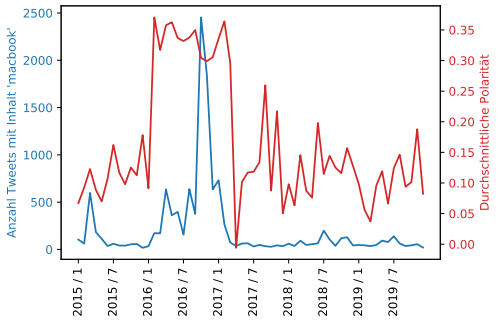

Abbildung 7: Anzahl Tweets zu „Macbook“ und deren Polarität

Eine Analyse auf der betriebswirtschaftlichen Domänenebene würde an diese Auswertungen anknüpfen, um die Fragestellung bezüglich Kundenzufriedenheit und Produktperformance zu erörtern. Dies wird im Rahmen der Arbeit aus Gründen des Umfangs und des inhaltlichen Bezugs nicht weiter vertieft.

```
1 mergeCols = udf(lambda year, month: str(year) + ' / ' + str(month))
2 apple_date = apple_agg_sentiment.withColumn("date", mergeCols(col("year"), col("month")))
3              .select('date')
4
5 dates = apple_date.select("date").rdd.flatMap(lambda x: x).collect()
6
7 numTweets = apple_agg_sentiment.select("number_of_tweets").rdd.flatMap(lambda x: x).collect()
8 meanSent = apple_agg_sentiment.select("mean_sentiment").rdd.flatMap(lambda x: x).collect()
```

Quelltext 10: Aufbereitung der Ergebnisse zur Erstellung einer Visualisierung

Datenanalyseprozess im zonenlosen Data Lake

Wird der beschriebene Anwendungsfall auf Grundlage eines zonenlosen Data Lakes aufgebaut, so ergibt sich prinzipiell derselbe Ablauf. Dennoch gilt es wichtige Punkte zu unterscheiden. In diesem Beispiel wird ein zonenloser Data Lake als gänzlich strukturloses Schema definiert und praktisch durch ein einzelnes Verzeichnis „XX_Zoneless“ im HDFS abgebildet. Ebenfalls wird auf die Speicherung von Zwischenergebnissen verzichtet. Dies führt zu einem analogen Ablauf im Vergleich zum zonenbasierten Data Lake aus Abschnitt 6.2. Jedoch werden ausschließlich die Eingangsdaten initial importiert und anschließend alle folgenden Verarbeitungsschritte durch sequentielle Ausführung berechnet. Diese Herangehensweise führt zu demselben Ergebnis, welches in Abschnitt 6.2 vorgestellt wurde.

Diese Ausführung eröffnet Spielraum für Optimierungen gegenüber der wiederholten Speicherung der Zwischenergebnisse im zonenbasierten Data Lake. Somit fallen beispielsweise keine zusätzlichen Im-/Export-Schritte bis hin zum finalen Ergebnis an. Dadurch wird die Speichernutzung und die Komplexität des Codes signifikant reduziert. Kapitel 7 geht dabei näher auf die Bewertung der unterschiedlichen Herangehensweisen ein. In diesem Anwendungsfall gibt es ansonsten keine gravierenden Unterschiede im allgemeinen Ablauf der Verarbeitungsschritte.

Nach Abschluss des Anwendungsfalls der Marktforschungs stellt sich der Inhalt des HDFS wie folgt dar:

```
/HDFS/DL/
├── 01_Landing
├── 02_Raw
│   ├── Company.csv
│   ├── Company_Tweet.csv
│   └── Tweet.csv
├── 03_Harmonized
│   ├── Company.csv
│   ├── Company_Tweet.csv
│   ├── Tweet.csv
│   └── Tweet_Writer_null.csv
├── 04_Distilled
│   ├── Tweets_Apple_Sentiment_Aggregated.csv
│   ├── Tweets_Apple-Macbook_Sentiment_Aggregated.csv
│   ├── Tweets_Companies_Joined.csv
│   └── Tweets_per_Company_per_Month.csv
├── 05_Explorative
│   └── Tweets_Companies_Sentiment.csv
├── 06_Delivery
│   ├── Tweets_Apple_Sentiment_Aggregated.csv
│   ├── Tweets_Apple_Sentiment_Plot.csv
│   ├── Tweets_Apple-Macbook_Sentiment_Aggregated.csv
│   └── Tweets_Apple-macbook_Sentiment_Plot.csv
└── XX_Zoneless
    ├── Company.csv
    ├── Company_Tweet.csv
    ├── Tweet.csv
    ├── Tweets_Apple_Sentiment_Aggregated.csv
    ├── Tweets_Apple_Sentiment_Plot.csv
    ├── Tweets_Apple-Macbook_Sentiment_Aggregated.csv
    └── Tweets_Apple-macbook_Sentiment_Plot.csv
```

Abbildung 8: Verzeichnisstruktur des HDFS nach Abschluss der Marktforschung (vereinfacht)

6.3 Anwendungsfall 2: Marktbeobachtung

Nun wird der Anwendungsfall „Marktbeobachtung“ beleuchtet. Dieser beschäftigt sich mit der Analyse von Marktstellung und Popularität einer Marke gegenüber ihrer Konkurrenz. Um die Möglichkeiten eines wachsenden Data Lakes mit vorhandenen historischen Daten zu simulieren wird hierbei angenommen, dass diese Analyse nach Abschluss der Marktforschung aus Abschnitt 6.2 stattfindet. Nachfolgend rücken die Unterschiede in den Verarbeitungsschritten in den Fokus, da die technische Durchführung der Herangehensweise des vorherigen Abschnitts analog folgt.

Das Ziel dieses Anwendungsfalls besteht in der Aggregation der monatlichen Anzahl an Nachrichten, jeweils gruppiert nach Unternehmen, um den zeitlichen Verlauf des Marktgeschehens nachzustellen. In den beiden Unterkapiteln wird auf die Unterschiede bei Nutzung eines zonenbasierten oder zonenlosen Data Lakes eingegangen.

Datenanalyseprozess im zonenbasierten Data Lake

Wird nun angenommen, dass im zonenbasierten Data Lake die gespeicherten Zwischenergebnisse (siehe Abbildung 8) der vorherigen Analyse bereits vorhanden sind, so können erhebliche Optimierungen im Ablauf des Datenanalyseprozesses angewendet werden. Für die gewählte Fragestellung sind keine zusätzlichen Datensätze nötig. Daher können sämtliche Zwischenergebnisse der vorherigen Analyse wiederverwendet werden. Der Datensatz der monatsweise aggregierten Nachrichten kann somit direkt aus der Reduktionszone importiert werden. Mithilfe weniger Filter-Operationen wird unkompliziert und schnell die geforderte Unterteilung der verschiedenen Marken erreicht. Die erhaltenen Ergebnisse werden wiederum für mögliche zukünftige Zwecke in der Reduktionszone gesichert. Beispielhaft sind die nötigen Schritte zur Aggregation der Nachrichten über „Amazon“ in Quelltext 11 zusammengefasst.

```
aggregate = sparkSession.read.csv('hdfs://localhost:9820/DL/04_Distilled/
          Tweets_per_Company_per_Month.csv', header='true', inferSchema='true')
          .orderBy(["year", "month"], ascending=True)

aggAmazon = aggregate.filter(aggregate.company_name=="Amazon.com")

aggAmazon.write.csv('hdfs://localhost:9820/DL/04_Distilled/
          Tweets_Amazon_Aggregated.csv', header='true')
```

Quelltext 11: Filterung der Nachrichten über „Amazon“

Diese Vorgehensweise wird für die restlichen Marken analog durchgeführt, wodurch bereits das gewünschte Ergebnis der Analyse erzielt wird. Damit schließt sich direkt eine Visualisierung entsprechend des Prozesses aus Abschnitt 6.2 an. Das resultierende Diagramm ist in Abbildung 9 dargestellt.

Es kann bereits festgehalten werden, dass der Datenanalyseprozess in diesem Anwendungsfall signifikant durch die Wiederverwendung von vorhandenen Daten profitiert. Hierzu stellt der zonenbasierte Data Lake durch seinen schematischen Aufbau entlang des Datenanalyseprozesses eine ideale Umgebung zur Verfügung. Eine detaillierte Diskussion folgt in Kapitel 7.

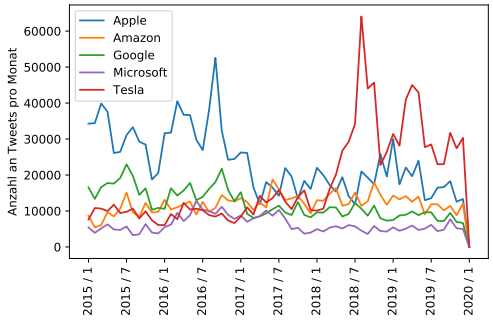

Abbildung 9: Anzahl Tweets zu den „Top-Companies“ des NASDAQ

Datenanalyseprozess im zonenlosen Data Lake

Wird dieser Anwendungsfall durch einen Data Lake mit zugrundeliegender zonenloser Architektur realisert, so ergibt sich ein signifikanter Unterschied im Ablauf des Datenanalyseprozesses. Wie bereits in Abschnitt 6.2 beschrieben, wurde auf die Speicherung von Zwischenergebnissen in der zonenlosen Struktur verzichtet. Somit ist es hier *nicht* möglich direkt auf den aggregierten Daten aufzubauen. Daher ist es nötig, alle Verarbeitungsschritte von der Beschaffung bis hin zur Aggregation der Daten erneut sequentiell durchzuführen. Erst danach kann analog zu Abschnitt 6.3 die geforderte Unterteilung der Daten anhand der verschiedenen Marken erstellt werden.

Somit zeigt sich, dass vor allem die konsequente Speicherung der berechneten Zwischenergebnisse die Umsetzung von zukünftigen Anwendungsfällen durch Wiederverwendung der historischen Daten enorm unterstützen kann. Das Fehlen solcher Zwischenergebnisse führt dazu, dass bei Nutzung einer zonenlosen Data-Lake-Architektur oftmals redundante Berechnungsschritte notwendig sind. Eine detaillierte Evaluation der Unterschiede zwischen den Architekturstilen wird in Kapitel 7 durchgeführt.

6.4 Anwendungsfall 3: Prognose zukünftiger Entwicklungen

Der dritte beschriebene Anwendungsfall aus Abschnitt 4.2.3 wurde aufgrund des Umfangs nicht praktisch implementiert. Dennoch kann eine konzeptionelle Vorgehensweise mithilfe der Erfahrungen aus den realisierten Anwendungsfällen der vorherigen Abschnitte abgeleitet werden.

Sollen auf dieser Basis Aussagen über die Ausfallwahrscheinlichkeit eines Produktes getroffen werden, so können beispielsweise die Auswertungen bezüglich der Kundenzufriedenheit bei Macbooks (siehe Abschnitt 6.2) erneut herangezogen werden. Hierbei können die Nachrichtentexte detailliert betrachtet werden, um Informationen über Defekte und Fehler zu finden. Dieses Wissen könnte wiederum mit Daten zu Produktneuheiten und deren produzierten Einheiten verglichen werden, um damit Muster bezüglich des Ausfallgeschehens einzelner Produkte zu erkennen.

Dieses Szenario würde analog zum Anwendungsfall der Marktforschung von bereits vorhandenen Daten innerhalb des zonenbasierten Data Lakes profitieren. Fließen zudem die Produktionsdaten in den Data Lake ein, so werden diese in der Praxis im Sinne des innerbetrieblichen Reportings monatsweise aggregiert. Unter der Annahme, dass bei Nutzung einer zonenbasierten Architektur diese Ergebnisse wieder in den Data Lake zurückgeschrieben werden, sind alle benötigten Daten für das beschriebene Szenario bereits vorhanden und können für die Analyse genutzt werden. Wird ein zonenloser Data Lake ohne Zwischenergebnisse angenommen, so sind für alle Teilschritte (Verarbeitung der Produktionsdaten, Analyse der Social-Media-Daten über das jeweilige Produkt, etc.) die separaten Vorarbeiten nötig, bevor der Anwendungsfall konkret analysiert werden kann.

Unter dem Gesichtspunkt der Arbeitsteilung und Kooperation verschiedener Fachbereiche ist es ebenfalls denkbar, dass die einzelnen Teilergebnisse durch unterschiedliche Spezialisten erarbeitet werden. So könnten beispielsweise die Ergebnisse der Marktforschung durch einen weiteren Data Scientisten berechnet werden, während die Vorverarbeitung der Produktionsdaten von einem Data Engineer ausgeführt wird. Dabei unterstützt die Struktur eines zonenbasierten Data Lakes diese kooperative Herangehensweise. Die Informationen über den Zustand der benötigten Daten werden dabei bereits anhand der Unterteilung nach den jeweiligen Verarbeitungsgraden innerhalb der Zonen impliziert. Ein zonenloser Data Lake ohne klaren Aufbau und Struktur erschwert den Informationsaustausch jedoch erheblich. Hier wären zusätzliche Absprachen zwischen den beteiligten Spezialisten und Fachbereichen nötig.

Der Beschreibung der prototypischen Implementierung der verschiedenen Anwendungsfälle schließt sich eine Diskussion zur Bewertung der gewonnen Erfahrungen in Kapitel 7 an. Hierbei wird der Fokus auf die Vor- und Nachteile bei der Nutzung von zonenbasierten und zonenlosen Data Lakes gelegt, um die Vorzüge beider Architekturstile bei der Umsetzung von Datenanalysen herauszustellen. Vorab fasst Tabelle VII die Vorgehensweise bei der Umsetzung der Anwendungsfälle kompakt zusammen. Sie stellt dabei ebenfalls gegenüber, inwiefern die prototypische Implementierung sich an den definierten allgemeinen Anforderungen an Data-Science-Anwendungsfälle aus Abschnitt 4.1 und dem Leitfaden für die Implementierung aus Abschnitt 5.4 orientiert.

Anforderungen an Data-Science-Anwendungsfälle	**Leitfaden für die Implementierung**	**Anwendungsfall 1: Marktforschung**	**Anwendungsfall 2: Marktbeobachtung**
Definition von Frage- & Problemstellungen	• Selektion der nötigen Schritte durch Vorüberlegungen	• Auswertungen von Twitter-Nachrichten zu Marke „Apple“ und Produkt „Macbook“ • Durchführung einer Meinungsanalyse → Alle Zonen des Data Lakes werden benötigt!	• Aggregation der monatlichen Anzahl an Nachrichten je Unternehmen • Auswertung des Marktverhaltens • Annahme: Analyse im Anschluss an Anwendungsfall 1 → Nutzung vorheriger Ergebnisse, somit nicht alle Verarbeitungsschritte erneut benötigt
Bereitstellung qualitativ hochwertiger Daten	• Bereitstellung durch die Zonen je nach Verarbeitungsgrad • Ankunft der Daten in Landezone • Minimale Änderungen möglich vor Weiterleitung in Rohdatenzone • Überführung in einheitliches Format in harmonisierter Zone, als Grundlage für die weiteren Schritte • Zusammenführung und Gruppierung der Daten in der Reduktionszone	• Benötigte Daten in Landezone vorab geladen • Anonymisierung der Personendaten vor Export in Rohdatenzone • Konvertierung des Datumsformats • Aufspaltung der Datensätze zur Normalisierung • Export an harmonisierte Zone • Erstellung eines Gesamtdatensatzes • Aggregation auf monatlicher Basis • Export an Reduktionszone	• Nutzung des vorliegenden Gesamtdatensatzes • Filterung nach den betrachteten Marken • Export der Ergebnisse an die Reduktionszone
Explorative Datenanalyse	• Verstehen der Problemstellung, Koordination der Vorgehensweise, Untersuchung relevanter Daten • Erstellung aussagekräftiger Analysen • Rückführung der Ergebnisse in Reduktionszone und Bereitstellung über Auslieferungszone	• Durchführung der Analysen für Marke „Apple“ und Produkt „Macbook“ • Meinungsanalyse der Nachrichten und Bestimmung der Polarität • Filterung der Ergebnisse nach „Apple“ und „Macbook“ • Export der Ergebnisse in Reduktionszone und Auslieferungszone	• Durchführung der Analysen für Marken „Amazon“, „Google“, „Microsoft“, „Tesla“ • Wiederverwendung der Ergebnisse für Marke „Apple“ aus Anwendungsfall 1
Wissensgewinnung und Visualisierung	• Gewinnung von Wissen durch Interpretation der erhaltenen Informationen • Erstellen einer Visualisierung und Übergabe der abstrahierten Daten an Domänenexperten	• Aggregation der Resultate der Meinungsanalyse • Erstellung aussagekräftiger Diagramme zur Unterstützung anschließender Entscheidungsprozesse • Bereitstellung der Diagramme und Ergebnisse in der Auslieferungszone	• Aggregation der Marktbeobachtungsdaten • Erstellung aussagekräftiger Diagramme zur Unterstützung anschließender Entscheidungsprozesse • Bereitstellung der Diagramme und Ergebnisse in der Auslieferungszone

Tabelle VII: Gegenüberstellung zwischen Leitfaden und konkreter Implementierung der Anwendungsfälle

7 Evaluation

In Kapitel 6 wurde anhand einer prototypischen Implementierung die Umsetzbarkeit von Data-Science-Anwendungsfällen in zonenbasierten Data Lakes untersucht. Hierbei wurden zwei Anwendungsfälle praktisch umgesetzt, um die Vorzüge und Potentiale gegenüber einer zonenlosen Architektur ohne Speicherung von Zwischenergebnissen herauszustellen.

Es zeigt sich, dass der zugrundeliegende Architekturstil des Data Lakes Anzahl und Umfang der notwendigen Verarbeitungsschritte im Datenanalyseprozess maßgeblich beeinflusst. Der Aufbau einer umfangreichen Datengrundlage im zonenbasierten Data Lake kostet Zeit und Ressourcen, vor allem hinsichtlich Speicherkapazität und Rechenzeit beim Im- & Export. Jedoch können zukünftige Analysen durch die Nutzung dieser vorhandener Daten in hohem Maß profitieren. Durch den Import bereits bestehender Daten konnte bei der Marktbeobachtung (Abschnitt 6.3) auf die vorgelagerten Schritte der Datenbeschaffung und -vorverarbeitung verzichtet werden. Dadurch ergibt sich eine signifikante Verringerung des Gesamtaufwands für die Analyse des Anwendungsfalls bei Nutzung der zonenbasierten Architektur. Zur Validierung der erkannten Muster wurden die realen Rechenzeiten des Testsystems bei der Ausführung der verschiedenen Verarbeitungsschritte erfasst und in Tabelle VIII dokumentiert.

Bei Betrachtung der Laufzeitanalyse lassen sich mehrere Punkte hervorheben:

1. Die Gesamtlaufzeiten der implementierten Anwendungsfälle sind jeweils bei Nutzung einer zonenbasierten Data-Lake-Architektur niedriger als bei einer zonenlosen Struktur.
2. Die Verarbeitungsschritte von der Beschaffung bis einschließlich der Aggregation der Daten werden bei Nutzung einer zonenlosen Architektur insgesamt - aber auch jeweils einzeln - schneller berechnet.
3. Die Durchführung der explorativen Datenanalyse ist jeweils deutlich performanter unter Nutzung eines zonenbasierten Data Lakes
4. Besonders auffällig ist, dass in Anwendungsfall 2 (Abschnitt 6.3) während der Beschaffung, Normalisierung und Aggregation keine Zeit für Berechnungen benötigt wird.

Diese Beobachtungen eröffnen verschiedene Vor- und Nachteile der verglichenen Architekturstile. Wie in Punkt 2 beschrieben, zeigt sich, dass die einzelnen Schritte der Datenverarbeitung und -aufbereitung bei Nutzung einer zonenlosen Architektur schneller durchgeführt werden können. Hierbei führt die Reduzierung der notwendigen Transformationsschritte sowie insbesondere die Ersparnis der Im- & Exportschritte für die berechneten Zwischenergebnisse zu einer signifikanten Verkürzung der Laufzeit gegenüber einer zonenbasierten Architektur.

Laufzeit [**sek.**]	**Anwendungsfall 1**		**Anwendungsfall 2**	
	Zonenbasiert	Zonenlos	Zonenbasiert	Zonenlos
1. Beschaffung	111,6	58,8	0	59,1
2. Normalisierung	105,2	37,6	0	32,6
3. Aggregation	300,0	97,0	0	97,6
4. Datenanalyse	1.022,2	3.061,2	352,2	497,1
Gesamtlaufzeit	1.539,0	3.254,6	352,2	686,4

Tabelle VIII: Laufzeitanalyse der betrachteten Anwendungsfälle nach Architekturstil des Data Lakes

Der aufgebrachte Mehraufwand während diesen Verarbeitungsschritten resultiert jedoch in einem deutlich niedrigeren Rechenaufwand bei der explorativen Datenanalyse in zonenbasierten Data Lakes (Punkt 3). Aufgrund des Imports vorab erstellter Zwischenergebnisse kann bei der Nutzung dieser Daten während des Analyseprozesses auf erneute Berechnungen verzichtet werden. Aus den zeitlichen Einsparungen bei der Datenanalyse folgt schließlich die kürzere Gesamtlaufzeit (Punkt 1) im Vergleich zur zonenlosen Architektur.

Diese Erkenntnis lässt sich ebenfalls durch eine Charakteristik der genutzten Testumgebung bestätigen. Die eingesetzte PySpark-Umgebung, welche auf Basis von Spark (siehe Abschnitt 2.2) als Anwendung für die Verarbeitung von Big Data konzipiert ist, basiert auf dem Prinzip der trägen Auswertung (engl. *„lazy evaluation strategy“*) [34]. Dieses besagt, dass Ausdrücke und Programmteile erst ausgewertet und berechnet werden, sobald eine konkrete Aktion vorliegt. Dies wird zur Optimierung des Zugriffs und der Nutzung vorhandener Ressourcen (Hauptspeicher, Laufwerksspeicher, Prozessoren, etc.) angewandt und ermöglicht eine höhere Rechenleistung von Spark gegenüber einer naiven Implementierung des *MapReduce*-Paradigmas ([35]). Im Umfeld von Big Data ist durch den Umfang der Datenmenge jedoch kein Caching eines gesamten Datensatzes durch lokale, begrenzte Ressourcen möglich. Daher wurde bei der durchgeführten Simulation eines solchen Umfelds ebenfalls auf Caching und weitere Optimierungen verzichtet.

Dies hat zur Folge, dass bei Nutzung eines zonenlosen Data Lakes, durch die fehlende Berechnung und Speicherung von Zwischenergebnissen, lange Berechnungsketten aufgebaut werden. Diese rechenintensiven Anweisungen führen zu der beobachteten zeitintensiven Datenanalyse bei zonenlosen Data Lakes.

Zusätzlich wurde bei der Marktbeobachtung im zonenbasierten Data Lake in Abschnitt 6.3 gezeigt, dass durch die Speicherung von Zwischenergebnissen und der Einführung einer Struktur innerhalb des Data Lakes vor allem zukünftige Anwendungsfälle profitieren können. Durch den Import bereits vorab aggregierter Daten konnte direkt im Schritt der explorativen Datenanalyse eingestiegen werden (Punkt 4). Mit wachsendem Umfang des Data Lakes steigt somit auch die Wahrscheinlichkeit, dass die benötigten Daten für die jeweiligen Auswertungen bereits fertig aufbereitet vorliegen und genutzt werden können. Dies führt letztendlich dazu, dass stetig wiederholte Berechnungen für häufig genutzte Daten durch einmalige Berechnung und Speicherung als Zwischenergebnis ersetzt werden können. Hierbei überwiegt der Nutzen durch die Ersparnis an Rechenzeit den zusätzlichen Aufwand an Speicherbedarf, da ebenfalls die Kosten für Speicher kontinuierlich sinken [36]. In den implementierten Anwendungsfällen stellt sich dies wie in Tabelle IX beschrieben dar.

Ressourcenaufwand	Zonenbasiert		Zonenlos	
	Rechenzeit	Speicherplatz	Rechenzeit	Speicherplatz
Anwendungsfall 1	26min	4.194 MB	54min	736 MB
Anwendungsfall 2	6min	< 1 MB	11min	< 1 MB
Gesamtaufwand	32min	4.195 MB	65min	737 MB

Tabelle IX: Übersicht der aufgewendeten Ressourcen der betrachteten Anwendungsfälle

Aus der Nutzung von Daten mit verschiedenen, vorab berechneten Verarbeitungsgraden ergeben sich ebenfalls weitere Möglichkeiten im Umgang mit zonenbasierten Data Lakes. In Abschnitt 5.4 wurde bereits diskutiert, dass zwischen dem schematischen Aufbau einer zonenbasierten Data-Lake-Architektur und den allgemeinen Anforderungen an Data-Science-Anwendungsfällen Parallelen bestehen. Daher ist es ebenfalls interessant die Anforderungen an zonenbasierte Data Lakes ebenso in Bezug auf die verschiedenen Data-Science-Spezialisten zu betrachten, analog zu Abschnitt 5.3.

Hierbei wird ersichtlich, dass durch die Einführung einer strukturierten Datenhaltung innerhalb des Data Lakes, beispielsweise durch eines der Zonenmodelle, eine klare Unterteilung der einzelnen Verarbeitungsschritte des Datenanalyseprozesses leicht möglich ist. Deshalb wurde während der prototypischen Implementierung der Anwendungsfälle im zonenbasierten Data Lake bereits darauf geachtet, jeden Verarbeitungsschritt separat zu isolieren. Auf dieser Basis wäre es problemlos möglich, die einzelnen Aufgaben entlang des Datenanalyseprozesses unter verschiedenen Data-Science-Spezialisten aufzuteilen. Somit könnten zonenbasierte Data Lakes die Kooperation von Data-Science-Teams unterstützen und eine verteilte Bearbeitung von Anwendungsfällen unkompliziert ermöglichen.

Hinsichtlich der Kooperation verschiedener Spezialisten und unterschiedlichen Teams entlang des Datenanalyseprozesses würden die verschiedenen Zonenmodelle dementsprechend von einer erneuten Evaluation profitieren. Eine Möglichkeit zur Verbesserung des Informationsaustauschs zwischen den verschiedenen Beteiligten besteht in der Einführung eines *Transfer*-Bereichs.

Im Rahmen laufender Analyseprozessen sind der stetige Austausch und die Kooperation zwischen mehreren Spezialisten oder ganzen Teams gängig. Hierbei gilt häufig, dass die berechneten Zwischenergebnisse und Modelle noch keinen exportfähigen Zustand für die Reduktions- oder Auslieferungszone erreicht haben. Anhand der bestehenden Gegebenheiten von Zonenmodellen wäre somit bei der Arbeit von einzelnen Data Scientisten ein einfacher Austausch zum Stand der jeweiligen Arbeit nicht möglich. Daher wird an dieser Stelle ein zusätzlicher Bereich für den agilen und ungebundenen Informationsfluss angedacht. Um einen derartigen Bereich zu integrieren sind verschiedene Ansätze denkbar. Einerseits könnte die *explorative Zone* des Zonenreferenzmodells durch eine untergeordnete Zone erweitert werden. Dadurch wird der unkomplizierte Austausch aktueller Projektstände zwischen Data Scientisten innerhalb der isolierten Zone ermöglicht. Dieser Ansatz ist in Abbildung 10 dargestellt.

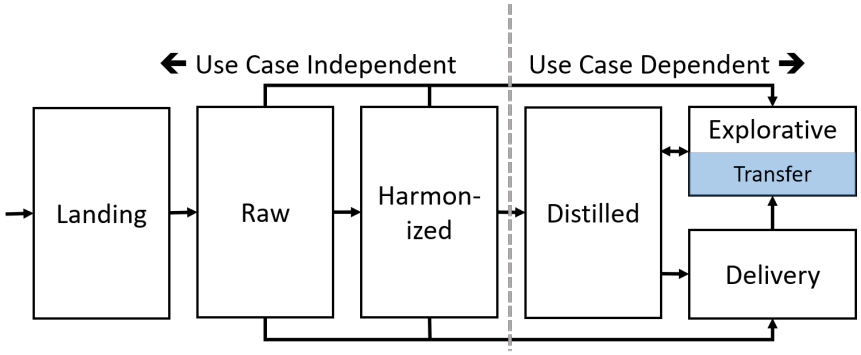

Abbildung 10: Zonenreferenzmodell mit Transfer-Bereich in der explorativen Zone

Da in der *explorativen Zone* strikte Zugriffsbeschränkungen gelten, wäre andererseits die Definition einer neuen, unabhängigen Zone denkbar. In einer möglichen *Transferzone* könnten demnach alle Nutzer des Data Lakes die Zwischenstände ihrer Arbeit untereinander austauschen. Ebenso würden Dialoge zu Informationen, welche (noch) nicht den geforderten Grad der Verarbeitung zur dauerhaften Speicherung innerhalb des Data Lakes erreicht haben, erleichtert werden. Dieser Ansatz ist schematisch in Abbildung 11 skizziert. Damit wäre es denkbar, dass durch die Einführung eines *Transfer*-Bereichs nicht nur die Kooperation zwischen den verschiedenen Beteiligten gestärkt wird, sondern auch die Datenanalyse von einer unkomplizierten Möglichkeit zum Austausch (bei auftretenden Problemstellungen) profitiert.

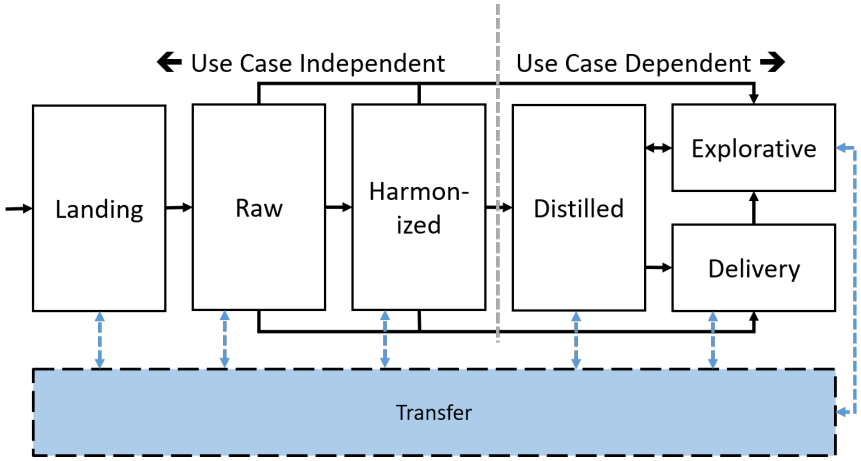

Abbildung 11: Zonenreferenzmodell mit zusätzlicher Transferzone

Es lässt sich zusammenfassen, dass die Anforderungen von Data-Science-Anwendungsfällen an zonenbasierte Data Lakes durch den starken Praxisbezug der Konzepte erfüllt sind. Die prototypische Implementierung beispielhafter Anwendungsfälle zeigt, dass zonenbasierte Data Lakes ebenfalls Vorzüge bei der Datenanalyse bieten. So ermöglichen sie es beispielsweise, vorab berechnete Zwischenergebnisse für zukünftige Auswertungen wiederzuverwenden, sodass redundante Berechnungen reduziert werden können.

Das folgende Kapitel gibt abschließend einen Überblick über die Erkenntnisse aus dieser Arbeit. Im Ausblick werden dabei zudem offene Fragestellungen für zukünftige Forschung diskutiert.

8 Zusammenfassung

Durch die stetig wachsende Menge an verschiedensten Daten im privaten und speziell auch wirtschaftlichen Umfeld sind in den vergangenen Jahren *Data Lakes* als Konzept zur flexiblen Speicherung von *Big Data* entstanden. Durch ihre Eigenschaften als zentralisierter Knoten zur Datenhaltung liegt ein besonderer Fokus auf der Strukturierung der beinhalteten Menge an Daten. Mithilfe des Aufbaus einer geordneten Struktur bleibt auch bei steigendem Datenvolumen die Übersichtlichkeit über die gespeicherten Daten erhalten. Dies führt zu einer Verbesserung der Effizienz bei der Nutzung eines Data Lakes und hilft somit schließlich auch dabei, die Analyseprozesse zu unterstützen. Hieraus haben sich mit dem Fokus auf die praktischen Abläufe in der Datenanalyse verschiedene zonenbasierte Architekturen gebildet. Sie unterteilen die Datenhaltung innerhalb eines Data Lakes in verschiedene Bereiche („Zonen“) anhand des Verarbeitungsgrads der jeweiligen Daten.

Da Data Lakes eine große Menge der unternehmensweiten Daten beinhalten, sind sie ebenfalls in Bezug auf Datenanalyse und *Data Science* von Relevanz. Alle gängige Zonenmodellen für Data Lakes thematisieren folglich die Rolle von Data Scientisten und stellen ihnen oftmals zusätzlich einen isolierten Bereich zur freien Analyse von Daten bereit. Dennoch existiert wenig wissenschaftliche Forschung in diesem Bereich. Daher wurden in dieser Ausarbeitung Anforderungen an Data-Science-Anwendungsfällen im Zonenreferenzmodell definiert, um die Vorzüge dieses Architekturstils in Bezug auf Data Science zu untersuchen.

Diese Arbeit beschreibt daher grundlegende Konzepte im Bereich *Big Data*, *Data Science* und *Data Lakes*. Neben Techniken und digitalen „Werkzeugen“ zur effektiven Analyse von Daten in Big-Data-Größenordnung wurde ebenfalls der konzeptionelle Aufbau von Data Lakes skizziert. Das Zonenreferenzmodell von Giebler et al. [4] charakterisiert hierbei stellvertretend zonenbasierte Data-Lake-Architekturen. Dieses Modell legt einen besonderen Fokus auf die Rolle von Data Scientisten und weist ihnen neben erweiterten Zugriffsmöglichkeiten auf die einzelnen Bereiche des Data Lakes ebenfalls eine separierte Zone als Testumgebung für die explorative Datenanalyse zu. Im Rahmen dieser Arbeit konnte hier angeknüpft werden, um die praktischen Anforderungen von Data Scientisten anhand der Durchführung typischer Anwendungsfälle zu untersuchen.

Die definierten allgemeinen Anforderungen zur Realisierung von Data-Science-Anwendungsfällen wurden in vier Bereiche unterteilt. Diese umfassen die Definition der konkreten Frage- und Problemstellung, die Bereitstellung qualitativ hochwertiger Daten, die explorative Datenanalyse, sowie die Gewinnung von Wissen und der Visualisierung der Ergebnisse. Die praktische Umsetzbarkeit dieser Anforderungen konnte anhand drei beispielhafter Anwendungsfälle entlang des Produktlebenszyklus untersucht werden. Diese Beispiele umfassen die Bereiche Marktbeobachtung, Marktforschung und die Erstellung von Prognosen für zukünftige Entwicklungen.

Um eine Basis für die spätere protoypische Implementierung und deren Evaluation zu schaffen, wurden in Kapitel 5 allgemeinen Konzepte mit Bezug auf den betrachteten Datenanalyseprozess diskutiert. Anhand des Prozesses zur *Knowledge Discovery in Databases* (kurz: KDD-Prozess) konnte eine Vorgehensweise zur Gewinnung von Wissen aus Daten im Umfeld von Data Lakes abgeleitet werden. Dabei zeigt sich, dass die einzelnen Schritte des Datenanalyseprozesses gut durch die Zonenmodelle von Data Lakes, aufgrund ihrer Prägung durch praktische Einflüsse in der Entwicklung, abgebildet werden können. Hierbei stellte sich zusätzlich heraus, dass sich die Arbeitsgebiete der fachlichen Spezialisten im Bereich Data Science (Data Scientisten, Data Engineers, Data Analysts) ebenfalls auf die Bereiche des Datenanalyseprozesses übertragen lassen. Die unterschiedlichen Rollen und Arbeitsweisen der Data-Science-Spezialisten im Umgang mit zonenbasierten Data Lakes konnten daraus abgeleitet werden.

Anschließend wurden die vorab definierten Anforderungen und Konzepte anhand einer prototypischen Implementierung betrachtet, um Erfahrungen zu ihrer Umsetzbarkeit sammeln zu können. Im Rahmen der Umsetzung wurde zwischen einem zonenbasierten und einem zonenlosen Aufbau des zugrundeliegenden Data Lakes unterschieden.

Zur Bewertung der Kundenzufriedenheit wurde im Anwendungsfall der Marktforschung als Simulation des Datenanalyseprozesses eine schrittweise Durchführung aller Verarbeitungsschritte beschrieben. Die Speicherung der Zwischenergebnisse von einzelnen Verarbeitungsschritten benötigte zusätzlichen Rechen- und Speicheraufwand. Jedoch können somit vor allem zukünftige, wiederholte Berechnungen und sogar vollständige Prozessschritte profitieren, indem ohne zusätzlichen Rechenaufwand direkt auf die vorverarbeiteten Daten zugegriffen wird. Dies resultiert in einer signifikant niedrigeren Laufzeit und der Reduktion der nötigen Rechenleistung. Verstärkt wurde dieser Effekt bei Durchführung des zweiten Anwendungsfalls. Hierzu wurden zum Zweck der Marktbeobachtung die Trends der monatlichen Twitter-Nachrichten über die Top-Unternehmen des NASDAQ in Monatsintervalle gruppiert. Durch die Annahme, dass die berechneten Daten aus der Marktforschung bereits im Data Lake vorliegen, konnte mittels dieser historischen Daten der Datenanalyseprozess derart verkürzt werden, dass direkt mit der explorativen Analyse eingestiegen werden konnte.

Die geordnete Speicherung der Zwischenergebnisse fördert darüber hinaus eine Struktur der Zusammenarbeit, da die Bearbeitung der jeweiligen Fragestellungen unkompliziert auf die einzelnen Data-Science-Spezialisten verteilt werden kann. Durch die Zuweisung der Zuständigkeiten innerhalb eines Projektes wären die jeweiligen Rollen im Datenanalyseprozess klar aufgeteilt und der Austausch der beteiligten Partien würde über die Bereitstellung der verarbeiteten Daten geschehen. Für diesen Austausch der Zwischenstände könnte beispielsweise ein neuer, unabhängiger *Transfer*-Bereich definiert werden. Dieser würde die Zusammenarbeit erleichtern und könnte bei auftretenden Schwierigkeiten während der Datenanalyse zum unkomplizierten Austausch der Beteiligten genutzt werden.

Abschließend lässt sich zusammenfassen, dass die gestellten Anforderungen an die Durchführung von Data-Science-Anwendungsfällen in zonenbasierten Data Lakes sowohl durch den Aufbau, als auch ihre praktische Handhabung des Data Lakes vollumfänglich erfüllt sind. Im Folgenden wird ein Ausblick über Themen gegeben, welche im Umfang dieser Arbeit nicht weiter untersucht wurden, jedoch aus den gewonnenen Erkenntnissen hervorgehen.

Ausblick

Die prototypische Implementierung im Rahmen dieser Ausarbeitung fokussierte sich auf Umsetzbarkeit und die Anforderungen von Data-Science-Anwendungsfällen in zonenbasierten Data Lakes. Daher würde zukünftige Forschung davon profitieren, die Erkenntnisse dieser beispielhaften Implementierung durch ein konkretes Praxisbeispiel mit *Big-Data*-Umfang zu evaluieren.

Darüber hinaus gilt es, Rahmenbedingungen für die Kooperation der spezialisierten Anwender während der Datenanalyse zu untersuchen. Mögliche Vorteile durch den Aufbau einer strukturierten Datenhaltung in zonenbasierten Data Lakes sowie dem Potential hinsichtlich der Zusammenarbeit verschiedener Beteiligter entlang des Datenanalyseprozesses wurden bei der Evaluation der Implementierung thematisiert. Dies konnte im Rahmen dieser Arbeit jedoch nicht weiter vertieft werden. Daher würde das Thema von zusätzlicher Forschung in diesem Bereich profitieren. Eine Erweiterung des Konzepts der Zonenmodelle, um eine Komponente der Zusammenarbeit in der Anwenderebene, ist hierbei denkbar. Gerade in der modernen Wirtschaft könnte dieser Aspekt durch den wachsenden Einfluss von agilen Arbeitsmethoden und der verteilten, teamübergreifenden Bearbeitung von Aufgaben auf Interesse stoßen.

Zusätzlich wäre es ebenfalls von Interesse, das Thema der Zugriffsbeschränkungen innerhalb zonenbasierter Data Lakes für Data-Science-Spezialisten zu präzisieren. Auf diese Weise kann eine striktere Handhabung der umfangreichen Berechtigungen von Data Scientisten, basierend auf den jeweiligen Aufgabengebieten der Spezialisten, angewendet werden.

Literaturverzeichnis

[1] J. Rydning, D. Reinsel, J. Gantz. „The digitization of the world from edge to core“. In: *Framingham: International Data Corporation* (2018).

[2] T. H. Davenport, D. J. Patil. „Data scientist: the sexiest job of the 21st century“. In: *Harvard Business Review* (Okt. 2012).

[3] J. Dixon. *Pentaho, Hadoop, and Data Lakes*. Okt. 2010. URL: https://jamesdixon.wordpress.com/2010/10/14/pentaho-hadoop-and-data-lakes.

[4] C. Giebler, C. Gröger, E. Hoos, H. Schwarz, B. Mitschang. „A Zone Reference Model for Enterprise-Grade Data Lake Management“. In: *2020 IEEE 24th International Enterprise Distributed Object Computing Conference (EDOC)*. IEEE. 2020, S. 57–66.

[5] N. Miloslavskaya, A. Tolstoy. „Big data, fast data and data lake concepts“. In: *Procedia Computer Science* 88.300-305 (2016), S. 63.

[6] A. McAfee, E. Brynjolfsson. „Big data: the management revolution“. In: *Harvard business review* 90.10 (2012), S. 60–68.

[7] M. Chen, S. Mao, Y. Liu. „Big data: A survey“. In: *Mobile networks and applications* 19.2 (2014), S. 171–209.

[8] P. Russom et al. „Big data analytics“. In: *TDWI best practices report, fourth quarter* 19.4 (2011), S. 1–34.

[9] Oracle Cloud Blog. *Hadoop heißt, Big Data endlich in den Griff zu kriegen*. Nov. 2018. URL: https://blogs.oracle.com/de-cloud/was-ist-hadoop.

[10] J. Dean, S. Ghemawat. „MapReduce: simplified data processing on large clusters“. In: *Communications of the ACM* 51.1 (2008), S. 107–113.

[11] K. Shvachko, H. Kuang, S. Radia, R. Chansler. „The hadoop distributed file system“. In: *2010 IEEE 26th symposium on mass storage systems and technologies (MSST)*. IEEE. 2010, S. 1–10.

[12] A. Holst. *Volume of data/information created, captured, copied, and consumed worldwide from 2010 to 2024*. Feb. 2021. URL: https://www.statista.com/statistics/871513/worldwide-data-created/.

[13] C. Schumann, P. Zschech, A. Hilbert. „Das aufstrebende Berufsbild des Data Scientist“. In: *HMD Praxis der Wirtschaftsinformatik* 53.4 (2016), S. 453–466.

[14] G. DeSantis. *Data Scientist vs. Data Analyst vs. Data Engineer*. Juli 2019. URL: https://medium.com/@gdesantis7/data-scientist-vs-data-analyst-vs-data-engineer-bd4868f9b31e.

[15] F. Nurfikri. *Data Engineer, Data Science and Data Analyst — What the Difference?* Mai 2020. URL: https://towardsdatascience.com/data-engineer-data-science-and-data-analyst-what-the-difference-8f31eec127dc.

[16] C. Giebler, C. Gröger, E. Hoos, H. Schwarz, B. Mitschang. „Leveraging the Data Lake: Current State and Challenges“. In: *International Conference on Big Data Analytics and Knowledge Discovery*. Springer. 2019, S. 179–188.

[17] I. G. Terrizzano, P. M. Schwarz, M. Roth, J. E. Colino. „Data Wrangling: The Challenging Yourney from the Wild to the Lake.“ In: *CIDR*. 2015.

[18] J. Dixon. *Data Lakes Revisited*. Sep. 2014. URL: https://jamesdixon.wordpress.com/2014/09/25/data-lakes-revisited.

[19] T. Marschall, H. Baars. „Pi-Architektur“. In: *Online Special Self-Service Data Preparation* (2017).

[20] H. Fang. „Managing data lakes in big data era: What's a data lake and why has it became popular in data management ecosystem“. In: *2015 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)*. IEEE. 2015, S. 820–824.

[21] A. Gorelik. *The enterprise big data lake: Delivering the promise of big data and data science*. O'Reilly Media, 2019.

[22] F. Ravat, Y. Zhao. „Data lakes: Trends and perspectives“. In: *International Conference on Database and Expert Systems Applications*. Springer. 2019, S. 304–313.

[23] V. Wisdom, R. Gupta. „An introduction to twitter data analysis in python“. In: *Artigence Inc* (2016).

[24] S. B. Mane, Y. Sawant, S. Kazi, V. Shinde. „Real time sentiment analysis of twitter data using hadoop“. In: *IJCSIT) International Journal of Computer Science and Information Technologies* 5.3 (2014), S. 3098–3100.

[25] A. Go, R. Bhayani, L. Huang. „Twitter sentiment classification using distant supervision“. In: *CS224N project report, Stanford* 1.12 (2009), S. 2009.

[26] A. Agarwal, B. Xie, I. Vovsha, O. Rambow, R. J. Passonneau. „Sentiment analysis of twitter data“. In: *Proceedings of the workshop on language in social media (LSM 2011)*. 2011, S. 30–38.

[27] M. Madsen. „How to build an enterprise data lake: Important considerations before jumping in“. In: *Third Nature Inc* (2015), S. 13–17.

[28] B. Sharma. *Architecting data lakes: Data management architectures for advanced business use cases*. O'Reilly Media, 2018.

[29] J. Hardin, R. Hoerl, N. J. Horton, D. Nolan, B. Baumer, O. Hall-Holt, P. Murrell, R. Peng, P. Roback, D. Temple Lang et al. „Data science in statistics curricula: Preparing students to “think with data”“. In: *The American Statistician* 69.4 (2015), S. 343–353.

[30] M. Tiedemann. *Was ist Data Science?* Mai 2018. URL: https://www.alexanderthamm.com/de/blog/data-science-schluessel-der-digitalen-transformation/.

[31] J. Li, F. Tao, Y. Cheng, L. Zhao. „Big data in product lifecycle management“. In: *The International Journal of Advanced Manufacturing Technology* 81.1 (2015), S. 667–684.

[32] T. A. Oliva, R. L. Oliver, W. O. Bearden. „The relationships among consumer satisfaction, involvement, and product performance: A catastrophe theory application“. In: *Behavioral Science* 40.2 (1995), S. 104–132.

[33] U. Fayyad, G. Piatetsky-Shapiro, P. Smyth. „From data mining to knowledge discovery in databases“. In: *AI magazine* 17.3 (1996), S. 37–37.

[34] J. L. Reyes-Ortiz, L. Oneto, D. Anguita. „Big Data Analytics in the Cloud: Spark on Hadoop vs MPI/OpenMP on Beowulf“. In: *Procedia Computer Science* 53 (2015), S. 121–130. DOI: https://doi.org/10.1016/j.procs.2015.07.286.

[35] R. S. Xin, J. Rosen, M. Zaharia, M. J. Franklin, S. Shenker, I. Stoica. „Shark: SQL and rich analytics at scale“. In: *Proceedings of the 2013 ACM SIGMOD International Conference on Management of data*. 2013, S. 13–24.

[36] A. K. Dutta, R. Hasan. „How much does storage really cost? Towards a full cost accounting model for data storage“. In: *International Conference on Grid Economics and Business Models*. 2013, S. 29–43.

Alle URLs wurden zuletzt am 15. 04. 2021 geprüft.

Erklärung

Ich versichere, diese Arbeit selbstständig verfasst zu haben. Ich habe keine anderen als die angegebenen Quellen benutzt und alle wörtlich oder sinngemäß aus anderen Werken übernommene Aussagen als solche gekennzeichnet. Weder diese Arbeit noch wesentliche Teile daraus waren bisher Gegenstand eines anderen Prüfungsverfahrens. Ich habe diese Arbeit bisher weder teilweise noch vollständig veröffentlicht. Das elektronische Exemplar stimmt mit allen eingereichten Exemplaren überein.

Murrhardt, 15.04.2021

Ort, Datum, Unterschrift