Kursanalyse von Kryptowährungen mit Azure Machine Learning

Price analysis of cryptocurrencies using Azure Machine Learning

Abschlussarbeit

zur Erlangung des akademischen Grades
Master of Science

vorgelegt von

Sebastian Lischewski

geboren am 08.08.1991 in Rosenheim
Matrikelnummer: 04326912

München, den 6. Januar 2018

Prüfer: Prof. Dr. Patrick Möbert, Hochschule München

Erklärung

Hiermit erkläre ich, dass ich die Bachelorarbeit selbständig verfasst, noch nicht anderweitig für Prüfungszwecke vorgelegt, keine anderen als die angegebenen Quellen oder Hilfsmittel benutzt sowie wörtliche und sinngemäße Zitate als solche gekennzeichnet habe.

Ort, Datum Unterschrift

Zusammenfassung

ZIEL
In der vorliegenden Arbeit werden Einflussfaktoren auf den Kurs von ausgewählten Kryptowährungen (Bitcoin [BTC] und Ethereum [ETH]) gesucht und mit Hilfe von Machine Learning versucht, herauszufunden, ob sich die Kurse der digitalen Währungen voraussagen lassen. Das Machine Learning wird mit dem Werkzeug Azure Machine Learning Studio von Microsoft durchgeführt.

MOTIVATION
Es besteht in der Informationstechnik großes Interesse an Kryptowährungen und der zugrundeliegenden Technik (Blockchain). Vorteile sind die einfache Benutzung, die Sicherheit, die Anonymität und die Möglichkeit der Integration in das Internet der Dinge (engl. Internet of Things, IoT), zum Beispiel in Form von Smart Contracts.
Ebenfalls große Bedeutsamkeit kommt dem interdisziplinären Themengebiet des Machine Learning zu. Dies zeigen bekannte Projekte wie Googles DeepMind, IMBs Watson oder Sprachassistenten wie Apples Siri, Amazons Alexa oder Samsungs Bixby.
Es wird ein Clouddienst, genauer eine Software as a Service (SaaS) für das Machine Learning genutzt, da dieser Sektor in den letzten 10 Jahren um 700% gewachsen ist.

VORGEHEN
Zuerst werden im Theorieteil die Grundlagen (Data Mining Frameworks, Machine Learning, Kryptowährungen und Azure Machine Learning Studio) beschrieben. Hier wird die Wahl getroffen, die Analyse mit dem CRISP-DM-Referenzmodell durchzuführen.
Anschließend folgt der Praxisteil. Es werden die ersten fünf Phasen des Referenzmodells durchlaufen. Die letzte Phase (Deployment) wird weggelassen.

GENUTZTE EINFLUSSFAKTOREN
Folgende Einflussfaktoren werden für die Untersuchung genutzt:

- Kryptowährungs-eigene Faktoren (Handelsvolumen, Erzeugungsschwierigkeit, Preis am Vortag, Anzahl der Transaktionen etc.)
- öffentliches Interesse (Google suchen, Google Newssuchen, Wikipedia Seitenaufrufe, Zeitungsüberschriften)
- Aktienindices (aus verschiedenen Wirtschaftsregionen und unterschiedlichen Sektoren)
- der Preis für natürliche Ressourcen (zwei Ölsorten, Gold, Silber)
- historische Kursdaten für den BTC- und ETH-Kurs

ERGEBNISSE DES MACHINE LEARNING

Algorithmus	**F1-Score**	**Accuracy**	**Precision**	**Recall**	**AUC**
Two-Class Support Vector Machine	0.565445	0.538889	0.568421	0.562500	0.552703
Two-Class Neural Network	0.685259	0.561111	0.554839	0.895833	0.571181
Two-Class Logistic Regression	0.559585	0.527778	0.556701	0.562500	0.568204
Two-Class Locally-Deep Support Vector Machine	0.547264	0.494444	0.52381	0.572917	0.517361
Two-Class Decision Jungle	0.698413	0.577778	0.564103	0.916667	0.553571
Two-Class Decision Forest	0.556818	0.566667	0.6125	0.510417	0.573289
Two-Class Boosted Decision Tree	0.514620	0.538889	0.586667	0.458333	0.570188
Two-Class Bayes Point Machine	0.684211	0.533333	0.535294	0.947917	0.595982
Two-Class Averaged Perceptron	0.522727	0.533333	0.575000	0.479167	0.573289

Tabelle 0.1: Ergebnisse des Machine Learning: Ethereum Two-class Classification

Algorithmus	**F1-Score**	**Accuracy**	**Precision**	**Recall**	**AUC**
Two-Class Support Vector Machine	0.662937	0.552045	0.578049	0.777049	0.499951
Two-Class Neural Network	0.706199	0.594796	0.599542	0.859016	0.612805
Two-Class Logistic Regression	0.623946	0.585502	0.642361	0.606557	0.597031
Two-Class Locally-Deep Support Vector Machine	0.647555	0.611524	0.666667	0.629508	0.643130
Two-Class Decision Jungle	0.659236	0.602230	0.640867	0.678689	0.639724
Two-Class Decision Forest	0.656958	0.605948	0.648562	0.665574	0.650770
Two-Class Boosted Decision Tree	0.668750	0.605948	0.638806	0.701639	0.649595
Two-Class Bayes Point Machine	0.734082	0.604089	0.592742	0.963934	0.583691
Two-Class Averaged Perceptron	0.566957	0.537175	0.603704	0.534426	0.564160

Tabelle 0.2: Ergebnisse des Machine Learning: Bitcoin Two-class Classification

Algorithmus	R^2	**MAE**	**RMSE**	**RAE**	**RSE**
Neural Network Regression	-0.232605	69.99189	124.834987	0.737855	1.232605
Boosted Decision Tree Regression	0.999326	1.372442	2.918441	0.014468	0.000674
Decision Forest Regression	0.994616	3.514041	8.250093	0.037045	0.005384
Bayesian Linear Regression	0.999994	0.206625	0.272996	0.002178	0.000006

Tabelle 0.3: Ergebnisse des Machine Learning: Ethereum Regression

Algorithmus	R^2	**MAE**	**RMSE**	**RAE**	**RSE**
Neural Network Regression	-0.105524	496.619518	717.756528	1.225332	1.105524
Boosted Decision Tree Regression	0.995827	10.095783	44.099144	0.02491	0.004173
Decision Forest Regression	0.995791	13.301142	44.290157	0.032819	0.004209
Bayesian Linear Regression	0.999946	2.273447	5.032831	0.005609	0.000054

Tabelle 0.4: Ergebnisse des Machine Learning: Bitcoin Regression

Die Regressionen haben einen zu hohen Wert für R^2, deswegen ist es falsch eine Rangliste aufzustellen. Bei der Klassifikation gibt es folgendes Ranking (der beste Wert für den F1-Score ist 1, der Schlechteste 0):

Algorithmus	F1-Score
Two-Class Decision Jungle	0.698413
Two-Class Neural Network	0.685259
Two-Class Bayes Point Machine	0.684211

Tabelle 0.5: Rangliste der besten Ethereum Two-Class Classification Algorithmen

Algorithmus	F1-Score
Two-Class Bayes Point Machine	0.734082
Two-Class Neural Network	0.706199
Two-Class Boosted Decision Tree	0.668750

Tabelle 0.6: Rangliste der besten Bitcoin Two-Class Classification Algorithmen

Es lässt sich Schlussfolgern, dass Kurs mit den den angewandten Mitteln und gefunden Einflussfaktoren nicht vorher gesagt werden kann. Es wird empfohlen, weniger Einflussfaktoren zu nutzen und dafür diese genauer zu betrachten. Der Kurs des Bitcoins scheint stärker von Faktoren beeinflusst zu werden, als der des Ethers.

BEWERTUNG DES CRISP-DM-REFERENZMODELLS
CRISP-DM ist ein sehr vielseitiges Referenzmodell. Es bietet durch den User Guide eine große Hilfestellung für weniger Erfahrene. In leicht angepasster Version kann es auf eine Vielzahl an Projekten aus den Bereichen Data Mining, Machine Learning, Data Science oder Analytics angewandt werden.

BEWERTUNG DES AZURE MACHINE LEARNING STUDIOS
Bei Azure Machine Learnung Studio handelt es sich um ein Werkzeug, dass alle Grundfunktionen bereitstellt. Es ist selbst durch R- oder Python-Skripte erweiterbar. Es besitzt eine aktuelle und ausreichende Dokumentation, jedoch keine Versionsverwaltung. Es wird für Vorstudien, Evaluationsprojekte und zum Lernen, nicht aber für produktive Systeme empfohlen.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Listings

Abkürzungen und Erklärungen

R^2 (adjusted) R squared; R-Quadrat; adjustiertes oder angepasstes Bestimmtheitsmaß; statistische Größe bei der Bewertung einer Regression.

Altcoins Alternative Kryptowährung zum Bitcoin.

ANN artifical neuronal network; Künstliches neuronales Netze oder künstliches neuronales Netzwerk.

BRL Real; brasilianische Währung.

BTC Abkürzung für die Kryptowährung Bitcoin.

CNY Renminbi Yuan; chinesische Währung.

CRISP-DM CRoss Industrial Standard Process for Data Mining.

CSV comma seperated values.

EDA Explorative Data Analysis; Explorative Datenanalyse.

ETH Abkürzung für die Kryptowährung Ethereum.

Ether Währungsbezeichnung im Ethereumnetzwerk.

EUR Euro; Währung der Europäischen Wirtschafts- und Währungsunion.

Features Spalten eines Datensatzes; attributes; Attribute; columns.

GBP Pfund Sterling; britisches Pfund; Währung des Vereinigten Königreichs.

IDE Integrated Development Environment.

INR Indische Rupie; indische Währung.

JPY Yen; japanische Währung.

kaggle populäres Data Science Portal.

KDD Knowledge Discovery (in) Databases .

KDnuggets Webseite zum Lernen von und für Diskussionen über Data Science.

MICE Multivariate Imputation by Chained Equations.

ML Machine Learning; Maschinelles lernen.

Model von der englischen Bezeichung für ein mathematisches Modell.

Mrd. Milliarden, 1.000.000.000.

Observations Beobachtungen; Reihen/Zeilen eines Datensatzes; records, rows.

PCA Principal component analysis; Hauptkomponentenanalyse.

Python Programmiersprache.

R Programmiersprache.

SaaS Software as a Service.

SEMMA Sample, Explore, Modify, Model and Assess.

SOA service oriented atchitekture, Service-orientierte Architektur.

STLFSI St. Louis Fed Financial Stress Index.

test set Teil eines Datensatzes, das zum Testen eines trainierten Models eingesetzt wird.

train set Teil eines Datensatzes, das zum Trainieren eines Models genutzt wird.

TSV tab seperated values.

USD US-Dollar, United States Dollar.

1 Einletung und Motivation

1.1 Thema der Arbeit

In der vorliegenden Arbeit werden Einflussfaktoren auf den Kurs von ausgewählten Kryptowährungen gesucht und und mit Hilfe von Machine Learning versucht, herauszufinden, ob sich der Kurs der digitalen Währungen voraussagen lässt. Im nachfolgenden Kapitel wird auf die Motivation hinter der Analyse eingegangen. Das genaue Vorgehen und die Ziele werden in Kapitel 2 erläutert.

1.2 Bitcoin als Vorreiter der Krypotowährungen

Geld online von einem Teilnehmer direkt zu einem Anderen senden, ohne dabei (Transaktions-) Gebühren für einen zwischengelagerten Finanzdienstleister zahlen zu müssen, ist der Gedanke hinter dem „Peer-To-Peer Electronic Cash System“(Nakamoto, 2008) Bitcoin. Obwohl es Teilnehmern ohne Aufwand möglich ist, dem Netzwerk beizutreten oder es wieder zu verlassen, ist es solange unangreifbar, solange ein Angreifer nicht dauerhaft über mehr Rechenkapazität verfügt, als das komplette restliche Netzwerk.(Nakamoto, 2008) Ob das Bitcoinnetzwerk wirklich absolute Anonymität gewährt, wird stark kritisiert.(Reid and Harrigan, 2013; Androulaki et al., 2013). In der Tat werden beim Nutzen des Netzwerks jedoch keine persönlichen Informationen an ein Kreditinstitut (wie PayPal, Paydirekt, ApplePay oder Masterpass) weitergegeben. Diese Argumente (Kostenreduktion, Sicherheit und Anonymität) sorgen für Interesse an der digitalen Währung (auch hier gibt es Kritiker, die den Bitcoin als Investition und nicht als Währung bezeichnen)(Baur et al., 2015). Nicht zu vernachlässigen ist an dieser Stelle auch das Interesse der Industrie an „Smart Contracts“(Dannen, 2017, S. 10), die beispielsweise im Bereich des Internet of Things Anwendung finden.(Christidis and Devetsikiotis, 2016)

Neben Bitcoin hat sich deshalb zusätzlich eine Vielzahl an anderen sogenannten Kryptowährungen entwickelt. Die Währungen mit dem größten Marktvolumen sind Bitcoin und Ethereum (Wood, 2014).(Brandt, 2017; CoinMarketCap, 2017) Daneben gibt es noch sogenannte Altcoins (aus dem Englischen: alternative coin(Bajpai, 2014)). Zum Zeitpunkt dieser Arbeit umfassen diese 664 Bitcoin-Alternativen.(CoinDesk, 2017). Obgleich die tatsächliche Nutzung der Krypowährungen sehr gering ist (1% der Befragten in Deutschland(TSYS, 2016)), steigt das Interesse an Kryptowährungen(WikiTrends, 2017; GoogleTrends, 2017).
Genauere Informationen zu Kryptowährungen und technischen Eigenschaften werden in Punkt 5 behandelt.

1.3 Gemeinsamkeiten und Unterschiede von Machine Learning, Data Mining, Data Analysis und Data Science

Die Themen Machine Learning, Data Mining, Data Analysis und Data Science sind verwandte Begriffe aus dem interdisziplinären Bereich der Statistik und Informatik.
Machine Learning gehört in der Informatik und Mathematik zur Familie der Künstlichen Intelligenz.(Kim, 2017, S. 2; Swamynathan, 2017, S. 54). Es kann als „Sammlung von Algorithmen und Techniken“ verstanden werden, die „genutzt werden, um Computersysteme zu erstellen, die aus Daten lernen, um Vorhersagen zu erstellen“.(Swamynathan, 2017, S. 53; eigene Übersetzung) Bekannte Anwendungen aus dem Alltag sind Empfehlungssysteme oder Spamerkennungen.(Swamynathan, 2017, S. 53)
Data Mining beschreibt den Prozess, aus einer gewaltigen Menge an Daten die „richtigen Daten“, zur „richtigen Zeit“ für die „richtigen Entscheidungen“(Swamynathan, 2017, S. 61; eigene Übersetzung) zu gewinnen. Um diesen Prozess haben sich im Laufe Zeit drei Frameworks gebildet(Swamynathan, 2017, S. 69):

- Knowledge Discovery Databases (KDD) process model
- CRoss Industrial Standard Process for Data Mining (CRISP – DM)
- Sample, Explore, Modify, Model and Assess (SEMMA)

Neben Schnittmengen mit Künstlicher Intelligenz, Machine Learning und der Statistik, befasst sich Data Mining ebenfalls mit Datenbanksystemen.(Ramasubramanian and Singh, 2017, S. 4)

Eng verwandt mit dem Data Mining ist die Datenanalyse (engl. Data Analysis; in der Industrie auch Business Analytics(Swamynathan, 2017, S. 58)). Sie wird benutzt um(Hertle, 2016, S. 2; Teil 1)

1. Messdaten zu verstehen,
2. Gesetzmäßigkeiten zu extrahieren und
3. die Zukunft vorherzusagen.

Dazu bedient sie sich der deskriptiven Statistik, der explorativen Datentenanalyse (engl. Explorative Data Analysis; EDA) und der induktiven Statistik.(Hertle, 2016, S. 17)
Um

- den Anstieg der Datenmengen in der Datenanalyse,
- die Veränderung im Aussehen der Daten (unstrukturiert oder semi-strukturiert statt strukturiert) und
- die Wandlung in der Semantik der zugrundeliegenden Daten (Daten liegen in Markup-Sprachen vor und enthalten zusätzliche Informationen)

darzustellen, hat sich der Begriff Data Science entwickelt.(Dhar, 2013) Er versucht die geänderten Anforderungen der heutigen Datenanalyse abzubilden (siehe Abbildung 1.1).

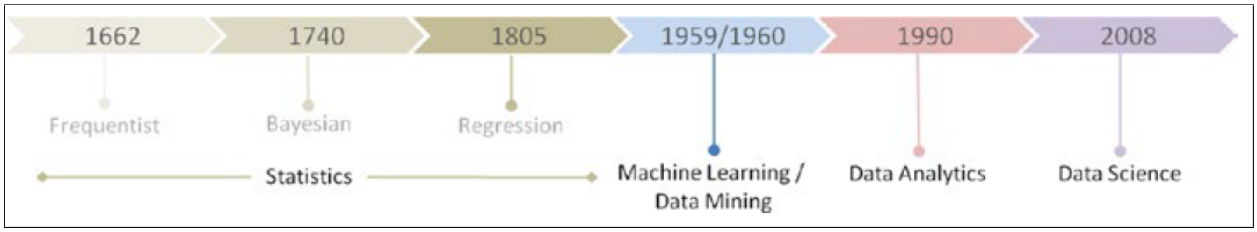

Abbildung 1.1: Learn from data evolution (Swamynathan, 2017, S. 66)

Wie anfänglich erwähnt, sind alle genannten Begriffe miteinander verwandt. Das Gewinnen von Erkenntnissen aus Daten, um beispielsweise die Zukunft vorherzusagen, nennt sich Data Analysis. Werden die Daten aus verschiedensten Datenbanken oder Datawarehouses gewonnen, spricht man von Data Mining. Handelt es sich dabei noch um Informationen unterschiedlicher Struktur und große Datensätze, so befindet man sich im Bereich der Data Science. Der inhärente Erkenntnisgewinn dieser Verfahren kann von von menschlicher Seite kommen oder durch Machine Learning geschehen.

Projekte wie Googles DeepMind(DeepMind, 2017b), IBMs Watson(IBM, 2017) oder Sprachassistenten wie Siri, Alexa und Bixby zeigen, dass großes Interesse an Machine Learning und Data Science herrscht. Deshalb haben sich ganze Berufsfelder wie „machine learning engineer“, „data engineer“ oder „data scientist“(Ramasubramanian and Singh, 2017, S. 1) gebildet.

1.4 Cloud-Dienste und SaaS

Cloud Computing beschreibt „ein Modell, das es erlaubt bei Bedarf, jederzeit und überall bequem über ein Netz auf einen geteilten Pool von konfigurierbaren Rechnerressourcen (z. B. Netze, Server, Speichersysteme, Anwendungen und Dienste) zuzugreifen, die schnell und mit minimalem Managementaufwand oder geringer Serviceproviderinteraktion zur Verfügung gestellt werden können“(Appelrath et al., 2014, S. 18). Innerhalb des Cloud Computing unterscheidet man weiterhin zwischen verschiedenen Cloud-Diensten (engl. cloud services). Nach (Appelrath et al., 2014, S. 20) differenziert man zwischen den Services in Tabelle 1.1.

Diensttyp	Beschreibung
Infrastructure as a Service (IaaS)	Virtuelle Hardware oder Infrastruktur, zum Beispiel Speicherplatz, Rechenleistung oder Netzwerkbandbreite
Platform as a Service (PaaS)	Programmierframeworks, Bibliotheken und Werkzeuge, um Anwendungen unter eigener Kontrolle auf Cloud-Infrastrukturen bereitstellen zu können, ohne die zugrunde liegende Infrastruktur wie Netzwerk, Server, Betriebssysteme oder Speicher managen oder kontrollieren zu müssen
Software as a Service (SaaS)	Vollständige Anwendungen, die auf Cloud-Infrastrukturen betrieben und beispielsweise über einen Webbrowser aufrufbar sind, wobei Nutzer weder die zugrunde liegende Cloud-Infrastruktur noch individuelle Anwendungseinstellungen (mit der möglichen Ausnahme der eingeschränkten Konfiguration von Nutzereinstellungen) kontrollieren müssen und können
Mashup as a Service (MaaS)	Verknüpfung einzelner Software-Komponenten (unter anderem auch Cloud-Dienste) zu einem aggregierten Cloud-Dienst
Business Process as a Service (BPaaS)	Konkrete Geschäftsanwendungen (beispielsweise CRM) als Verknüpfung einzelner Software-Komponenten (standardisierte MaaS)

Tabelle 1.1: Cloud-Diensttypen

(Appelrath et al., 2014, S. 23) sprechen generell von „Cloud Computing als disruptiver Innovationsfaktor“. In der vorliegenden Arbeit wird besonders Software as a Service betrachtet. Dort stieg der Umsatz von 10,75 Mrd. USD im Jahr 2010 auf 38,57 Mrd. USD im Jahr 2016. Für die Zukunft (2020) wird sogar ein Umsatz von 75,73 Mrd. USD prognostiziert.(Gartner, 2017) Das ist eine Steigerung von über 700% in nur 10 Jahren. Dies kann einerseits durch offensichtliche Vorteile, wie „höhere Stabilität und Planungssicherheit“, der „Möglichkeit Anwender schnell ins System einzuführen“ und „Erschließung neuer Kundengruppen“(Fraunhofer, 2010) erklärt werden, andererseits aber auch durch Tendenz der Softwarebranche hin zur serviceorientierten Architekturen (engl. service oriented atchitekture; SOA).(Appelrath et al., 2014, S. 22) Dieser Trend zu SaaS kann beobachtet werden, wenn reine Cloud-Anbieter wie Salesforce „klassi-

sche“ Anbieter wie SAP den Rang als „Spitze des Weltmarkts der Software für Customer Relationship Management (CRM)“(Fritsch, 2013) ablaufen.
Laut einer Studie von (Bitkom and KPMG, 2017) greifen 23% der befragten Unternehmen in Deutschland neben „Office Anwendungen aus der Cloud“, „Security as a Service“ und „Groupware“ auf „Business Intelligence/Big Data“-Software aus der Cloud zurück. Zu dieser Kategorie gehört auch Azure Machine Learning (kurz: Azure ML) von Microsoft, welches zur Analyse in dieser Arbeit verwendet wird.

2 Vorgehen und Ziele

Nachdem in Kapitel 1 das Thema der Arbeit (1.1) vorgestellt und die Motivation dahinter erläutert wurde (1.2 bis 1.4), wird nun der restliche Aufbau vorgestellt. Die nächsten vier Kapitel befassen sich mit den Grundlagen hinter der Arbeit (**Theorie**):

- Kapitel 3 befasst sich mit Data Mining Frameworks. Dabei werden das Knowledge Discovery in Databases (KDD) process model (3.1) und der CRoss Industrial Standard Process for Data Mining (CRISP – DM (3.2) betrachtet. Anschließend folgt eine Entscheidung für CRISP-DM in Punkt 3.3.
- Daraufhin (Kapitel 4) folgen Erklärungen zu den Kategorien des Machine Learning: Supervised Learning (4.1), Unsupervised Learning (4.2), Semi-supervised Learning (4.3), Active Learning (4.4) und Reinforcement Learning (4.5).
- Genauere (technische) Details zu Kryptowährungen sind in Kapitel 5 festgehalten. Hier sind Definitionen zu Begriffen zu finden, die später für die Analyse genutzt werden.
- Den Abschluss des Theorieteils stellt die Allgemeine Beschreibung (6.1), der Aufbau und die Komponenten (6.2) von Microsoft Azure Machine Learning Studio in Kapitel 6 dar.

Im **Praxisteil** in Kapitel 7 werden

- zuerst Ziele und Projektressourcen festgelegt (7.1).
- Daraufhin werden Daten beschafft, beschrieben und untersucht (7.2).
- Nach ihrer Auswahl und Aufbereitung (7.3) folgt
- die Auswahl der Machine Learning Algorithmen, die Festlegung des Testdesigns, Durchführen des Machine Learnings und die Auflistung der Ergebnisse (7.4).

- Die Ergebnisbewertung, der Prozessrückblick und die nächsten Schritte (7.4) stellen den Abschluss dar.

Das Schlusskapitel (8) bewertet die drei Kernelemente der Arbeit:

- das Azure Machine Learning Studio (8.1),
- das CRISP-DM Referenzmodell (8.2) und
- die Ergebnisse des Machine Learning (8.3).

Als Ziel steht über der Arbeit, ob es möglich ist, den Kurs oder Kursschwankungen von Kryptowährungen mit Hilfe von Machine Learning vorauszusagen oder nicht.
Für alle Teile sei angemerkt, dass die Fachliteratur fast ausschließlich in englischer Sprache verfügbar ist. In dieser Arbeit wird deswegen die Entscheidung getroffen, die Fachbegriffe nicht zu übersetzten. Auch die Prozessschritte und -artefakte (Outputs) des CRISP-DM-Referenzmodells sind davon betroffen.

3 Data Mining Frameworks

Wie in Abschnitt 1.3 bereits erwähnt, haben sich um das Data Mining drei bekannte Frameworks entwickelt. Zwei davon (KDD und CRISP-DM) werden im Nachfolgenden genauer beschrieben. Danach wird eines der beiden für die Untersuchung in der Arbeit ausgewählt. Da der SEMMA-Process dem CRIPS-DM Referenzmodell einerseits sehr ähnelt und andererseits genau genommen „keine Data Mining Methodik, sonder eher eine logische Organisation“ des Data Mining „tool set[s]“(EnterpriseMiner, 2012, eigene Übersetzung) ist, wird er hier ausgeklammert.

3.1 Knowledge Discovery in Databases (KDD) process model

Die Bezeichnung Knowledge Discovery in Databases wurde hauptsächlich von (Fayyad et al., 1996) geprägt. Sie beschreiben in ihrer Arbeit ein Problem der 1990er Jahre. Wie auch heute noch, stieg damals die Masse der gespeicherten Daten rapide an. Die manuelle Auswertung dieser Datensätze erforderte mehr Arbeitskraft als vorhanden war. (Fayyad et al., 1996, S. 38) beschreiben es als „data overload“. Deswegen versuchte man, die Prozesse zur Findung von Erkenntnissen zu automatisieren. Daraus hat sich ein Standardvorgehen entwickelt, dass das KDD-Prozessmodell darstellt (siehe Abbildung 3.1).

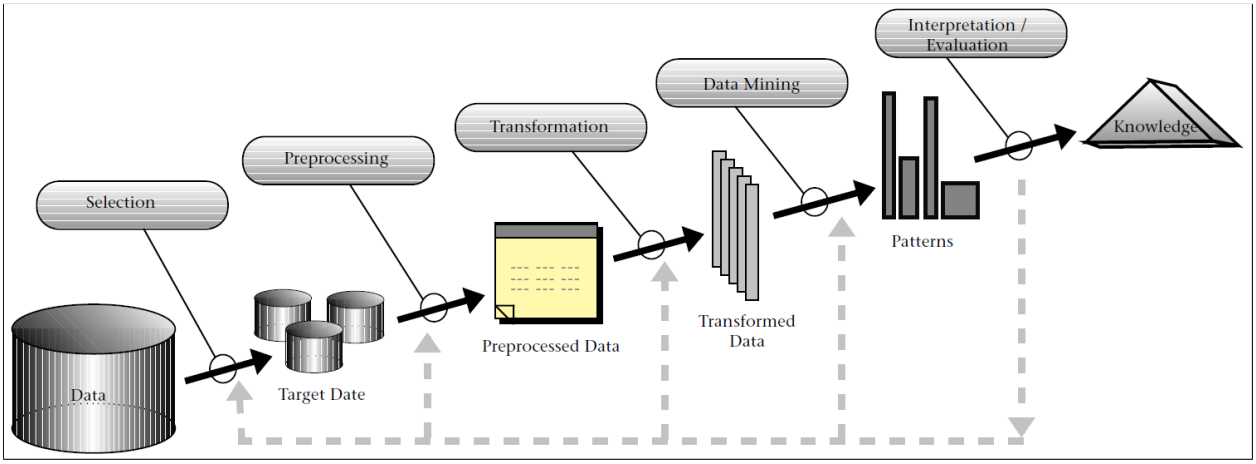

Abbildung 3.1: Ein Überblick über die Schritte des KDD Prozesses nach (Fayyad et al., 1996, S. 41)

3.1.1 Selection

Bevor der erste eigentliche Schritt, die Selektion der Daten, erfolgen kann, ist es unabdingbar, ein „Verständnis für das Anwendungsgebiet zu entwickeln“.(Fayyad et al., 1996, S. 42; eigene Übersetzung) Dies inkludiert auch, Ziele zu setzen und Fragen zu formulieren, die durch das spätere Data Mining (Schritt 3.1.4) beantwortet werden sollen. Ist das Verständnis hergestellt, kann ein „target data set“(Fayyad et al., 1996, S. 42) hergestellt werden. Dabei werden zuerst Daten aus unterschiedlichen - oft heterogenen - Quellen zusammengeführt und dann hinsichtlich des Ziels verdichtet.(Swamynathan, 2017, S. 70)

3.1.2 Preprocessing

Die verbleibende Teilmenge der ursprünglichen Daten muss nun gesäubert und für die nächsten Schritte vorbereitet werden. Dies geschieht, da unbereinigte Daten sowohl den Data Mining-Prozess verschlechtern können (unverlässliche oder falsche Ergebnisse), als auch die Zeit für das Mining deutlich verlängern können.(Swamynathan, 2017, S. 70) Um die Qualität der Daten und des Mining zu verbessern, werden unter anderem folgende Aspekte betrachtet(Fayyad et al., 1996, S. 42; Swamynathan, 2017, S. 70):

Outliner treatment

Ein Ausreißer (engl. outliner) kann beispielsweise ein „Extremer Wert in einer Variablen“ oder ein „Extremer Wert des Residuums bei einer sinnvollen Regression“(Hertle, 2016, S. 25; Teil 5b) sein. Ein Vorgehen für Ausreißer kann folgendermaßen aussehen (nach (Hertle, 2016, S. 25; Teil 5b)):

1. Identifizieren der Ausreißer (evtl. durch eine erste Regression)
2. Interpretation im Sachzusammenhang (Messfehler oder wichtiger Teil der Population)
3. Entscheidung, ob man eine Regression der Daten mit oder ohne diese Ausreißer haben möchte
4. In der Darstellung der Ergebnisse auf die Ausreißer explizit eingehen und Vorgehen erläutern

Noise removal

Auch in einem Datensatz, der auf Ausreißer untersucht wurde, befinden sich immer noch unbekannte, unvollständige, falsche und fehlende Werte („attribute noise“). Zusätzlich können Datenklassen falsch gekennzeichnet sein („class noise“). Ist ein Datensatz von diesen Problemen betroffen, spricht man von „noisy data“. Auf die Lösung dieses Problems wird an dieser Stelle nicht weiter eingegangen.

Identifying duplicated values

Wie oben angesprochen, wird der zu analysierende Datensatz aus mehreren Quellen zusammengeführt. Durch diesen Schritt können Datensätze doppelt (oder noch öfter) vorkommen. Das wird deutlich, wenn man folgendes Beispiel betrachtet:
Über eine Kundenkarte werden Daten von Kunden eines Supermarktes je Filiale gespeichert. Bei einer überregionalen Kundenanalyse tauchen Kunden mehrfach auf, die in verschiedenen Filialen eingekauft haben. Hier ist anzumerken, dass doppelte Werte nicht zwangsläufig gelöscht werden müssen, sie sollten jedoch bei der Analyse bedacht werden.

Check for inconsistency

Je größer ein Datensatz ist, umso wahrscheinlicher enthält der Inkonsistenzen. Dies wird ebenfalls durch die Fusion von mehreren Quellen verstärkt (Beispiel: unterschiedliches Alter für einen Kundenstammsatz). Auch hier muss geprüft werden, wie mit diesen Werten umzugehen ist. Eventuell können Regeln festgelegt werden wie 'immer der neuste Datenpunk ist der richtige'.

Time series and changes

Der letzte Punkt, der beim Preprocessing betrachtet werden muss, ist der Zusammenhang der Daten mit dem Erfassungszeitpunkt. So können sich im Laufe der Zeit die Messmethodik (z.B. andere Sensoren), die Messgenauigkeit (z.B. bessere Sensoren) oder die Abstände der Messungen verändern. Das wiederum kann zu ungleich verteilten Datensätzen oder inkonsistenter Genauigkeit führen.

3.1.3 Transformation

Der letzte Schritt vor dem eigentlichen Data Mining ist die Transformation. In diesem Prozessschritt geht es darum, „mit Dimensionsreduktions- oder -transformationsmethoden die effektive Anzahl an Variablen [...] zu reduzieren“(Fayyad et al., 1996, S. 42; eigene Übersetzung). Dies geschieht beispielsweise durch das identifizieren und eliminieren invarianter Variablen. Ebenfalls wird versucht, solche Variablen zu finden, die mehrere Andere repräsentieren. Anschaulich dargestellt an einem Beispiel:

	Person	Studium	ErfahrungExtern	ErfahrungIntern	Alter	Gehalt
1	1	6	1	4	24	46450
2	2	18	30	15	55	85150
3	3	11	7	7	31	55900
4	4	11	15	8	36	63650
5	5	10	1	16	33	59050
6	6	6	25	6	38	68750
7	7	10	20	20	50	79000
8	8	7	0	1	23	43050

Tabelle 3.1: Einfacher Datensatz mit Berufserfahrung und Gehalt

Tabelle 3.1 zeigt einen einfachen Datensatz, in dem die Mitarbeiter einer Firma und die zugehörigen Gehälter festgehalten sind. „Studium“ beschreibt die Anzahl der Halbjahre im Studium. Analog dazu „ErfahrungExtern“ und „ErfahrungIntern“ die Berufserfahrung in Halbjahren außerhalb und innerhalb der Firma. Zusätzlich ist das Alter der Personen gegeben. Führt man eine lineare Regression (Listing 3.1) für den Datensatz durch (mit Studium, ErfahrungExtern, ErfahrungIntern, Alter als unabhängige und Gehalt als abhängige Variablen), ergibt sich das Ergebnis in Tabelle 3.2.

```
1 #Daten einlesen
2 data <- read.csv2("Beispiel_Berufserfahrung_Datensatz1.csv")
3
4 #Regression mit allen Faktoren
5 regression1 <- lm(Gehalt ~ Studium + ErfahrungExtern + ErfahrungIntern + Alter,
      data=data)
6 summary(regression1)
```

Listing 3.1: Regression mit allen Faktoren

	Estimate	Std. Error	t value	Pr(>||)
(Intercept)	40000	1.415e-11	2.828e+15	<2e-16 ***
Studium	300	5.167e-13	5.807e+14	<2e-16 ***
ErfahrungExtern	850	4.821e-13	1.763e+15	<2e-16 ***
ErfahrungIntern	950	6.308e-13	1.506e+15	<2e-16 ***
Alter	2.010e-13	8.103e-13	2.480e-01	0.82
Adjusted R-squared				1

Tabelle 3.2: Output der Regression mit allen Variablen

Ohne weiter auf die genauen Beizeichnungen einzugehen, gibt die Sternnotation von R an, dass die unabhängigen Variablen Studium, ErfahrungIntern und ErfahrungExtern signifikant sind (drei *), das Alter hingegen nicht (kein *). Die Regression hat ein adjustiertes Bestimmtheitsmaß (R^2; engl. adjusted R-squared) von 1. Das Bedeutet, dass das Gehalt vollständig durch die gegebenen Variablen erklärt werden kann (dies wird in der Realität jedoch nie erreicht).

```
1 #Regression mit signifikanten Faktoren
2 regression2 <- lm(Gehalt ~ Studium + ErfahrungExtern + ErfahrungIntern,
      data=data)
3 summary(regression2)
```

Listing 3.2: Regression ohne Alter

	Estimate	**Std. Error**	**t value**	**Pr(>||)**
(Intercept)	40000	2.393e-12	1.671e+16	<2e-16 ***
Studium	300	2.948e-13	1.018e+15	<2e-16 ***
ErfahrungExtern	850	8.948e-14	9.500e+15	<2e-16 ***
ErfahrungIntern	950	1.642e-13	5.787e+15.75	<2e-16 ***
Adjusted R-squared				1

Tabelle 3.3: Output der Regression ohne Alters-Variable

Führt man die Regression nun ohne das Alter durch (Listing 3.2 und Tabelle 3.3) bleibt R^2 gleich. Der Datensatz wurde also bereits um eine Variable reduziert, ohne das Ergebnis der Regression zu verschlechtern.
Betrachtet man die Faktoren ErfahrungExtern und ErfahrungIntern, so fällt auf, dass sie einen ähnlichen Einfluss auf das Gehalt erzielen (850 und 950).

```
1 #Transformation
2 data[,"ErfahrungGesamt"] <- data[,3] + data[,4]
3
4 #Regression
5 regression3 <- lm(Gehalt ~ Studium + ErfahrungGesamt, data=data)
6 summary(regression3)
```

Listing 3.3: Regression mit zusammengefassten Werten

Fasst man beide Variablen zusammen (Listing 3.3), zeigt sich im Ergebnis (Tabelle 3.4), dass R^2 bei 0,9988 liegt. Die Güte der Regression hat sich also nur minimal verschlechtert.

	Estimate	Std. Error	t value	Pr(>||)
(Intercept)	40107.10	528.87	75.83	7.55e-09 ***
ErfahrungGesamt	876.69	15.86	55.26	3.67e-08 ***
Studium	327.17	64.27	5.09	0.0038 **
Adjusted R-squared				.9988 1

Tabelle 3.4: Output der Regression mit zusammengefassten Werten

Zusammenfassend lässt sich für dieses Beispiel sagen, dass die Variablen im Datensatz um die Hälfte reduziert wurden, ohne die Aussagekraft deutlich zu verschlechtern. In einem realen Datensatz ist diese Arbeit zwar nicht so trivial und offensichtlich, jedoch gelten die gleichen Prinzipien.
Nach (Swamynathan, 2017, S. 71; veränderte Version) gibt es zur Transformation folgende Möglichkeiten:

- Smoothing (binning, clustering, regression, etc.)
- Aggregation (im Beispiel: das Zusammenfassen der Berufserfahrung)
- Generalization (Ersetzen von primitiven Datenobjekten durch höherstufige Konzepte)
- Normalization (min-max-scaling oder z-score)
- Feature construction aus bereits bestehenden Attributen durch Techniken wie die Hauptkomponentenanalyse (engl. principal components analysis; PCA), Multidimensional scaling (MDS) oder Locally-linear embedding (LLE)
- Compression (zum Beispiel wavelets, PCA, clustering etc.)
- andere Datenreduzierungstechniken bei denen das Datenvolumen sinkt, ohne die Integrität der Originaldaten zu verletzten

3.1.4 Data Mining

Ist der Datensatz präpariert, so findet das eigentliche Data Mining statt. Dabei muss sich der Anwender für eine oder auch mehrere Methoden für das Mining entscheiden, um die anfänglichen Ziele zu erreichen und die Fragestellungen zu beantworten. Zur Auswahl stehen beispielsweise(Fayyad et al., 1996, S. 42; Swamynathan, 2017, S. 71):

- zusammenfassende und beschreibende Methoden: Mittelwert (arithmetisches Mittel), Median, Modus, Standardabweichung, Klassen-und Konzeptbeschreibungen, grafische Plots,
- Vorhersagende Modelle (engl. predictive models): Klassifikationen und Regressionen und
- Cluster-Analysen.

Eine genauere Beschreibung der Methoden (und der zugehörigen Algorithmen) im Kontext des Machine Learning befindet sich in Kapitel 4. Je nach Beschaffenheit der zugrundeliegenden Daten und der gewählten Methode, muss ein passender Algorithmus gewählt und dieser korrekt parametrisiert werden. Zum Data Mining gehört auch, Hypothesen zu formulieren und das Ergebnis im Auge zu behalten: Ist der Endnutzer der Analyse an einem vorhersagenden Model interessiert (zum Beispiel für Wartungsarbeiten) oder an einem Jetzt-bezogenen (zum Beispiel für eine strategische Ausrichtung nach den aktuellen Kundensegmenten)?
Anschließend erfolgt das (automatische) Mining der Daten. Je besser die vorhergehenden Schritte durchgeführt wurden, desto potenter ist das Ergebnis.(Fayyad et al., 1996, S. 42) Aus diesem Grund ist es jederzeit möglich, zu einem vorangegangen Prozessschritt zurückzukehren, um neu erlangte Einsichten einfließen zu lassen (siehe zurückspringender gestrichelter, grauer Pfeil in Abbildung 3.1).

3.1.5 Interpretation/Evaluation

Zuletzt werden die gefundenen Muster und trainierten Modelle interpretiert. Ein Muster macht Aussagen über jeden Datenpunkt im betrachteten Raum. Ein Beispiel bei einem einfachen linearen Model:

$$y = m \times x + t$$

Zu obigem Fall:

$$Gehalt = Studium \times 327,17 + ErfahrungGesamt \times 876,69 + 40107,10$$

Ein Muster (engl. pattern) beschreibt dagegen nur eine kleine „lokale Struktur“, die „nur über einen begrenzen Bereich“ Aussagen macht.(Swamynathan, 2017, S. 71; eigene Übersetzung) Im

Fall des linearen Model, wäre es eine bestimmte Gleichung, zum Beispiel

$$y = 2 \times x + 5$$

oder

$$6 \times 327,17 + 5 \times 876,69 + 40107,10 = 46453,57$$

(Kraker and Dennerlein, 2013). „Fayyad et al. benutzt patterns und models synonym“.(Kraker and Dennerlein, 2013)
Das Interpretieren der Ergebnisse beinhaltet ebenfalls das Zusammenfassen der Erkenntnisse und gegebenenfalls das Visualisieren.(Swamynathan, 2017, S. 71) Als Evaluieren wird das Eingliedern der Resultate in andere Systeme (zur Weiterverarbeitung oder Verbreitung), das Prüfen auf (und Lösen von) Konflikten mit anderen Untersuchungen und nicht zuletzt das Dokumentieren der Befunde bezeichnet.(Fayyad et al., 1996, S. 42)
An dieser Stelle sei erneut angemerkt, dass das erste Ergebnis des KDD-Prozesses nicht das Endergebnis sein muss. Es kann durchaus viele Iterationen geben, die auch „loops between any two steps“ beinhalten können.(Fayyad et al., 1996, S. 42)

3.2 CRoss Industrial Standard Process for Data Mining (CRISP – DM)

Bei CRoss Industrial Standard Process for Data Mining handelt es sich - wie bei KDD - um ein Referenzmodell für Data Mining. Das Modell wurde von einem 1996 gegründeten Konsortium aus „Daimler-Benz (now DaimlerChrysler), Integral Solutions Ltd. (ISL) [jetzt SPSS], NCR, and OHRA“(Shearer, 2000, S. 13) erarbeitet. Die Version 1.0 wurde 2000 vorgestellt.(Shearer, 2000, S. 13) In Umfragen (1999, 2002, 2004, 2007) wird das Modell als führend in Bereich von „data mining/predictive analytics projects“(Swamynathan, 2017, S. 72) bezeichnet. Das Modell ist „nicht-properitär, dokumentiert und frei verfügbar“(Shearer, 2000, S. 13; eigene Übersetzung). Es ist ebenfalls in vielen Bereichen nutzbar, da es weder Industriesektor-, Werkzeugs- noch Anwendungsspezifisch ist. Grundsätzlich bekräftigt das Modell best practices und soll zu besseren und schnelleren Ergebnissen führen.(Shearer, 2000, S. 13; eigene Übersetzung)

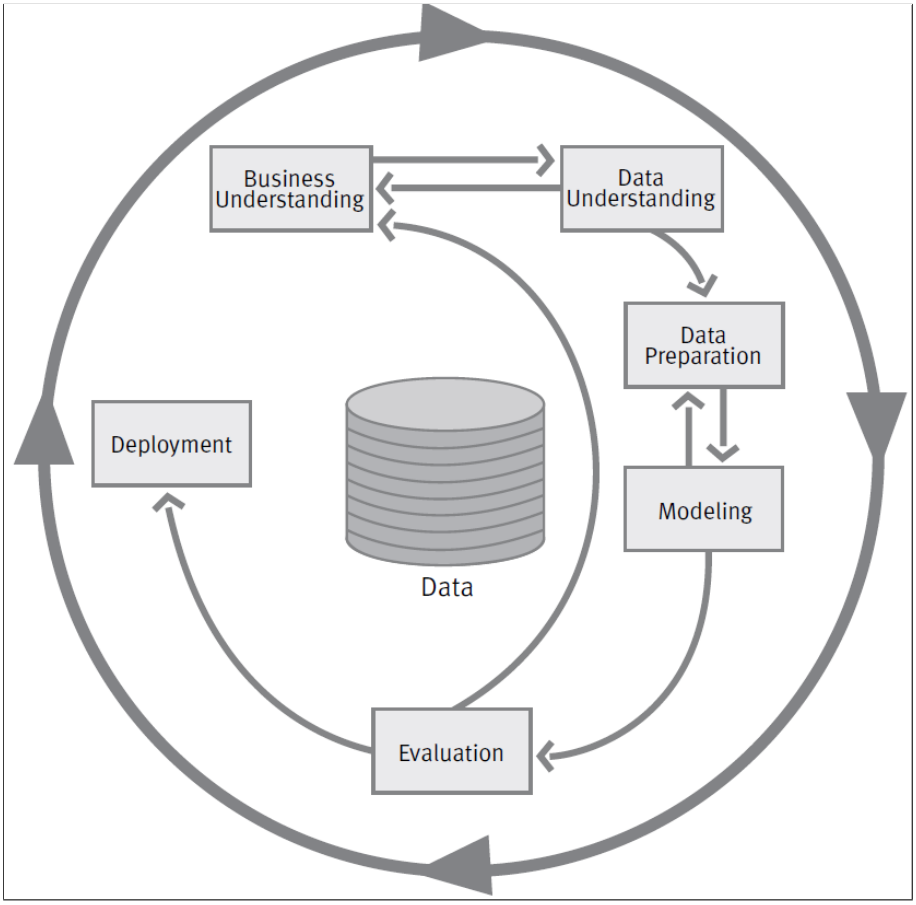

Abbildung 3.2: Phasen des CRISP-DM Referenzmodells nach (Chapman et al., 2000, S. 10)

Wie in Abbildung 3.2 zu sehen ist, umfasst das Referenzmodell sechs Phasen. Genau wie beim KDD-Prozessmodell handelt es sich nicht um ein lineares Modell, sondern um eines, das Rückschritte und Iterationen erlaubt. Im Nachfolgenden wird zuerst immer ein Prozessschritt kurz vorgestellt und darunter detaillierter betrachtet. Als Referenz dient unter anderem Abbildung 3.3, die den Output der einzelnen Schritte zeigt.

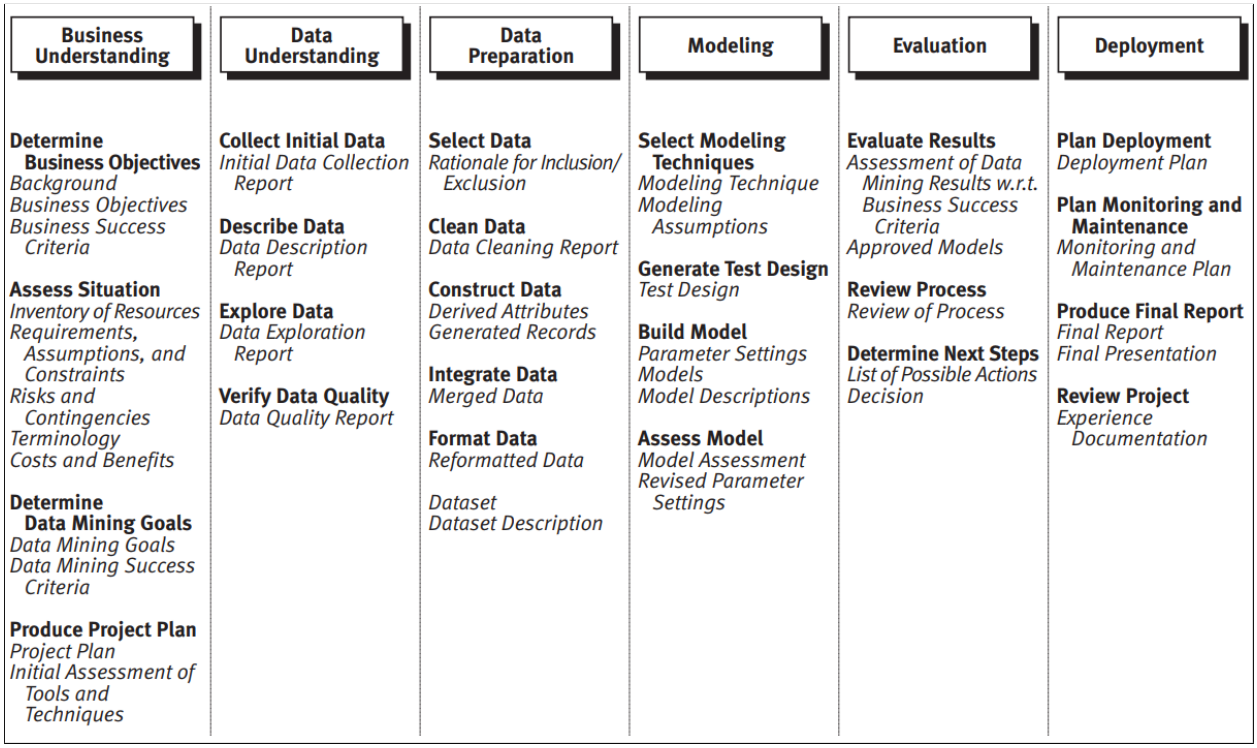

Abbildung 3.3: Generische Aufgaben (**fett**) und Output (*kursiv*) des CRISP-DM Referenzmodells (Chapman et al., 2000, S. 12)

3.2.1 Business Understanding

Die erste und vielleicht wichtigste Phase(Shearer, 2000, S. 14) des CRISP-DM Prozesses ist das „Business Understanding“, oder auch „Research Understanding“(Larose, 2014, Punkt 1.4.1.1). Die Aufgabe dieser Phase ist, die „Ziele und Erwartungen“(Swamynathan, 2017, S. 73) des Projektes zu verstehen, dieses „Wissen in eine Machine Learning Problem Definition zu übersetzen“ und schließlich einen „Vorläufigen Plan“(Shearer, 2000, S. 14) aufzustellen:

Determine the Business Objectives

Dieser Teilschritt soll hauptsächlich die Frage beantworten, warum die Analyse durchgeführt wird. Dies hat direkten Einfluss auf die Ziele des Projekts und soll verhindern, dass „viel Aufwand für das Finden von richtigen Antworten auf falsche Fragen“(Chapman et al., 2000, S. 14) verschwendet wird.

Assess the Situation

Um das Ziel der Analyse so genau wie möglich zu treffen, muss genau nachgeforscht werden, welche Ressourcen verfügbar sind, welchen Zwängen und Grenzen die Analyse unterlegen ist und unter welchen Annahmen sie stattfindet.(Chapman et al., 2000, S. 14) Vereinfacht lässt sich sagen, dass hier die Fragen aus dem vorhergehenden Schritt detaillierter betrachtet werden.

Determine the Data Mining Goals

Die gefundenen Ziele sind meist in Geschäftssprache formuliert. Für die Analyse müssen die Ziele jedoch im Terminus technicus des Data Mining formuliert sein. Ein Beispiel dazu ist die Übersetzung von „Increase catalog sales to existing customers“ in „Predict how many widgets a customer will buy, given their purchases over the past three years, demographic information (age, salary, city, etc.), and the price of the item.“(Chapman et al., 2000, S. 16)

Produce a Project Plan

Der Projektplan (engl. Project Plan) liefert einen konkreten Plan, wie die gesetzten Ziele zu erreichen sind. Nach (Shearer, 2000, S. 15) beinhaltet er:

- Die Schritte, die nacheinander durchzuführen sind.
- Eine Timeline für die Durchführung.
- Eine Auflistung potentieller Risiken im Projektverlauf.
- Eine Aufstellung der zu nutzenden Werkzeuge und Techniken (Chapman et al., 2000, S. 16)

3.2.2 Data Understanding

In der vorhergegangenen Phase wurde festgelegt, welche Daten zum Erreichen der Ziele benötigt werden. Nun werden diese Daten gesammelt und untersucht. Im Fokus der Untersuchung liegt dabei(Swamynathan, 2017, S. 73)

- Datenlücken zu finden,
- die Relevanz der erfassten Daten (hinsichtlich der Ziele) zu klären,
- die allgemeine Datenqualität festzustellen und
- „erste Einblicke in Daten“ zu erhalten, um „geeignete Hypothesen“(Swamynathan, 2017, S. 73; eigene Übersetzung) zu formulieren.

Im Zuge dessen können auch bereits „subsets“ isoliert werden, die „actionable patters“(Larose, 2014, Punkt 1.4.1.2.d) (etwa: verfolgbare Muster) enthalten könnten. Mit jedem Fortschritt in dieser Phase ist es eventuell nötig, das Ergebnis der Business Understanding-Phase zu adjustieren. Durch dieses Vorgehen, entsteht ein iterativer Prozess.(Swamynathan, 2017, S. 73) (Larose, 2014, Punkt 1.4.1.2.b) empfiehlt den Einsatz von explorativer Datenanalyse (siehe Abschnitt 1.3).

Collect the Initial Data

Die Hauptaufgabe dieses ersten Schrittes ist, die benötigten Daten zu beziehen. Dabei kann entweder der direkte Zugriff auf die Daten gemeint sein oder nur das Erhalten der Zugangsinformationen. Eventuell werden die Datensätze gleich in Systeme oder Werkzeuge geladen, die zur späteren Weiterverarbeitung genutzt werden. Die Integration inhomogener Daten (unterschiedliche Strukturen/Formate etc.) kann bereits hier erfolgen oder in der Data Preperation (siehe Punkt 3.2.3). (Chapman et al., 2000, S. 18)

Sollten in diesem Schritt Probleme auftauchen, sollten sie - wenn möglich mit Lösung - gut dokumentiert werden, um den Projektverlauf reproduzierbar zu gestalten. Ein Beispiel können lange Antwortzeiten einiger Datenquellen sein.(Shearer, 2000, S. 15)

Describe the Data

Im Schritt „Describe the Data“ werden die beschafften Daten dann oberflächlich beschrieben.(Chapman et al., 2000, S. 18) Dabei wird unter anderem auf

- das Format der Daten,
- die Größe des Datensatzes,
- die Anzahl der Beobachtungen und Einträge in den Daten und
- die Beschaffenheit der Einträge

geachtet. Dabei soll einerseits die Frage geklärt werden, ob die vorhandenen Daten alle relevanten Daten für die Ziele des Data Mining enthalten. Andererseits wird das Verständnis der Daten geschärft.(Shearer, 2000, S. 15)

Anhand des nachfolgenden Beispiels (Datensatz von (Hertle, 2016, Case Lasagne Test.xlsx)) werden die Schritte „Collect the Initial Data“ und „Describe the Data“ kurz visualisiert. Zuerst werden die Daten eingelesen (Listing 3.4) und anschließend oberflächlich betrachtet.

```
1 data <- read.csv2("Case␣Lasagne␣Test.csv")
2 head(data)
```

Listing 3.4: Einlesen aller Daten und Betrachten des „Kopfes“

Aus Tabelle 3.5 kann abgelesen werden, dass es sich um einen Datensatz mit 12 Variablen handelt.

Person	Alter	Einkommen	Angestellt	Wert.Auto	Umsatz.Kreditkarte	Geschlecht
1	48	91700	nein	2190	3510	m
2	33	40740	nein	2110	740	w
3	51	45080	ja	5140	910	m
4	56	26600	nein	700	1620	w
5	28	113960	ja	26620	600	m
6	51	102200	ja	24520	950	w

Gewicht	alleinstehend	Wohnung	Supermarkt.besuche.pro.Monat	Lasagne.probiert
65	nein	Haus	7	nein
75	nein	Wohnung	4	ja
70	nein	Wohnung	1	nein
91	nein	Haus	3	nein
81	nein	Appartement	3	ja
65	nein	Wohnung	2	nein

Tabelle 3.5: Aufruf des head()-Befehls zum Betrachten der Daten

Ebenfalls entnommen werden kann, dass es sich um demographische Angaben über Personen handelt. Zusätzlich wurde zu jeder Person erfasst ob sie Lasagne probiert hat. In Abbildung 3.4 wird der Datensatz schließlich in RStudio betrachtet.

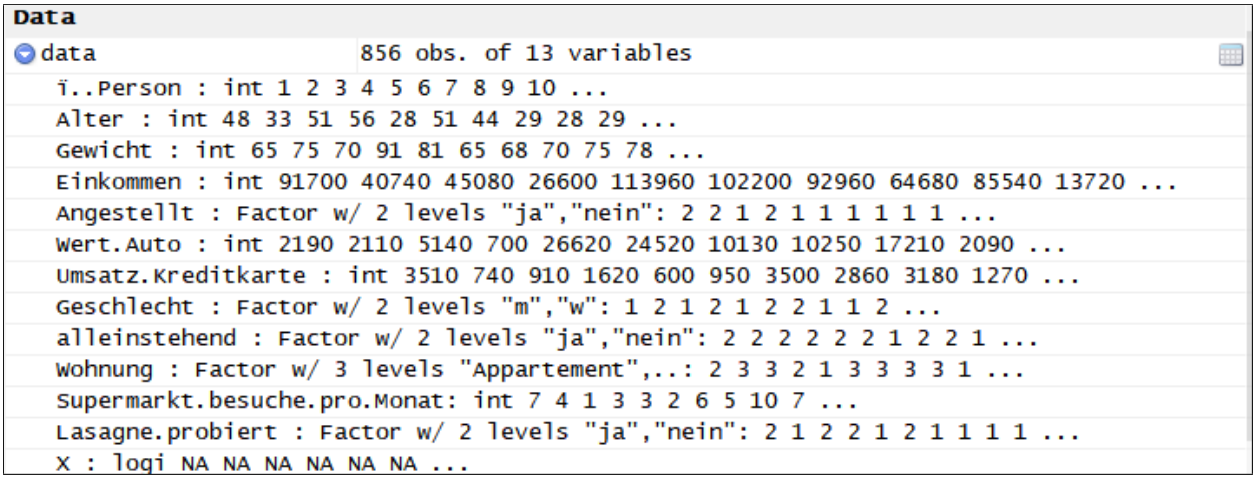

Abbildung 3.4: Betrachten der Datentypen des Datensatzes in RStudio

Zu sehen ist hier, dass es sich um einen Datensatz mit 13 Variablen und 856 Beobachtungen handelt (hier scheint ein falscher Zeichensatz vorzuliegen, da eine zusätzliche leere Spalte „X“

angezeigt wird). Zusätzlich können die vorgeschlagenen Datentypen von R betrachtet werden. Bei den meisten Spalten handelt es sich um Ganzzahlen (int) oder Faktoren (manchmal auch als Enums bezeichnet) wie m/w für männlich und weiblich.

Explore the Data

Ist die grobe Sichtung der Daten abgeschlossen, wird enger an der Fragestellung des Data Mining gearbeitet. Dazu werden „Abfrage-, Visualisierungs- und Reporting[-Techniken]“(Shearer, 2000, S. 16; eigene Übersetzung) eingesetzt. Um der Antwort auf die ursprüngliche Fragestellung näher zu kommen oder die Fragestellung zu verfeinern, werden beispielsweise folgende Eigenschaften betrachtet(Chapman et al., 2000, S. 18; eigene Übersetzung):

- Die Verteilung der Schlüsselattribute (zum Beispiel der Zielvariablen bei einer Vorhersage).
- Die Beziehungen zwischen Wertepaaren oder kleinen Attributgruppen.
- Die Ergebnisse einfacher Aggregationen.
- Die Beschaffenheit von aussagekräftigen Teilgruppen von Werten.
- Die Ergebnisse einfacher statistischer Analysen.

Verify Data Quality

Der letzte Schritt der zweiten Phase evaluiert die Qualität der Daten. (Chapman et al., 2000, S.19) nutzen die Zielfragen:

- „Is the data complete (does it cover all the cases required)?“
- „Is it correct, or does it contain errors and, if there are errors, how common are they?“
- „Are there missing values in the data?“
- „If so, how are they represented, where do they occur, and how common are they?“

(Shearer, 2000, S. 16) empfiehlt zusätzlich noch, zu prüfen, ob die Werte plausibel sind, wie die Schreibweisen sind, ob Attribute mit unterschiedlichen Werten aber gleicher Bedeutung vorhanden sind und schließlich ob es einen „conflict with common sense“, wie „teenagers with high income“(Shearer, 2000, S. 16) gibt.

3.2.3 Data Preparation

In der aufwendigsten Phase des ganzen Prozesses (siehe Abbildung 3.5) wird das „final data set“(Larose, 2014, Punkt 1.4.1.3.a; Shearer, 2000, S. 16) erzeugt. Dies geschieht durch(Swamynathan, 2017, S. 73)

- generelle Transformationen,
- Füllen der Datenlücken, die in vorangegangenen Schritten aufgedeckt wurden,
- Befassen mit fehlenden Werten,
- Herausarbeiten, welche Features des Datensatzes die größte Relevanz haben und welche neuen Features sinnvoll wären.

Wie bereits erwähnt, handelt es sich nicht nur um die Phase, die den meisten Aufwand erfordert, sondern auch um die, von der die Genauigkeit des Endresultates zu großen Stücken abhängt.(Swamynathan, 2017, S. 73)

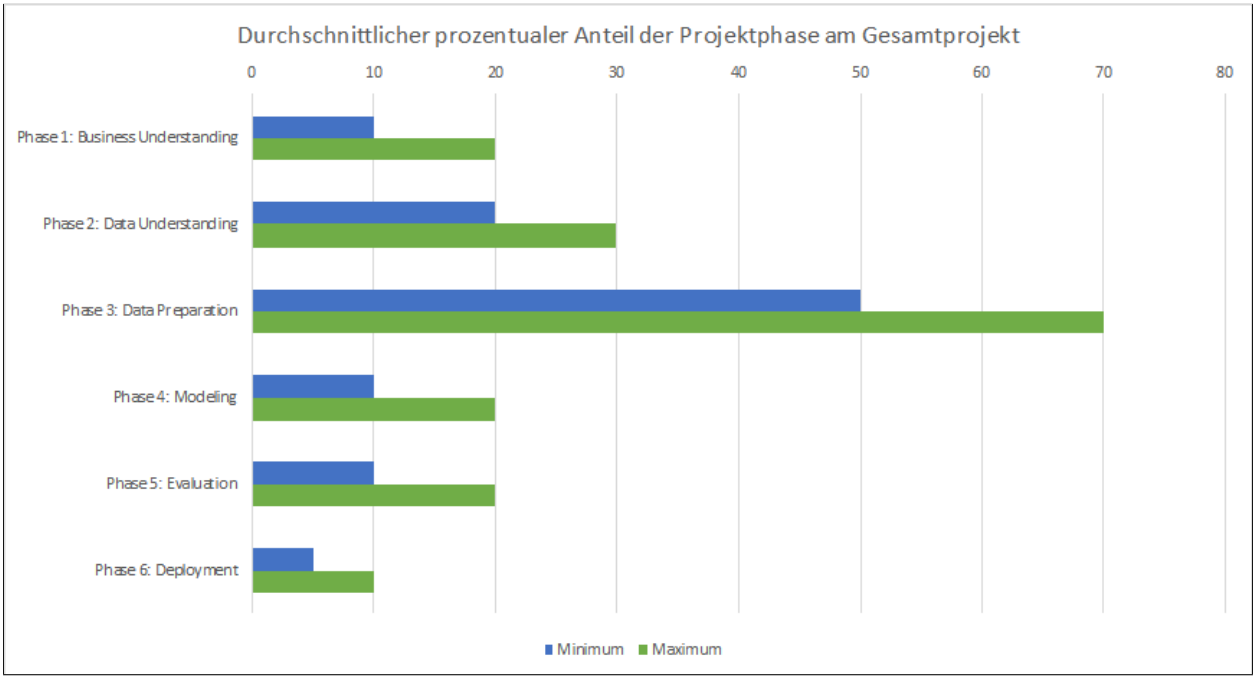

Abbildung 3.5: Durchschnittlicher prozentualer Anteil der CRISP-DM-Projektphase am Gesamtprojekt nach (Shearer, 2000, S. 15; eigene Darstellung)

Select Data

Genauer wird dabei im ersten Schritt ausgewählt, welche Daten Teil der Analyse bleiben und welche exkludiert werden. Kriterien sind dabei, die Relevanz hinsichtlich der Ziele, die Qualität der Daten und technische Grenzen(Shearer, 2000, S. 16) (wie „data volume or data types“(Chapman et al., 2000, S. 21)). Zusätzlich kann überlegt werden, ob einige Attribute wichtiger sind als andere. So kann beispielsweise bei einer landesweiten Kundenanalyse die Postleitzahl der Kunden ausreichen und Straße und Hausnummer vernachlässigt werden.(Shearer, 2000, S. 16) (Chapman et al., 2000, S.21) merken an, dass diese Phase sowohl „attributes (columns)“, als auch „records (rows)“ umfasst. Wie bereits in den vorhergehenden Schritten muss auch hier erklärt werden, warum Entscheidungen getroffen wurden und eine Dokumentation angefertigt werden.(Shearer, 2000, S. 16)

Clean Data

Im Schritt „Verify Data Quality“ wurde herausgearbeitet, wie die Qualität der Daten ist und wie mängelbehaftet sie sind. Jetzt werden Maßnahmen dagegen ergriffen. Neben trivialen

Vorgehen wie „Auswahl von reinen Untermengen“ oder „Einfügen von passenden Standardwerten“ können „anspruchsvollere Techniken wie das Schätzen von fehlenden Werten“(Chapman et al., 2000, S.21; eigene Übersetzung) zum Zuge kommen.

Construct Data

Die gereinigten Daten sind noch nicht fertig für die Modeling-Phase. Manchmal ist es notwendig, einem Datensatz neue Zeilen hinzuzufügen. Betrachtet man wieder eine Kundenanalyse, so ist es vielleicht nötig, für einen Kunden, der in einem Quartal keinen Einkauf getätigt hat, einen leeren Einkauf (null Euro) anzulegen, falls der eingesetzte Algorithmus dies erfordert.(Chapman et al., 2000, S. 22) Er kann auch verlangen, dass abgeleitete statt der Ursprungswerte benötigt werden. Dabei gibt es nach (Shearer, 2000, S. 16) zwei Fälle:

1. Wenn zu einem Kunden ein Bewegungsprofil vorhanden ist (in welchem Geschäft er einkaufen war), so ist möglicherweise sinnvoll, nicht das gesamte Profil zu betrachten, sondern lediglich die Fläche zu betrachten, in der er sich bewegt hat.

2. Ebenfalls zielführend kann eine „single-attribute transformation“ sein. Dabei wird beispielsweise das genaue alter der Kunden in Altersspannen umgewandelt oder sprechende Werte wie ’(„definitely yes,“ „yes,“ „don’t know,“ „no“)’ in numerische Werte übersetzt.

(Shearer, 2000, S. 16; eigene Übersetzung) merkt aber auch an, dass es nicht immer sinnvoll ist dies zu tun, auf jeden Fall „nicht nur um die Anzahl der Inputattribute zu reduzieren.“

Integrate Data

Der vorletzte Schritt der Data Pereperation ist das Zusammenführen von mehren Quellen oder Tabellen mit dem gleichen Thema. Dadurch können „neue Beobachtungen oder Werte“(Chapman et al., 2000, S. 22) gewonnen werden. In Tabelle 3.6 werden die zwei Hauptaufgaben(Shearer, 2000, S. 17) genauer erläutert.

Aufgabe	**Erläuterung**	**Beispiel**
Join	Mehrere Tabellen zum gleichen Thema werden zusammengeführt.	Die drei Ausgangstabellen • „information about each store's general characteristics (e.g., floor space, type of mall)“, • „summarized sales data (e.g., profit, percent change in sales from previous year)“ und • „information about the demographics of the surrounding area“ werden in • „a new table with one record for each store, combining fields from the source tables“(Shearer, 2000, S. 16) zusammengeführt.
Aggregation	Errechnen neuer Werte aus Informationen verschiedener Tabellen.	Das Überführen von einer • „table of customer purchases, where there is one record for each purchase“ in eine • „new table where there is one record for each customer“ mit den Feldern • „number of purchases, the average purchase amount, the percent of orders charged to credit cards, the percent of items under promotion, etc“.(Shearer, 2000, S. 17)

Tabelle 3.6: Die zwei Hauptaufgaben des Schrittes Integrate Data

Format Data

Die Datenformatierung umfasst „hauptsächlich syntaktische Abänderungen“ und „verändert nicht die Bedeutung“(Chapman et al., 2000, S. 22; eigene Übersetzung) der Daten. Das kann kann zum Beispiel das „Entfernen von unerlaubten Zeichen in Zeichenketten“(Shearer, 2000, S. 17; eigene Übersetzung) sein.
Mit diesem Schritt ist die Data Preperation abgeschlossen und es kann mit dem Modeling begonnen werden.

3.2.4 Modeling

Die Phase des Modeling umfasst die Auswahl einer oder mehrerer Data Mining-Algorithmen, die Optimierung ihrer Parameter und Settings und der Evaluierung des erzeugten Models.(Swamynathan, 2017, S. 73; Larose, 2014, Punkt 1.4.1.4) Eventuell muss der Datensatz noch angepasst werden, sodass die Data Preperation-Phase noch einmal durchlaufen werden muss.(Larose, 2014, Punkt 1.4.1.4)

Select the Modeling Technique

Wie der Name des ersten Schrittes bereits andeutet, wird eine Modellierungstechnik ausgewählt. Werden mehrere Techniken ausgewählt, so wird diese Phase mehrfach (parallel) durchlaufen. Wichtig ist hier, dass getroffene Annahmen (wie „alle Attribute sind stetig Gleichverteilt“ oder „fehlende Werte sind nicht zugelassen“(Chapman et al., 2000, S. 24)) dokumentiert werden.(Shearer, 2000, S. 17)

Generate Test Design

Vor der eigentlichen Modellierung wird festgelegt, wie die „Qualität und Validität“(Chapman et al., 2000, S. 24) festgestellt werden soll. Für „supervised data mining“ werden dabei meist „error rates“(Chapman et al., 2000, S. 24) herangezogen. Dazu wird das Modell mit einem Datensatz (train set) trainiert und mit einem anderen (test set) getestet.(Shearer, 2000)

Build the Model

Der kürzeste Schritt dieser Phase ist „Build the Model“. Hier wird das Model mit Hilfe eines - möglicherweise vorher bereits gewählten - Werkzeugs erzeugt.(Chapman et al., 2000, S. 24; Shearer, 2000, S. 17) Wurden zwei Schritte zuvor mehrere Modellierungstechniken ausgewählt, so liegen an dieser Stelle mehrere Models vor.

Assess the Model

Ist das Modellieren abgeschlossen, werden die Ergebnisse auf Basis

- des Verständnisses aus der ersten Phase (Business Understanding),
- der Data Mining-Ziele und
- des Test Designs aus dieser Phase

interpretiert. Der Analyst hat die Aufgabe, den Grad des Erfolgs des Data Mining zu bestimmen. Dazu kann er Experten heranziehen, um das Ergebnis beispielsweise auf Geschäftsebene zu diskutieren. Zusätzlich wird eine Rangliste aller Modelle aufgestellt, die den Erfolg hinsichtlich der „Business Ziele“ (aus Phase „Business Understanding“, Schritt „Determine the Business Objectives“) abbildet.(Chapman et al., 2000, S. 25; Shearer, 2000, S. 17)
In diesem Schritt werden die Modelle ein erstes Mal gedeutet. Eine genauere Auswertung und zusätzliche Ergebnisse, Erkenntnisse und Dokumente aus den vorhergehenden Schritten werden in der nachfolgenden Evaluation-Phase bewertet.(Chapman et al., 2000, S.25)

3.2.5 Evaluation

Die Tatsache, dass es sich beim CRISP-DM-Referenzmodell um einen Prozess handelt und nicht um strikt getrennte Einzelschritte, wird besonders in der Evaluations-Phase deutlich. Erstens wird das Ranking aus dem vorherigen Schritt in in einem „Benchmarking“ über die „Models mit einer hohen Genauigkeit“(Swamynathan, 2017, S. 73; eigene Übersetzung) verfeinert. Zweitens werden die Models erneut mit frischen Daten (nicht aus Schritt „Generate Test Design“) verifiziert und gegen die Business-Anforderungen aus Phase 1 geprüft.(Swamynathan, 2017, S. 73) Ziel dieser Phase ist vor allem, dem Projektleiter genug Wissen an die Hand zu

geben, um zu Entscheiden, wie mit den Ergebnissen des ganzen Prozesse weiter verfahren wird.

Evaluate Results

Während sich bisher hauptsächlich um die „Genauigkeit und Allgemeingültigkeit“ der Modelle gekümmert wurde, wird jetzt auch betrachtet, ob es „irgendwelche Businessgründe gibt“, durch die das Model „mangelhaft“(Shearer, 2000, S. 18; eigene Übersetzung) wird. Falls „time und budget“(Chapman et al., 2000, S. 26) es erlauben, können die Ergebnisse bereits in echte Systeme in Testumgebungen implementiert werden. Wie bereits angemerkt, werden in dieser Phase auch andere „findings“ evaluiert, die beispielsweise auf zukünftige Herausforderungen hinweisen.(Shearer, 2000, S. 18) Ist dies geschehen, „fasst der Data Analyst die Bewertungen der Ergebnisse hinsichtlich der geschäftlichen Erfolgskriterien zusammen“ und gibt seine Wertung ab, „ob das Projekt bereits die initialen geschäftlichen Ziele erreicht“(Shearer, 2000, S. 18; eigene Übersetzung).

Review Process

Im Review wird abgesichert, dass kein Faktor unbeachtet geblieben ist und keine Aufgabe vergessen wurde. Ebenfalls wird die Qualität gesichert (zum Beispiel Bugs in Softwarekomponenten gesucht) und rechtliche Überlegungen angestellt („Dürfen wir diese Kundendaten produktiv für diese Analyse benutzen?“).(Shearer, 2000, S. 18; Chapman et al., 2000, S. 27)

Determine Next Steps

Schließlich werden alle Bewertungen bis hierher genutzt, um zu entscheiden, ob eine weitere Prozessiteration durchlaufen wird, oder, ob in die Deployment-Phase übergegangen wird. Laut (Shearer, 2000, S. 18) trifft diese Entscheidung der Projektleiter. (Chapman et al., 2000, S. 17) sind der Meinung, dass das ganze Projektteam entscheiden sollte.

3.2.6 Deployment

Ist die Entscheidung für das Deployment gefallen, wird die letzte Phase initiiert. Zu Beginn des CRISP-DM-Prozesses wurden Ziele festgelegt, die begründen, weshalb das Data Mining durchgeführt werden soll. Eine einfache Implementierung wäre das Erstellen eines Reports, eine Komplexere dagegen, den Data Mining Prozess in eine andere Abteilung zu portieren(Larose, 2014, Punkte 1.4.1.6.b und c) oder „Echtzeit-Personalisierung von Webseiten“(Shearer, 2000, S. 18; eigene Übersetzung) durchzuführen.
Die Implementierung des Models in die produktiven Systeme befriedigt diese Ziele nicht alleine. Auch das Training jener Personen, die das Wissen im Geschäftsprozess anwenden, muss durchgeführt werden. Dies beinhaltet sowohl die Fähigkeit, die Ergebnisse zu interpretieren, als auch zu verstehen, wie sie die Entscheidungsfindung unterstützen können.(Swamynathan, 2017, S. 73)
Da die weiteren Aufgaben oft nicht vom Data Analyst durchgeführt werden (Larose, 2014, Punkt 1.4.1.6.d), muss der Anwender die Pflege eines Machine Learning-Models verstehen und übernehmen (z.B. in welchen Intervallen das Model trainiert wird).(Swamynathan, 2017, S. 74)

Plan Deployment

Der Erste Schritt der Deploymentphase ist die Auswahl und Dokumentation einer geeigneten Strategie für den Einsatz oder das Rollout in die Geschäftsumgebung.(Shearer, 2000, S. 18; Chapman et al., 2000, S. 28).

Plan Monitoring and Maintenance

Zusätzlich zum Rollout muss die Überwachung und Wartung bedacht und geplant werden. Das soll der Fehlbenutzung der Data Mining-Ergebnisse vorbeugen.(Shearer, 2000, S. 18; Chapman et al., 2000, S. 29)

Produce Final Report

Ein nicht unbedingt Data-Mining-spezifischer Schritt, ist das erstellen eines Abschlussberichts. Dieser kann sich je nach Projekttyp unterscheiden. Er kann die Form einer Zusammenfas-

sung haben oder eine ausgedehnte und detaillierte Präsentation sein.(Shearer, 2000, S. 18; Chapman et al., 2000, S. 29) (Larose, 2014, Punkt 1.4.1.1) merkt an, dass es sich auch um Forschungsprojekte handeln kann. In diesem Fall ist der Report wahrscheinlich eine Veröffentlichung der Ergebnisse. Der Abschlussbericht „enthält alle bisher erzeugten Auslieferungsgegenstände und fasst [...] die Ergebnisse zusammen."(Shearer, 2000, S 18; eigene Übersetzung)

Review Project

Den Schlussstrich zieht das Review des Projektes. Hier wird festgehalten, was im Projektverlauf gut und schlecht lief. Zusätzlich soll das Wissen konserviert werden, wie der Prozess optimiert werden könnte.(Shearer, 2000, S. 18; Chapman et al., 2000, S. 29) (Shearer, 2000, S. 18; eigene Übersetzung) empfiehlt „Interviews mit allen wichtigen Projektteilnehmern". In „idealen Projekten" umfasst das Review „alle Reports, die in vorhergehenden Projektphasen [...] verfasst wurden."(Chapman et al., 2000, S. 29; eigene Übersetzung)

3.3 Auswahl

Wie zu sehen ist, handelt es sich sowohl bei KDD, als auch bei CRISP-DM um sehr umfangreiche Methodiken. Beide sind vollwertig und im Falle dieser Arbeit anwendbar. Da laut einer KDnuggets Umfrage, CRISP-DM das mit Abstand häufigst eingesetzte Prozessmodell in „analytics, data mining or data science procjets" ist (Piatetsky, 2014) und es einen User Guide bereit stellt(Chapman et al., 2000, S. 7), orientiert sich der Praxisteil der Arbeit an ihm.

4 Machine Learning

Der nun folgende Abschnitt befasst sich mit Machine Learning. Zu Beginn der Arbeit wurde bereits erwähnt, dass maschinelles Lernen als „Sammlung von Algorithmen und Techniken“ verstanden werden kann, die „genutzt werden, um Computersysteme zu erstellen, die aus Daten lernen, um Vorhersagen zu erstellen“.(Swamynathan, 2017, S. 53; eigene Übersetzung) Um diese Algorithmensammlung genauer zu betrachten, ist es sinnvoll sie nach bestimmten Kategorien zu ordnen.

- (Kubat, 2017) unterscheidet in seinem Werk unter anderem nach verschiedenen Klassifikationen (Baysianisch, Nearest-Neighbor, Linear und Polynomial), künstlichen neuronalen Netzen (engl. artifical neuronal network, kurz: ANN), Entscheidungsbäumen, Unsupervised Learning, genetischen Algorithmen und Reinforcement Learning.
- Einen anderen Ansatz wählt (Swamynathan, 2017). Er gliedert die Algorithmen nach Supervised Learning (mit Regressionen und Klassifikationen), Unsupervised Learning (mit Clusteranalyse, Dimensionsreduzierung und Anomalie-Erkennung) und Reinforcement Learning (Markow-Entscheidungsprozess, Q-Learning, Temporal Difference- und Monte-Carlo Methoden). (Kim, 2017) wählt die gleiche Kategorisierung in die drei Typen.
- (Paluszek and Thomas, 2017) sehen neben Supervised und Unsupervised Learning noch Semisupervised und Online Learning.

Diese unterschiedlichen Gliederungen erklären (Ramasubramanian and Singh, 2017, S. 222) damit, dass entweder nach „Learning types“ (siehe Abbildung 4.1) oder „Subjective grouping“ (siehe Tabelle 4.1) klassifiziert werden kann.

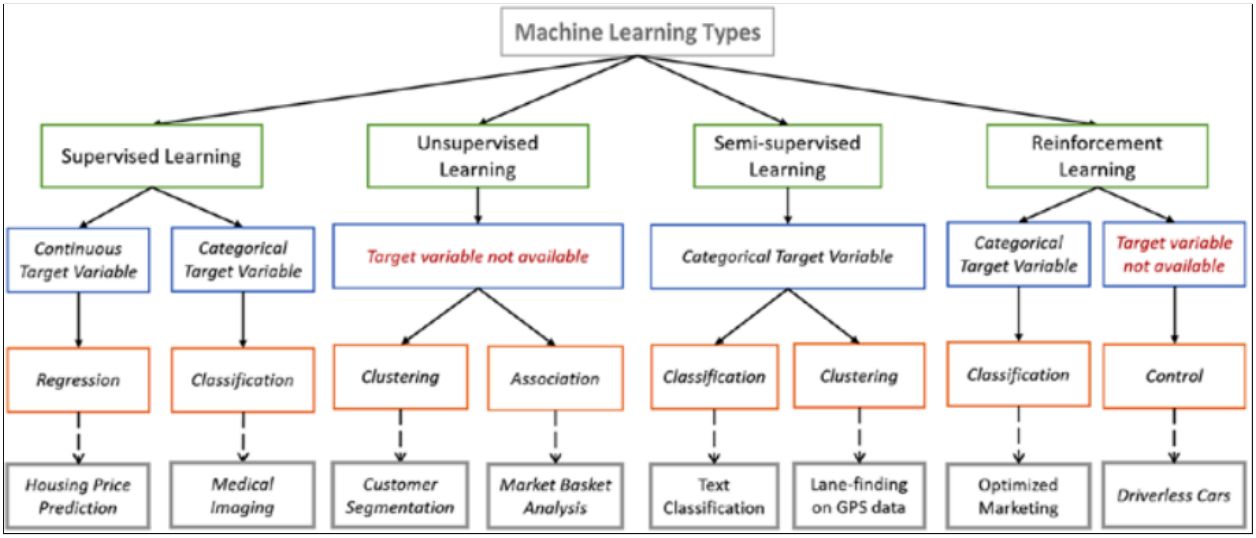

Abbildung 4.1: Machine Learning Types nach (Ramasubramanian and Singh, 2017, S. 222)

Gruppe	Algorithmen
Regression Analysis	Ordinary Least Square Regression (OLSR) Linear Regression Logistic Regression Stepwise Regression Polynomial Regression Locally Estimated Scatterplot Smoothing (LOESS)
Distance-based algorithms	k-nearest Neighbor (kNN) Learning Vector Quantization (LVQ) Self-Organizing Map (SOM)
Regularization algorithms	Ridge Regression Least Absolute Shrinkage and Selector Operator (LASSO) Elastic Net Least-Angle Regression (LARS)
Decision tree algorithms	Classification and Regression Tree (CART) Iterative Dichotomiser 3 (ID3) C4.5 and C5.0 (different versions of a powerful approach) Chi-squared Automatic Interaction Detection (CHAID) Random Forest Conditional Decision Tree
Bayesian algorithms	Naive Bayes Gaussian Naive Bayes Multinomial Naive Bayes Nayesian Belief Network (BNN) Bayesian Network (BN)
Clustering algorithms	k-Means k-Medians Partitioning Around Medoids (PAM) Hierarchical Clustering
Association rule mining	Apriori algorithm Eclat algorithm FP-growth algorithm Context Based Rule Mining
Artificial neural networks	Perception Back-Propagation Hopfield Network Radial Basis Function Network (RBFN)
Deep learning algorithms	Beep Boltzmann Machine (DBM) Deep Belief Networks (DBN) Convolutional Neural Network (CNN) Stacked Auto-Encoders
Dimensionality reduction algorithms	Principam Component Analysis (PCA) Principam Component Regression (PCR) Partial Least Squares Regression (PLSR) Multidimensional Scaling (MDS) Linear Discriminant Analysis (LDA) Mixture Discriminant Analysis (MDA) Quadratic Discriminant Analysis (QDA)
Ensemble learning	Boosting Bagging AdaBoost Stacked Generalization (blending) Gradient Boost Machines (GBM)
Text mining algorithms	Automatic summarization Named entity recognition (NER) Optical character recognition (OCR) Part-of-speech tagging Sentiment analysis Speec recognition Topic Modeling

Tabelle 4.1: Sebjective Grouping nach (Ramasubramanian and Singh, 2017, S. 224-229)

Da der Artikel „How to choose algorithms for Microsoft Azure Machine Learning“ von (Ericson and Rohm, 2017b) nach „Supervised“, „Unsupervised“ und „Reinforcement learning“ zurückgreift, wird nachfolgend diese Gliederung genutzt und um „Semi-supervised Learning“

und „Active Learning“ erweitert.

4.1 Supervised Learning

Die möglicherweise am einfachsten nachvollziehbare Gruppe des Machine Learning ist das Supervised Learning, da es „sehr ähnlich zu dem Prozess ist, in dem Menschen Dinge lernen“(Kim, 2017, S. 13; eigene Übersetzung). Es existiert ein Datensatz, bei dem für jeden Input ein Output vorhanden ist. Ein Beispiel können Patientendaten sein, die als Output eine Variable besitzen, die angibt, ob ein Patient an Krebs erkrankt ist oder nicht.(Ramasubramanian and Singh, 2017, S. 222) Diese „response variable“ (Krebs oder nicht Krebs) wird als „label“(Ramasubramanian and Singh, 2017, S. 222) bezeichnet. Die Aufgabe des Learning Prozesses ist es dann, einen Zusammenhang zwischen dem Input (Patientendaten) und dem Label (Krebs oder nicht Krebs) herzustellen. Dies geschieht mit sogenannten „training sets“(Paluszek and Thomas, 2017, S. 5). Der zweite Schritt ist dann das Überprüfen des entstandenen Models. Dabei wird das Model auf ein zweites gelabeltes „test set“(Paluszek and Thomas, 2017, S. 5) angewandt und das Ergebnis überprüft.
Die Algorithmen des Supervised Learning lassen sich erneut aufteilen:

4.1.1 Classification

Betrachtet man die Labels und stellt fest, dass sie die Datensätze in Kategorien unterteilen (Krebs oder nicht Krebs) oder eine Wahrscheinlichkeit angeben (Person ist zu 89% Max Mustermann bei einer Gesichtserkennung), handelt es sich um eine Klassifikation (engl. Classification).(Swamynathan, 2017, S. 67) (Kauchak, 2016, S. 5) nennt als Beispiele

- biometrische Erkennungen (Gesicht, Iris, Unterschrift etc.),
- Buchstabenerkennung,
- Spamfilter und
- medizinische Diagnosen.

Weiterhin kann zwischen Klassifikationen mit nur zwei Labels („two-class or binomial classification“(Ericson and Rohm, 2017a)) (siehe Abbildung 4.2) oder mehr als zwei Labels („multi-class classification“(Ericson and Rohm, 2017a)) unterschieden werden.

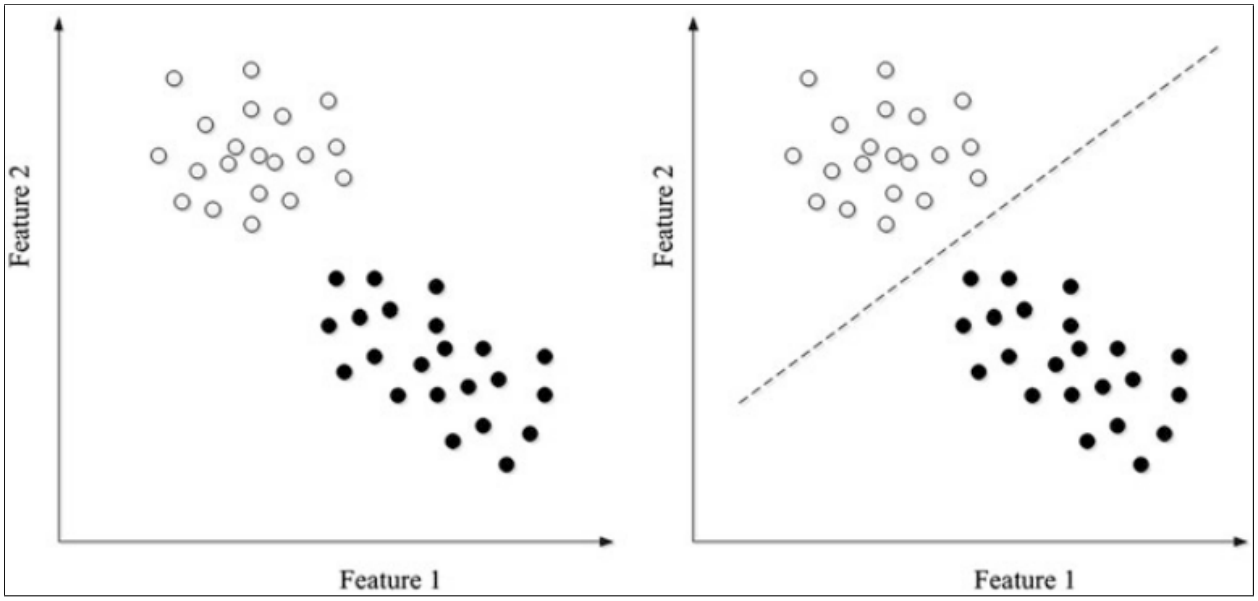

Abbildung 4.2: Beispiel für eine Binomialklassifikation aus (Suthaharan, 2016, S. 8)

4.1.2 Regression

Wenn eine Unterteilung in Kategorien, wie gerade genannt, nicht möglich ist und die Output variable ein fortlaufender Wert (engl. continuous value) ist, werden Regressionen herangezogen. Dabei liegt der „Hauptfokus [...] darin, einen Zusammenhang zwischen einer abhängigen Variablen und einer oder mehreren unabhängigen [...] Variablen herzustellen“(Swamynathan, 2017, S. 60; eigene Übersetzung). Neben dem bekanntesten Beispiel

- einen Kurs an der Börse vorherzusagen (**?**, **?**, S. 5; Kubat, 2017, S. 207; Ericson and Rohm, 2017a), gibt es Anwendungsfälle in
- der Epidemiologie,
- der Auto- und Flugzeugnavigation und
- Analysen im zeitlichen Verlauf (Wetterveränderung im Verlauf der Zeit)(**?**, S. 5).

4.1.3 Anomaly detection

Im bereits mehrfach zitierten Artikel „How to choose algorithms for Microsoft Azure Machine Learning“ von (Ericson and Rohm, 2017a) wird noch eine zusätzliche Kategorie genannt: Anomaly detection. Bei (Swamynathan, 2017, S. 68) ist die Anomaly detection dem Unsupervised Learning zugeordnet. Dies rührt daher, dass es darauf ankommt, ob eine Outputvariable („Label“) vorhanden ist, oder nicht. Tatsächlich ist es so, dass es sowohl Szenarien für Supervised und Unsupervised Anomaly detection gibt, als auch für das später noch beschriebene Semi-supervised Learning.(Chandola et al., 2009, S. 15:10) Unabhängig davon beschreibt Anomaly detection das Finden von Mustern in Daten, die vom erwarteten Verhalten abweichen. Diese Pattern werden meist als Anomalie (engl. anomaly) oder Ausreißer (engl. outliner) bezeichnet.(Chandola et al., 2009, S. 15:1) Die Anomaly detection kann für folgende Szenarien genutzt werden(Chandola et al., 2009, S. 15:2):

- Kreditkartenbetrugserkennung
- Versicherungsbetrugserkennung
- Gesundheitsprüfungen
- Intrusion Detection
- Militärische Überwachungen
- Anwendungen in „der Welt des Internet der Dinge“[S. 68](Swamynathan, 2017)

4.2 Unsupervised Learning

Der ’Glücksfall’, dass der vorhandene Datensatz Label besitzt, ist unter realen Umständen häufig nicht der Fall. Um aus diesen Daten trotzdem Schlüsse zu ziehen, werden Methoden des Unsupervised Learning herangezogen. Hier liegt der Fokus auf dem „Entdecken von aufschlussreichen Eigenschaften der verfügbaren Daten“(Kubat, 2017, S.277; eigene Übersetzung) und der „Untersuchung der Charakteristik der Daten“(Kim, 2017, S. 13; eigene Übersetzung). Dies kann das Ziel haben, komplexe und vielschichtige Daten zu vereinfachen und zu strukturieren(Ericson and Rohm, 2017a) oder die Daten in Gruppen aufzuteilen(Lison, 2012, S. 22). Um diese Gruppen ähnlicher Daten - sogenannte Cluster - geht es im nächsten

Abschnitt.
Supervised Learning findet man nach (Ramasubramanian and Singh, 2017, S. 223)

- bei der Aufteilung von Kundendaten in Segmente,
- in Analysen von sozialen Netzwerken,
- in der Klimatologie,
- bei der Bildkompression und
- in der Bioinformatik.(**?**, S. 6)

4.2.1 Clustering

Beim angesprochenen Clustering handelt es sich um das „Identifizieren von distinkten Gruppen [...] basierend auf irgendeiner Art der Ähnlichkeit innerhalb des vorliegenden Datensatzes“(Swamynathan, 2017, S. 195; eigene Übersetzung). Damit die Objekte der gebildeten Cluster „aussagekräftig und sinnvoll“ sind, sollen „die Objekte innerhalb eines Clusters [...] homogen sein“ und „zu Objekten anderer Cluster“(Ramasubramanian and Singh, 2017, S. 337; eigene Übersetzung) heterogen (siehe Abbildung 4.3). Es kann jedoch auch sein, dass Objekte zu mehreren Clustern gehören. Dies nennt sich „Soft Clustering“ - im Gegensatz zum „Hard Clustering“(Ramasubramanian and Singh, 2017, S. 339). Die Metrik für die Ähnlichkeit ist nicht festgelegt. Möglich sind

- die „Distanz [...] zwischen Beobachtungen“,
- die „Entfernung vom Mittelwert jeder Beobachtung/des Clusters“,
- die „Signifikanz [einer] statistischen Verteilung“ oder
- die „Dichte im Datenraum“(Ramasubramanian and Singh, 2017, S. 338; eigene Übersetzung).

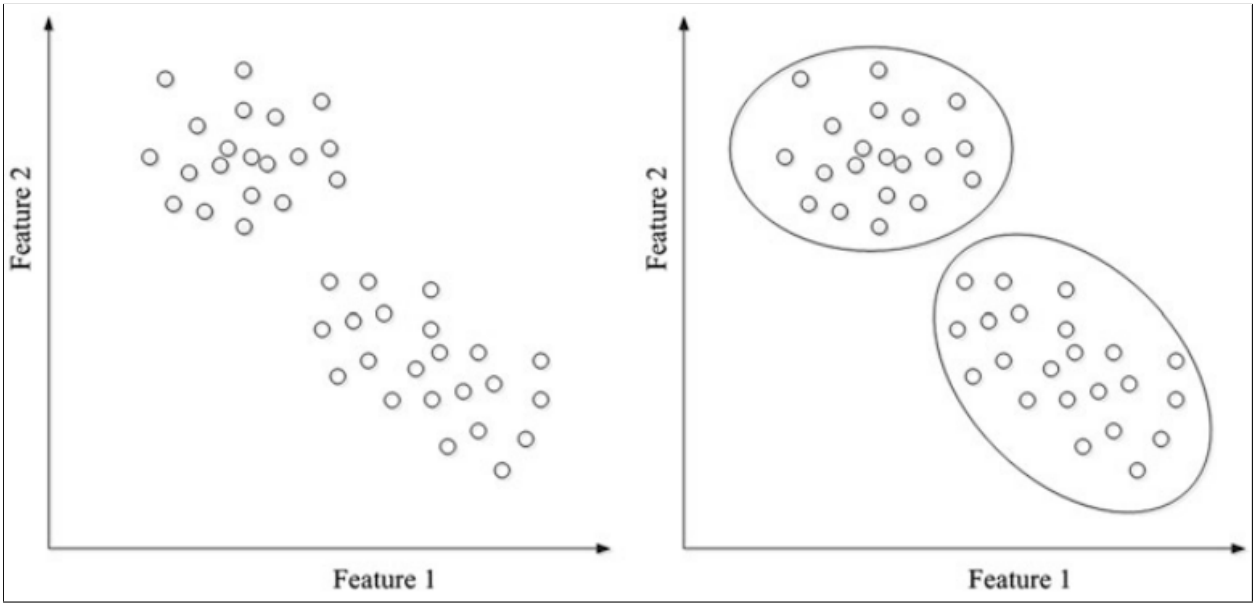

Abbildung 4.3: Beispiel für zwei Cluster (Suthaharan, 2016, S. 9)

4.3 Semi-supervised Learning

Bis jetzt wurden zwei Extremfälle beschrieben: entweder es gibt keine Labels oder alle Daten sind gelabelt. Es existieren jedoch auch Abstufungen dazwischen. Besitzen die meisten Daten ein Label, so ist es eventuell möglich, die Datensätze ohne für das Learning zu entfernen und ein Model aus dem Supervised Learning heranzuziehen. Tritt aber ein Problem auf, bei dem nur sehr wenige Daten gelabelt sind, empfiehlt sich ein Vorgehen aus dem Semi-supervised Learning. Methoden dieser Familie des Machine Learning beruhen auf der Annahme, dass „die Daten wichtige Informationen über die Gruppenzugehörigkeit beinhalten“, „obwohl die Gruppenzugehörigkeit [...] unbekannt ist“(Ramasubramanian and Singh, 2017, S. 223; eigene Übersetzung). Als umfassendes Werk für Semi-supervised Learning ist an dieser Stelle (Chapelle et al., 2006) zu nennen.
Zur Anwendung kommt Semi-supervised Learning seit den 1990er Jahre in „natural language problems“ und „text classification“(Chapelle et al., 2006, S. 4).

4.4 Active Learning

Ein Sonderfall des Semi-Supervised Learning ist das Active Learning. Hier wird davon ausgegangen, dass nur wenige oder keine Labels vorhanden sind, diese jedoch durch einen „Menschen mit umfangreichem Wissen im Themengebiet“(Olsson, 2009, S. i; eigene Übersetzung) hinzugefügt werden können. Alle Daten mit einem Label zu versehen wäre jedoch zu „schwer, zeitaufwendig oder zu teuer“(Settles, 2010, Abstract; eigene Übersetzung). Mit dem Ziel an den Menschen nur so wenig Anfragen (engl. Queries) wie nötig zu stellen(Olsson, 2009, Abstract), werden Algorithmen entworfen, die selbst wählen könne, welche Datensätze gelabet werden sollen.(Settles, 2010, Abstract)
Nach (Settles, 2010, S. 4) wird Active Learning beispielsweise in

- der Spracherkennung,
- der Informationsextraktion oder in
- der Klassifikation oder dem Filtern von Dokumenten oder Mediendateien eingesetzt.

4.5 Reinforcement Learning

„Beim Reinforcement Learning interagiert der Lerner“ - also ein Programm - „mit seiner Umwelt“(Settles, 2010, S. 45; eigene Übersetzung). Er „experimentiert“ dabei, um eine Lösung auf das gestellte Problem zu finden und erhält je nach Ausgang seiner Aktion eine Belohnung oder eine Bestrafung.(Kubat, 2017, S. 331) Es wird also nicht direkt mit einem Output (Label) gearbeitet, sondern lediglich die Qualität des Outputs bewertet. Das Ziel ist, durch ein iteratives Vorgehen (siehe Abbildung 4.4), ein maximales Endergebnis (größer Reward) zu erreichen.(Swamynathan, 2017, S. 69)

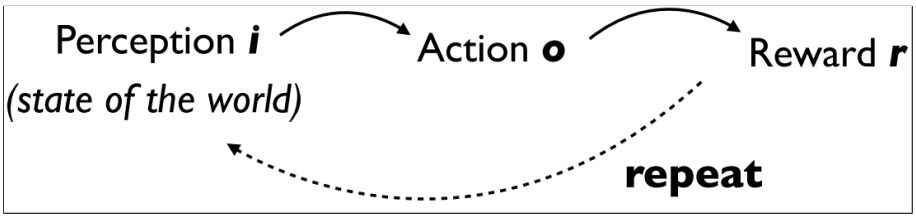

Abbildung 4.4: Iterativer Reinforcement Learning Prozess aus (Lison, 2012, S. 25)

Reinforcement Learning kommt in Szenarien zum Einsatz, in denen noch keine Daten zum Lernen zur Verfügung stehen oder erst nach und nach aktualisiert werden.(Ramasubramanian and Singh, 2017, S. 223)
Ein prominentes Beispiel für ein System mit Reinforcement Learning ist Google DeepMinds AlphaGo. Es handelt sich dabei um ein Go (asiatisches Brettspiel) spielende Computersystem, dass seine Spielstärke durch spielen gegen sich selbst erreichte.(Silver et al., 2017) AlphaGo besiegte 2015 den „legendären Mr Lee Sedol“ (weitgehend als bester Spieler des Jahrzehnts geachtet) vor 200 Millionen Zuschauern.(DeepMind, 2017a)

5 Krypotwährungen

In der Motivation (siehe Abschnitt 1.2) wurde bereits auf Kryptowährungen allgemein und auf Bitcoin im Speziellen eingegangen. Es wurde ebenfalls angemerkt, dass die zwei Währungen mit dem größten Marktvolumen Bitcoin (BTC) und Etherium (ETH) sind. Aus diesem Grund werden die Kurse dieser beiden Währungen analysiert. In diesem Abschnitt wird darauf verzichtet, weiter auf die allgemeine Technik hinter BTC und ETH einzugehen. In Punkt 7.2.2 - vor allem in den Tabellen 7.31 und 7.32 - tauchen bei der Beschreibung der Datensätze jedoch Begriffe auf, die einer genaueren Erklärung bedürfen:

- Die **Marktkapitalisierung** ist der aktuelle Marktwert einer Kryptowährung. Er ergibt sich aus dem Produkt des aktuellen Handelskurses und der totalen Anzahl der geschürften Währungseinheiten. Bei einem Kurs von 500 USD pro Ether und einem Volumen von 1000 Ether, ist die Marktkapitalisierung 500000 USD.
- Das Handelsvolumen (engl. **trade volume**) ist die Anzahl der Kryptowährungseinheiten, die in einem bestimmten Zeitfenster - meist 24 Stunden - gehandelt wurden.
- Die **Blocksize** ist die Größe eines Blocks (meinst in Megabyte).
- **Stale Blocks** sind Blöcke der Blockchain, die korrekt sind, aber ihren Weg in die Blockchain nicht gefunden haben. Dies kann passieren, wenn zwei Blöcke gleichzeitig fertiggestellt werden, jedoch nur einer davon in die Blockchain aufgenommen werden kann.(BitcoinProject, 2017c) Im Ethereumnetzwerk heißen diese Blöcke '**Uncles**'. Entgegen dem Bitcoinnetzwerk werden dort Miner für Stale Blocks (Uncles) entlohnt.(Jdebunt, 2017) Es besteht auch die Möglichkeit eines '**Orphaned**' Blocks. Das ist ein Block, über dessen Elternblöcke ein berechnender Knoten des Netzwerks keine Informationen hat und ihn deswegen nicht validieren kann.(BitcoinProject, 2017b)
- Die Bestätigungszeit (engl. **confirmation time**) ist die Zeit, die zwischen einer Transaktion und ihrer Aufnahme in die Blockchain vergeht.(Kumar, 2017)

- Bei der **Hashrate** handelt es sich um die Anzahl der Hashes eines Netzwerk pro Sekunde. Trifft man auf den Begriff im Kontext des Bitcoins, handelt es sich meist um Terahashes, bei Ethereum um Gigahashes (entspricht $\frac{1}{1000}$ Terahash).(Kumar, 2017)
- Miningschwierigkeit (**difficulty**) meint die Schwierigkeit, einen validen Block zu finden. Als Referenz dient die einfachste Möglichkeit einen Block zu finden mit der Schwierigkeit 1.(BitcoinProject, 2017a)
- Aufgebrachte Rechenleistung wird im Ethereumnetzwerk in **Gas** (engl. Aussprache) bemessen. Wie in Teil 1.2 bereits genannt, können im Netzwerk beispielsweise Smartcontracts ausgeführt werden. Die Kosten für die dort verbrauchte Leistung wird in Gas gezahlt. Wenn die Leistung vorher nicht abgemessen werden kann, wird ein Limit (**gas limit**) gesetzt, nach dessen Verbrauch eine Aktion abgebrochen wird.(Wood, 2014, S. 4)

6 Microsoft Azure ML Studio

Im nun folgenden Abschnitt der Arbeit wird das Microsoft Azure ML Studio in drei Schritten erklärt:

1. Zuerst erfolgt eine allgemeine Beschreibung des Machine Learning Studios.
2. Dann wird der Aufbau eines Projektes beschrieben und
3. zuletzt die vorgefertigten Komponenten vorgestellt.

Es sollte in diesem Teil beachtet werden, dass die Quellen hauptsächlich Microsoft-eigen sind, da außer einigen Blockartikeln wenig externe Literatur oder Meinungen verfügbar sind.

6.1 Allgemeine Beschreibung

Zum Ende des Abschnittest 1.4 wurde die **S**oftware **a**s **a** **S**ervice (SaaS) Microsoft Azure Machine Learning Studio bereits angesprochen. Es handelt sich dabei um eine Cloud-basierte Web-IDE (**I**ntegrated **D**evelopment **E**nvironment). Der Kern der Anwendung ist ein „drag and-drop tool“, das genutzt wird um „predictive analytics solutions“ zu entwerfen und bereitzustellen.(Ericson and Rohm, 2017c) Es können sowohl vorgefertigte Bausteine genutzt werden, als auch selbst entworfene Scripte (in R und/oder Python).

6.2 Aufbau und Komponenten

Die strukturellen Hauptkomponenten sind Tabelle 6.1 zu entnehmen. Die Tabelle basiert auf dem Artikel „What is Azure Machine Learning Studio?“ von (Ericson and Rohm, 2017c) und dem Machine Learning Studio selbst (https://studio.azureml.net/Home).

Komponente	**Beschreibung**
Settings	Hier finden sich Einstellmöglichkeiten zur Benutzerverwaltung (für kollaborative Arbeit im Studio), das Preismodell (10GB Workspace Storage sind in der kostenfreien Version enthalten) und zu den Authorisierungstoken. Die Settings sind in diesem Kontext nicht weiter interessant.
Notebooks	Die Notebooks sind Verknüpfungen auf Jupyter Notebooks. Diese wiederum sind „Webanwendungen“, die genutzt werden um „Dokumente, die live code, Gleichungen, Virtualisierungen und erzählenden Text“ enthalten „zu erstellen und zu teilen“. Genutzt werden sie unter andrem für • „data cleaning“, • „transformation, • numerical simulation, • statistical modeling, • data visualization“ und • „machine learning“(ProjectJupyter, 2017b). Project Jupyter ist non-profit und open-source.(ProjectJupyter, 2017a)
Experiments	Experimente sind das Kernelemente des Azure ML Studios. Sie stellen den Rahmen für das erproben verschiedener Problemlösungen. Experimente sollten dabei mindestens ein Datenset (dazu gleich mehr) und ein Modul enthalten. Ist ein Experiment ausgereift, kann der Status von „training“ to „predictive“ geändert werden und als Web Service bereit gestellt werden (dazu ebenfalls nachfolgend mehr).

Datasets	Für das Experimentieren benötige Datensätze (z.B. .csv-, .tsv-, .arff-, .txt-, .zip-, oder .RData-Dateine) können in die Web-IDE geladen und als Datasets abgespeichert werden. So können die Dateien ohne erneutes hochladen wiederverwendet werden. Eine andere Möglichkeit, Daten im Studio zu benutzen, ist das Experiment Item „Import Data“, das unter anderem auf Daten aus Azure Blob Storage, Azure Table Storage, Azure DocumentDB, Azure SQL Datenbanken, Hive Queries oder HTTP-Quellen zugreifen kann.
Experiment Items	Die Sammlung aller drag-and-drop Elemente im Studio werden Experiment Items genannt. Sie umfassen Elemente für alle Prozessschritte, wie Input, Transformation, Selection, Machine Learning Module, Visualisierung, Scoring etc.
Trained Models	Fertig trainierte Machine Learning Module können als eigenständige Komponenten genutzt werden. Ein Beispiel dafür wäre die Wiederverwendung in einem anderen Experiment mit neuen Daten oder der Vergleich zweier fertiger Modelle.
Web Services	Ist ein Experiment auf dem Status „predictive“, können die Input- und Output-Felder in Web Services umgewandelt werden, um sie in andere Anwendungen oder Systeme zu integrieren.
Projects	Projects dienen als Organisatorisches Strukturelement. Durch sie können Experimente, Datasets, Notebooks etc. gruppiert werden.

Tabelle 6.1: Hauptkomponenten des Azure Machine Learning Studios

7 Praxis: Durchführung der Analyse

Nachdem in den vorhergegangen Abschnitten alle Aspekte des Themas „Kursanalyse von Kryptowährungen mit Azure Machine Learning“ betrachtet wurden, widmet sich dieser Abschnitt der praktischen Umsetzung. Jeder Themenblock hat seinen Teil zum Erstellen eines Kontextes beigetragen, in dem die Analyse durchgeführt werden kann (siehe Tabelle 7.1).

Themenblock	**Inhalt**	**Ziel**
Data Mining Frameworks (Kapitel 3)	Beschreibung der bekanntesten Frameworks des Data Mining Prozesses und Auswahl eines Frameworks für die vorliegende Arbeit	Detailliertes Beschreiben des nachfolgend genutzten Prozessmodells
Machine Learning (Kapitel 4)	Vorstellung einer Möglichkeit zur Einordnung von Machine Learning Typen und Algorithmen	Verständnisaufbau für die Analyse in diesem Teil der Arbeit
Kryptowährungen (Kapitel 5)	Hintergrundwissen zu Kryptowährungen	Für eine Analyse ist Hintergrundwissen wichtig. Dieses sogenannte 'Domain'-Wissen ist essentieller Bestandteil einer Analyse mit CRISP-DM.
Microsoft Azure Machine Learning Studio (Kapitel 6)	Allgemeine Beschreibung und Aufbau des Werkzeugs	Das Studio soll als Werkzeug zur Analyse eingesetzt werden.

Tabelle 7.1: Behandelte theoretische Abschnitte im Kontext der Arbeit

Wie in Punkt 3.3 angesprochen, wird als Hilfe für das Prozessmodell CRISP-DM der zugehörige User Guide (Chapman et al., 2000, S. 30-56) herangezogen. Die, im Guide genannten, Outputs jedes Prozessschritts werden nachfolgend (wenn möglich) in Tabellen hervorgehoben. Das

CRISP-DM Prozessmodell ist sehr generisch gehalten. Dies ist beispielsweise daran zu erkennen, dass das Modell in vier übereinander liegende Abstraktionsschichten gegliedert ist.(Chapman et al., 2000, S. 6) Dies dient der Anpassungsfähigkeit an viele heterogene Projekte. Diese Anpassungsfähigkeit wird gleich im ersten Schritt genutzt. Die englischen Bezeichnungen in den Überschriften zeigen an, um welchen CRIPS-DM-Prozessschritt es sich handelt.

7.1 Business Understanding

7.1.1 Festlegung der fachlichen Projektziele (Determine the Business Objectives)

In gewöhnlichen Industrie- oder Forschungsprojekten ist es wichtig, die Stakeholder (vor allem Geldgeber) und den Reifegrad und die Akzeptanz des Data Mining im Projektumfeld zu analysieren. Dies rückt im vorliegenden Fall in den Hintergrund. Die Anderen Outputs sind jedoch ebenso wichtig.

Output	**Beschreibung**
Background	Die Analyse wird im Rahmen einer Masterarbeit durchgeführt. Nur eine Person ist daran beteiligt.
Business objectives	Die Untersuchung hat zwei Hauptziele. Das eine ist die Analyse der Kryptowährungen an sich. Es soll herausgefunden werden, ob der Kurs mit Hilfe der Isolation von Einflussfaktoren und den Mitteln des Machine Learning vorausgesagt werden kann. Das andere Ziel ist die Einarbeitung in das Werkzeug Azure Machine Learning Studio.
Business success criteria	Die Erkenntnis, dass eine Vorhersage nicht möglich ist, oder dass wichtige Einflüsse nicht gefunden wurden, ist durchaus möglich und bedeutet keinesfalls ein Scheitern des Projekts. Hinsichtlich des Werkzeugs Azure ML, ist es beispielsweise interessant, welchen Restriktionen das Tool unterlegen ist. Das Betrifft sowohl Funktionen, die (noch) nicht vorhanden sind, oder technische Limitationen, wie Geschwindigkeit, Volumenbegrenzungen etc..

Tabelle 7.2: Output des Schrittes „Determine the Business Objectives“

7.1.2 Aufstellung der Projetressourcen und Einflussfaktoren (Assess the Situation)

Dieser Teil befasst sich vor allem damit, welche Ressourcen zur Verfügung stehen (Hardware, Software, personell) und welche sonstigen Bedingungen erfüllt sein müssen oder das Projekt begrenzen. Dazu zählt auch das Finden von Daten, die für die Modellierung genutzt werden können. Anzumerken ist hierbei, dass es noch nicht um das tatsächliche Laden der Daten im Sinne von Dateien geht, sondern um das Finden von Quellen für Daten. Zusätzlich sollen noch eine Risiko- und eine Kosten-Nutzen-Analyse durchgeführt werden. Das Hauptaugenmerk liegt jedoch auf der Erschließung der Daten.
An Stelle einer vollständigen Risikoanalyse (Kontext herstellen, Risiken identifizieren, analysieren, evaluieren, managen(Sowa, 2017, S. 43)), tritt eine Aufzählung der zwei Hauptrisiken. Dies geschieht einerseits aus Gründen der Verhältnismäßigkeit, andererseits liegt der Fokus der Arbeit auf einem anderen Thema. Ähnlich verhält es sich mit der Kosten-Nutzen-Analyse. Sie wird zur Bewertung der Wirtschaftlichkeit herangezogen, was in diesem Kontext nicht relevant ist. Deswegen wird auf sie vollständig verzichtet.

Output	Beschreibung

Inventory of resources	Personal • 1 Person mit Zugang zu den Recherche Ressourcen der Hochschule München (OPAC, DBIS, ZDB etc.(HochschuleMünchen, 2017)) Hardware • 1 PC (CPU: AMD Ryzen 5 1600 Sechskern; RAM: 8GB; GPU: NVIDIA GeForce GTX 1060 (6GB VRAM); Windows 10 Education Build 15063.674) Software • 1 „Free“-Account Microsoft Azure Machine Learning Studio mit Workspace in „South Central US“ • Version des Juypter Notebooks 5.1.0 • Auf dem PC: R Version 3.4.1 • Auf dem PC: RStudio Version 1.0.153 • Excel 2016 (Microsoft Office 365 ProPlus) Version 1710 • Notepad++ Version 7.5.1 Daten • in Tabellen 7.4 bis 7.8 genannte Daten • Kurse BTC/USD und ETH/USD • Zusätzliche Eigenschaften von Bitcoin und Ethereum. (Kumar, 2017) stellt unter der 'CC0: Public Domain'-Lizenz einen Datensatz zur Verfügung, der besondere Eigenschaften der Währungen enthält. Beispiele für Bitcoin sind die „Anzahl der einzigartigen Adressen“ in der „Bitcoin Blockchain“; oder die „Anzahl der uncles pro Tag“(Kumar, 2017, eigene Übersetzung) für Ethereum.
Requirements, assumptions, and constraints	Zu Bedenken ist, dass bei einer kostenlosen Subscription im Azure ML Studio nur 10GB Storage verfügbar sind. Zusätzlich müssen Daten, die analysiert werden sollen, in das Tool geladen werden. Bei einem Upload vom PC wird das durch die Upload-Bandbreite limitiert. Eventuell müssen Daten im Projektverlauf auch öfter Hochgeladen werden, was zu Verzögerungen führen könnte. Außerdem lässt sich in der freien Version nur jeweils ein Experiment gleichzeitig ausführen. Das parallele Trainieren von Modellen ist somit nicht möglich. Obwohl zur Hilfe neben der Web-IDE auch RStudio genutzt werden kann, soll die Untersuchung hauptsächlich mit den Azure ML Studio Bordmitteln durchgeführt werden. Schließlich sollen nur solche Daten genutzt werden, die entweder frei verfügbar sind oder mit dem Hochschulzugang zu beschaffen sind. Das schließt präparierte Datensätze von Bezahlseiten aus.

Risks and contingencies	• keinen Zugriff auf Azure ML Studio (Server-seitige Probleme; Ablauf der Free-Subscription) • benötigte Daten nicht verfügbar
Terminology	Als Glossar dienen die 'Abkürzungen und Erklärungen' zu Beginn der Arbeit.
Costs and benefits	—

Tabelle 7.3: Output des Schrittes „Assess the Situation“

Die schwerste Aufgabe dieses Teils ist das Finden von Daten, die Einfluss auf den Kurs von Kryptowährungen haben (könnten). Forschungen in diesem Bereich identifizierten einerseits öffentliches Interesse (soziale Medien, Google Suchanfragen, etc.)(Kristoufek, 2013; Garcia et al., 2014) und andererseits auch wirtschaftliche Faktoren („standard economic theory“, also Angebot und Nachfrage, Investoren)(Kristoufek, 2015) als Haupteinflussfaktoren. Anhand den Aussagen dieser Paper und zusätzlichen Überlegungen ergeben sich folgende Faktoren, die bei der Analyse bedacht werden (Tabellen 7.4 bis 7.8).
Darüber hinaus wird die Quelle aufgeführt, über die die Daten bezogen werden. Bei monetären Strömen oder Kursen, wird immer der Kurs in USD herangezogen.
Damit eine Analyse der Kryptowährungen möglich ist, müssen zu diesen historische Kursdaten beschafft werden. Dieser, auf den ersten Blick simpel wirkende Schritt, ist jedoch nicht trivial. Kryptowährungen werden an dutzenden Portalen gleichzeitig gehandelt. Dabei erscheinen genauso schnell neue Börsen, wie alte verschwinden. Auch das Handelsvolumen und die Handelswährung unterscheidet sich. Hinzu kommt, dass an Bitcoin- oder Ethereum-Handelsportalen meist 24 Stunden täglich gehandelt werden kann. Aus diesem Grund sind solche Datenquellen ausgewählt worden, die entweder ihre Daten direkt von der Blockchain erhalten oder einen gewichteten Mittelwert über die größten Handelsplattformen berechnen.
Die nächste Schwierigkeit ist die Auswahl der Aktienindizes, die für die Analyse herangezogen werden. Es existiert keine allgemein gültige oder anerkannte Liste mit 'den wichtigsten Aktienindizes'. Aus diesem Grund wurden die Indizes ausgewählt, die die Website investing.com als „major world indices“ deklariert.(FusionMediaLimited, 2017) Es liegt dabei eine große Überschneidung mir anderen Stock Market-Seiten vor.(liveindex.org, 2017; YahooFinance, 2017) Zusätzlich zu den Aktienindizes wird ein Financial Stress Index (FSI) herangezogen. Ein solcher „Index misst die aktuelle Belastung in einem finanzwirtschaftlichen System“(Vermeulen et al., 2014, S. 1; eigene Übersetzung) In diesem Fall ist es der St. Louis

Fed Financial Stress Index, der 18 Einzelfaktoren aus drei Kategorien bündelt.(**?**)
Ferner werden die Währungen der acht größten Volkswirtschaften(Fund, 2017) und die Kurse für Gold, Silber und Rohöl mit einbezogen. Es werden die Wechselkurse zum Dollar betrachtet, sprich Fremdwährung/USD. Bei den Ölkursen wird Brent, das wichtigste Rohöl für den europäischen Markt (Wikimedia, 2016), und West Texas Intermediate (WTI), das Pendant für den US-Markt, betrachtet(Wikimedia, 2017).

Daten	**für BTC**	**für ETH**
Handelsvolumen	ja, von https://bitcoincharts.com/	ja, von https://coinmarketcap.com/
Coin Volumen (Gesamtanzahl der vorhandenen Bitcoins/des Ethers)	ja, von https://blockchain.info/	ja, von https://etherscan.io/
Mining-Schwierigkeit	ja, von https://data.bitcoinity.org/	ja, von https://etherscan.io/
Anzahl der Transaktionen	ja, von https://blockchain.info/	ja, von https://etherscan.io/
Hashrate	ja, von https://www.kaggle.com/	ja, von https://www.kaggle.com/
Marktkapitalisierung	ja, von https://www.kaggle.com/	ja, von https://www.kaggle.com/

Tabelle 7.4: Mögliche Einflussfaktoren: Kryptowährungs-eigene Faktoren (für Begriffserklärungen siehe Punkt 5)

Daten	**für BTC**	**für ETH**
Google Websuchen	ja, von https://trends.google.de/trends/	ja, von https://trends.google.de/trends/
Google News-Suchen	ja, von https://trends.google.de/trends/	ja, von https://trends.google.de/trends/
Wikipedia Seitenaufrufe	ja, von https://wikimedia.org/api/rest _v1/	ja, von https://wikimedia.org/api/rest _v1/
Tweets (Twitter Nachrichten)	nein, nicht kostenlos verfügbar	
Zeitungsartikel/-Überschriften	ja, von https://www.kaggle.com/therohk/million-headlines/data/	

Tabelle 7.5: Mögliche Einflussfaktoren: Öffentliches Interesse

Daten	**für BTC**	**für ETH**

Dow 30, S&P 500, Nasdaq, SmallCap, S&P 500 VIX, S&P/TSX, TR Canada, Bovespa, IPC, DAX, FTSE 100, CAC 40, Euro Stoxx, AEX, IBEX, FTSE MIB, SMI, PSI, BEL, ATX, OMXS30, OMXC20, MICEX, RTSI, WIG20, Budapest SE, BIST 100, TA 35, Tadawul All Share, Nikkei 225, S&P/ASX 200, DJ New Zealand, Shanghai, SZSE Component, China A50, DJ Shanghai, Hang Seng, Taiwan Weighted, SET, KOSPI, IDX Composite, Nifty, BSE Sensex, PSEi Composite, STI Index, Karachi, HNX 30, CSE All-Share	ja, von https://www.investing.com/indices/
St. Louis Fed Financial Stress Index (STLFSI)	ja, von https://fred.stlouisfed.org/series/STLFSI

Tabelle 7.6: Mögliche Einflussfaktoren: (Aktien)indizes

Daten	**für BTC**	**für ETH**
China (CNY), Japan (JPY), Deutschland (EUR, Großbritannien (GBP), Frankreich (EUR), Indien (INR), Brasilien (BRL)	ja, von https://www.investing.com/currencies/single-currency-crosses	

Tabelle 7.7: Mögliche Einflussfaktoren: Währungen der größten Volkswirtschaften (nach BIP)

Daten	**für BTC**	**für ETH**
Goldpreis, Silberpreis, Brent (Rohöl Europa), WTI (Rohöl USA)	ja, von https://www.investing.com/commodities/	

Tabelle 7.8: Mögliche Einflussfaktoren: natürliche Ressourcen

7.1.3 Festlegen der technischen Projektziele (Determine the Data Mining Goals)

Es sollen mit Hilfe von Azure Machine Learning Studio mehrere Machine Learning Models erzeugt werden. Sie sollen die Szenarien in Tabelle 7.9 vorhersagen (Data mining goals). Möglichkeiten für die Bewertung einer Klassifikationen sind:

- Accuracy: $\frac{t_p+t_n}{t_p+t_n+f_p+f_n}$ (Anteil der insgesamt richtig vorhergesagten Werte an allen Vorhersagen)

- Precision: $\frac{t_p}{t_p+f_p}$ (Anteil der richtig vorhergesagten true-Werte an allen vorhergesagten true-Werten)
- Recall: $\frac{t_p}{t_p+f_n}$ (Anteil der richtig vorhergesagten true-Werte an allen eigentlich richtigen true-Werten)
- F1-score: $2 \times \frac{Precision \times Recall}{Precision+Recall}$ (Harmonisches Mittel aus Precision und Recall; Bester Wert = 1)
- AUC: Fläche unter der Receiver Operating Characteristics (ROC) Kurve; „Je besser die Klassifizierungsfähigkeit des Klassifikators desto höher ist der AUC-Wert“(Lohninger, 2013, ROC-Kurve)

mit den Bezeichnungen

$t_p : true \quad positive$
$t_n : true \quad negative$
$f_p : false \quad positive$
$f_n : false \quad negative$

Möglichkeiten für die Bewertung einer Regressionen sind:

- Mean absolute error (MAE): durchschnittlicher, absoluter Unterschied zwischen vorhergesagten und tatsächlichen Werten; je kleiner desto besser(Mircosoft, 2017)
- Root mean squared error (RMSE): Macht eine Aussage darüber, „wie gut eine Funktionskurve an vorliegende Daten angepasst ist“; „je größer der RMSE [...], desto schlechter“(Statista)
- Relative absolute error (RAE), Relative squared error (RSE): Ähnlich dem RMSE, aber für den Vergleich von Regressionen mit unterschiedlichen Maßeinheiten geeignet.(Dr. Sayad, 2017)
- Coefficient of determination (R^2): Gütemaß für die Aussagekraft der Regression;
  $Regression \quad passt \quad perfekt := 1$
  $Regression \quad erklärt \quad nichts := 0;$
  kleine Werte sind normal, große sollten misstrauisch machen(Mircosoft, 2017)

Die Auswahl fällte auf den F1-Score für Klassifikationen, da er Recall und Precision inkludiert und im Gegensatz zum AUC zwischen mehreren Modellen direkt vergleichbar ist (Maximalwert 1 statt nach oben offen). Bei der Regression fällt die Wahl auf R^2, da er Modelle ebenfalls direkt vergleichbar macht.

Output	**Beschreibung**
Data mining goals	• Steigt der durchschnittliche Kurs innerhalb eines 24-Stunden Zyklus über den durchschnittlichen Kurs des vorhergehenden 24-Stunden Zyklus? (Mögliche Antworten sind $ja := 1$ und $nein := 0$) • Wie hoch ist der durchschnittliche Kurs des nächsten 24-Stunden Zyklus? (Antwort: Preis eines Bitcoins/Ethers in USD) Im ersten Fall handelt es sich um eine two-class classification (siehe 4.1.1). Die zweite Frage fällt in das Gebiet der Regressionen (siehe 4.1.2).
Data mining success criteria	• F1-Score für Klassifikationen [Zielwert 1] • R^2 für Regressionen [Zielwert 1]

Tabelle 7.9: Output des Schrittes „Determine the Data Mining Goals“

7.1.4 Aufstellung eines groben Projektplans und der genutzten Werkzeuge (Produce a Project Plan)

Nach diesem sehr Daten-orientierten Teilschritt, befasst sich der nachfolgende Schritt einerseits mit der (groben) Planung des weiteren Projekts und andererseits mit einer anfänglichen Betrachtung der eingesetzten Werkzeuge. Grundsätzlich orientiert sich der Projektplan am Referenzmodell CRISP-DM. In einem komplexeren Projekt mit mehreren kollaborativ arbeitenden Personen, kommt diesem Schritt eine größere Rolle zu als im vorliegenden Fall. Wichtige Meilensteine, die zu bestimmten Zeitpunkten fertig vorliegen müssen, um einen unterbrechungsfreien Projektverlauf zu gewährleisten, stellen die Outputs jedes Prozessschrittes dar. Eine zeitliche Einschätzung ist dem Output „Project Plan“ in Tabelle 7.10 zu entnehmen. Zu sehen ist dort der geschätzte Arbeitsaufwand in Tagen für jeden Schritt und der Anteil am Gesamtprojekt. Anzumerken ist hier, dass die Angaben durch Rückschritte in frühere

Phasen verzerrt werden können. Auf das Deployment wird verzichtet.
Bei der initialen Betrachtung der Werkzeuge werden laut Prozessmodell verschiedene alternativen gegeneinander abgewogen und ihr Zweck im Projekt festgelegt. Da das Werkzeug Azure Machine Learning Studio bereits vorgegeben ist, wird hier nur der Zweck der Werkzeuge im Projektverlauf betrachtet. Wird das Werkzeug R (bzw. RStudio) eingesetzt, so erhält es den Vorzug gegenüber seiner Konkurrenz (z.b. Python) aufgrund der Erfahrung des Entwicklers.

Prozessschritt	**geschätzte Zeit in Tagen**	**Prozentualer Anteil**
Business understanding gesamt	5	11%
Data understanding gesamt	20	44%
Data preparation gesamt	10	22%
Modeling gesamt	5	11%
Evaluation gesamt	5	11%
Gesamt	$\sum 45$	99%

Tabelle 7.10: Output „Project Plan“ des Schrittes „Produce a Project Plan“

Werkzeug	**Zweck im Projekt**
Microsoft Azure Machine Learning Studio	Vorgabe im Projekt; soll in jeder Phase genutzt werden, wo es möglich ist
Juypter Notebooks	Integriert in Azure ML Studio; für R Code, der in der Cloud ausgeführt werden soll
R	zur Nutzung in Situationen für die Azure ML keine vorgefertigten Experiment Items bereitstellt
RStudio	zum lokalen entwerfen von Scripten vor der Ausführung in der Cloud und zur Vorbereitung der Datensätze
Excel	zur Betrachtung, Vorbereitung und Konvertierung von Daten vor dem Upload in die Cloud
Notepad++	zur Betrachtung, Vorbereitung und Konvertierung von Daten vor dem Upload in die Cloud

Tabelle 7.11: Output „Initial assessment of tools and techniques“ des Schrittes „Produce a Project Plan“

7.2 Data Understanding

7.2.1 Beschaffung der Daten und grobe Beschreibung (Collect the Initial Data)

In Schritt 7.1.2 wurden Einflussfaktoren auf den Kurs festgehalten. Zusätzlich wurden bereits Quellen für Daten zu diesen Faktoren gesucht. In diesem Schritt werden nun die Daten tatsächlich (im Sinne von 'echten' Dateien) bezogen und bereits in Azure Machine Learning geladen. Ebenfalls Teil dieses Schrittes ist es, die Daten grob zu beschreiben. Im vorliegen Fall werden die Datumsspanne und weitere offensichtliche Merkmale beschrieben. Die Herkunft der Daten ist nicht weiter beschrieben, da dies bereits in 7.1.2 abgehandelt wurde.
Alle gesammelten Daten enthalten Header-Informationen (Preis, Kurs, Datum etc.). Azure ML Studio kann diese Header auslesen und nutzt sie für einfache Select- oder Join-Befehle. Damit dies möglich ist, müssen die Daten als 'Genereic CSV File with a header (.csv)' oder 'Generic TSV File with a header (.tsv)' vorliegen. In manchen Fällen ist es möglich, dass Microsoft Excel beim Speichern einer Datei als '.csv' statt Kommas, Strichpunkte zum Trennen nutzt. Hier ist darauf zu achten, dass tatsächlich Kommas als Trennzeichen genutzt werden, da Azure ML Studio diese sonst nicht interpretieren kann. Die Datei-Endung '.tsv' ist nicht zwingend erforderlich. Ein 'Tab seperated values'-File kann auch mit der Endung '.txt' hochgeladen und richtig interpretiert werden. Dies ist wichtig, da Excel als Endung für diese Dateien nicht '.tsv' sondern '.txt' wählt.
Azure ML Studio bietet die Möglichkeit, mehrere Dateien gezippt in das Tool zu laden. Dies bietet sich an, wenn viele einzelne Dateien hochgeladen werden sollen. Leider kann das Zip-File pro Experiment nur einmal verwendet werden und nur eine Datei daraus kann entpackt werden. Dadurch ist diese Option im Falle dieser Arbeit nicht praktikabel. Es müssen also alle Dateien einzeln geladen werden.

Kryptowährungs-eigene Faktoren		
Datensatz	**Spanne**	**Besonderheiten**
BTC _Total _Volume _Daily _Full	3.1.2009 bis 5.11.2017	teilweise kein Anstieg des Volumens im Datensatz; große Lücken im Datensatz (2012-2016)
BTC _Difficulty _Daily _Full	6.11.2012 bis 5.11.2017	Zeit in UTC
BTC _Transaction _Number _Fully _Daily	3.1.2009 bis 5.11.2017	

BTC _Price _Multiple _Daily	17.7.2010 bis 5.11.2017	Zeit in UTC; sehr Lückenhaft, da erst ab 2016 alle enthaltenen Börsen operieren
ETH _Total _Volume _Daily _Full	30.7.2015 bis 5.11.2017	Zeit in UTC und UnixTimeStamp
ETH _Difficuly _Daily _Full	30.7.2015 bis 5.11.2017	Zeit in UTC und UnixTimeStamp
ETH _Transaction _Number _Fully _Daily	30.7.2015 bis 5.11.2017	Zeit in UTC und UnixTimeStamp; ganz am Anfang einige 0-Werte
Öffentliches Interesse		
Datensatz	**Spanne**	**Besonderheiten**
google _Trends _BTC _Websearch	01.2009 bis 11.2017	Daten im Abstand von 1 Monat; auf einer Scala von 0-100; erster nicht-null-Wert bei 05.2011
google _Trends _ETH _Websearch	01.2011 bis 11.2017	Daten im Abstand von 1 Monat; auf einer Scala von 0-100; erster nicht-null-Wert bei 08.2014
google _Trends _BTC _Newssearch	01.2009 bis 11.2017	Daten im Abstand von 1 Monat; auf einer Scala von 0-100; erster nicht-null-Wert bei 04.2011
google _Trends _ETH _Newssearch	01.2011 bis 11.2017	Daten im Abstand von 1 Monat; auf einer Scala von 0-100; erster nicht-null-Wert bei 07.2014
Wiki _Page _Views _BTC	1.7.2015 bis 7.11.2017	Enthält Datum als zusammengesetzte Zahl im Format yyyyddmm00; enthält 7 Features, von denen 5 für jede Zeile gleich sind
Wiki _Page _Views _ETH	1.7.2015 bis 7.11.2017	Enthält Datum als zusammengesetzte Zahl im Format yyyyddmm00; enthält 7 Features, von denen 5 für jede Zeile gleich sind
abcnews _Date _Text	19.2.2003 bis 30.9.2017	großer Datensatz; viel Text; kann von Excel nicht komplett geöffnet werden
(Aktien)indizes		
Datensatz	**Spanne**	**Besonderheiten**
AEX, BFX, XU100, BVSP, VIX, CSE, GDAXI, DJI, FTSE, FTMIB, HSI, IBEX, MXX, JKSE, KSE, KS11, MCX, IXIC, NSEI, N225, OMXC20, OMXS30, IRTS, SPX, AXJO, GSPTSE, SSEC, SSMI, TA35, TASI, TRX50CAP, US2000, WIG20	1.1.2009 bis 9.11.2017	alle Indizes enthalten Lücken für Wochenenden und Feiertage; Datum in einem Format mit Text; gilt auch für nachfolgende Datensätze mit anderen Zeitspannen

ATX (ATX) _history.txt	23.3.2015 bis 9.11.2017	verhältnismäßig kurze Zeitspanne
BSE Sensex 30 (BSESN) _history.txt	24.2.2011 bis 9.11.2017	
Budapest SE (BUX) _history.txt	7.3.2011 bis 9.11.2017	
Dow Jones New Zealand (NZDOW) _history.txt	25.8.2011 bis 9.11.2017	
Dow Jones Shanghai (DJSH) _history.txt	6.3.2011 bis 9.11.2017	
Euro Stoxx 50 (STOXX50E) _history.txt	15.8.2011 bis 9.11.2017	
FTSE China A50 (FTXIN9) _history.txt	19.3.2010 bis 9.11.2017	
FTSE Straits Times Singapore (STI) _history.txt	7.3.2011 bis 9.11.2017	
HNX 30 (HNX30) _history.txt	4.11.2014 bis 9.11.2017	verhältnismäßig kurze Zeitspanne
PSEi Composite (PSI) _history.txt	3.11.2011 bis 9.11.2017	
PSI 20 (PSI20) _history.txt	25.4.2010 bis 9.11.2017	
SET Index (SETI) _history.txt	18.3.2011 bis 9.11.2017	
SZSE Component (SZSC1) _history.txt	14.9.2012 bis 9.11.2017	
Taiwan Weighted (TWII) _history.txt	17.3.2011 bis 9.11.2017	
STLFSI _history.csv	2.1.2009 bis 27.10.2017	Zeilen im Abstand einer Woche; Index kann positive und negative Werte annehmen

Währungen der größten Volkswirtschaften		
Datensatz	**Spanne**	**Besonderheiten**
CNY _USD _history	1.1.2009 bis 13.11.2017	Datum in einem Format mit Text; Lücken an Wochenenden und Feiertagen
JPY _USD _history	12.5.2009 bis 13.11.2017	Datum in einem Format mit Text; Lücken an Wochenenden und Feiertagen
EUR _USD _history	1.1.2009 bis 13.11.2017	Datum in einem Format mit Text; Lücken an Wochenenden und Feiertagen

GBP _USD _history	1.1.2009 bis 13.11.2017	Datum in einem Format mit Text; Lücken an Wochenenden und Feiertagen
INR _USD _history	09.09.2014 bis 13.11.2017	Datum in einem Format mit Text; Fehlende Daten vor dem 09.09.2014; Lücken an Wochenenden und Feiertagen
BRL _USD _history	1.1.2009 bis 13.11.2017	Datum in einem Format mit Text; Lücken an Wochenenden und Feiertagen
natürliche Ressourcen		
Datensatz	**Spanne**	**Besonderheiten**
gold _history	1.1.2009 bis 13.11.2017	Datum in einem Format mit Text; Lücken an Wochenenden und Feiertagen
silver _history	1.1.2009 bis 13.11.2017	Datum in einem Format mit Text; Lücken an Wochenenden und Feiertagen
oil _brent _history	1.1.2009 bis 13.11.2017	Datum in einem Format mit Text; Lücken an Wochenenden und Feiertagen
oil _wti _history	1.1.2009 bis 13.11.2017	Datum in einem Format mit Text; Lücken an Wochenenden und Feiertagen
ETH/USD-Kurs		
Datensatz	**Spanne**	**Besonderheiten**
ETH _Price _Volume _Full _Daily	7.8.2015 bis 5.11.2017	historische Kursdaten für ETC (Preis und Volumen); Datum in einem Format mit Text; chronologisch absteigend sortiert (ältestes Datum zum Schluss)
BTC/USD-Kurs		
Datensatz	**Spanne**	**Besonderheiten**
BTC _Price _Volume _Full _Daily	13.9.2011 bis 6.11.2017	historische Kursdaten für BTC (Preis und Volumen); am Anfang einige Lücken
zusätzliche Eigenschaften		
Datensatz	**Spanne**	**Besonderheiten**
bitcoinDataset	6.10.2009 bis 30.10.2017	sehr detaillierte Bitcoin Eigenschaften; 24 Features, z.B. Anzahl der einzigartigen Adressen im Netzwerk oder Kosten pro Transaktion
ethereumDataset	30.7.2015 bis 3.10.2017	sehr detaillierte Ethereum Eigenschaften; 18 Features, z.B. Anzahl der Adressen im Ethereum-Netzwerk oder Anzahl der Blocks und Uncles

Tabelle 7.12: Output „Initial data collection report“ des Schrittes „Collect the Initial Data“

Wie in Tabelle 7.12 zu sehen ist, umfassen die Daten verschiedene Zeitspannen. Einige

Aktienindizes, vor allem ATX und HNX30, existieren noch nicht so lange, wie die historischen Daten des Bitcoins. Es gibt mehrere Wege, um damit umzugehen. Da auch ohne diese noch über 30 andere Indizes zur Verfügung stehen, fließen sie nicht weiter in die Untersuchung ein. Einige der Datensätze beinhalten nicht nur rohe Daten, sondern bereits Brechungen, wie den prozentualen Anstieg eines Kurses zum Vortag. Obwohl die Features des Datensatzes erst im nachfolgenden Schritt beschrieben werden, kann hier schon festgehalten werden, dass diese 'Zusatzinformationen' von der Analyse ausgeschlossen werden. Es handelt sich dabei lediglich um eine - für den menschlichen Betrachter einfach zu verstehende - andere Schreibweise für die Daten in einer Zeitreihe und nicht um zusätzliche Informationen.

7.2.2 Genaue Beschreibung der Daten (Describe the Data)

Eine detaillierte Beschreibung der oben genannten Datensätze erfolgt nun. Für jeden Satz, bzw. jede homogene Gruppe (wie die Aktienindizes), werden die Anzahl der Spalten (Features), die Anzahl der Reihen (Observations) und die Dateigröße (in Kilobyte; KB) angegeben. Außerdem werden die Features genauer erläutert (Datentyp, Besonderheit). Beim Datentyp 'numerisch' handelt es sich um eine Ganzzahl, bei 'numerisch (5)' um eine Gleitkommazahl mit bis zu fünf Nachkommastellen.

Datensatz	BTC _Total _Volume _Daily _Full	
Observations	1615	
Features	2	
Dateigröße	32	
Feature	**Datentyp**	**Besonderheit**
Date	Datum im Format 'dd/mm/yyyy HH:MM'	HH:MM unbenutzt
Volume	numerisch (1)	extrem große Lücke zwischen 20.9.2012 und 9.7.2016; eventuell nicht zu gebrauchen; zwischen 50.0 und 16665662.5

Tabelle 7.13: Data description report für BTC _Total _Volume _Daily _Full

Datensatz	BTC _Difficulty _Daily _Full
Observations	1826
Features	2
Dateigröße	78

Feature	Datentyp	Besonderheit
Date	Datum im Format 'yyyy-mm-dd HH:MM:SS UTC'	HH:MM:SS unbenutzt
Difficulty	numerisch (2)	Inkonsistenzen in der Difficulty, teilweise um Faktor 100 unterschiedliche Werte in aufeinanderfolgenden Zeilen; manche mit Nachkommastellen, manche ohne

Tabelle 7.14: Data description report für BTC _Difficulty _Daily _Full

Datensatz	BTC _Transaction _Number _Fully _Daily	
Observations	1615	
Features	2	
Dateigröße	46	
Feature	**Datentyp**	**Besonderheit**
Date	Datum im Format 'dd/mm/yy HH:MM'	HH:MM:SS unbenutzt
Transactions	numerisch (0)	

Tabelle 7.15: Data description report für BTC _Transaction _Number _Fully _Daily

Datensatz	BTC _Price _Multiple _Daily	
Observations	2669	
Features	11	
Dateigröße	279	
Feature	**Datentyp**	**Besonderheit**
Time	Datum im Format 'dd/mm/yy HH:MM'	HH:MM:SS unbenutzt
bit-x	numerisch	Wert eines Bitcoins in USD an dieser Börse; mit Lücken
bitbay	numerisch	Wert eines Bitcoins in USD an dieser Börse; mit Lücken
cex.io	numerisch	Wert eines Bitcoins in USD an dieser Börse; mit Lücken
coinbase	numerisch	Wert eines Bitcoins in USD an dieser Börse; mit Lücken
exmo	numerisch	Wert eines Bitcoins in USD an dieser Börse; mit Lücken

gemini	numerisch	Wert eines Bitcoins in USD an dieser Börse; mit Lücken
hitbtc	numerisch	Wert eines Bitcoins in USD an dieser Börse; mit Lücken
itbit	numerisch	Wert eines Bitcoins in USD an dieser Börse; mit Lücken
kraken	numerisch	Wert eines Bitcoins in USD an dieser Börse; mit Lücken
others	numerisch	Wert eines Bitcoins in USD an anderen Börsen; ohne Lücken

Tabelle 7.16: Data description report für BTC _Price _Multiple _Daily

Datensatz	ETH _Total _Volume _Daily _Full	
Observations	830	
Features	3	
Dateigröße	33	
Feature	**Datentyp**	**Besonderheit**
Date(UTC)	Datum im Format 'm/dd/yyyy'	
UnixTimeStamp	numerisch (0)	zusätzliches Datum als Unix Timestamp
Value	numerisch (0)	von 7204930659375 (min) bis 9554710634375 (max)

Tabelle 7.17: Data description report für ETH _Total _Volume _Daily _Full

Datensatz	ETH _Difficuly _Daily _Full	
Observations	830	
Features	3	
Dateigröße	32	
Feature	**Datentyp**	**Besonderheit**
Date(UTC)	Datum im Format 'm/dd/yyyy'	
UnixTimeStamp	numerisch (0)	zusätzliches Datum als Unix Timestamp
Value	numerisch (0)	

Tabelle 7.18: Data description report für ETH _Difficuly _Daily _Full

Datensatz	ETH _Transaction _Number _Fully _Daily
Observations	830

Features	3	
Dateigröße	31	
Feature	**Datentyp**	**Besonderheit**
Date(UTC)	Datum im Format 'm/dd/yyyy'	
UnixTimeStamp	numerisch (0)	zusätzliches Datum als Unix Timestamp
Value	numerisch (0)	

Tabelle 7.19: Data description report für ETH _Transaction _Number _Fully _Daily

Datensatz	google _Trends _BTC _Newssearch und google _Trends _BTC _Websearch	
Observations	107	
Features	2	
Dateigröße	2	
Feature	**Datentyp**	**Besonderheit**
Monat	Datum im Format 'yyyy-mm'	keine Stelle für 'Tag'
bitcoin: (Weltweit)	numerisch (0)	Scala von 0 bis 100

Tabelle 7.20: Data description report für google _Trends _BTC _Newssearch und google _Trends _BTC _Websearch

Datensatz	google _Trends _ETH _Newssearch und google _Trends _ETH _Websearch	
Observations	83	
Features	2	
Dateigröße	1	
Feature	**Datentyp**	**Besonderheit**
Monat	Datum im Format 'yyyy-mm'	keine Stelle für 'Tag'
Ethereum: (Weltweit)	numerisch (0)	Scala von 0 bis 100

Tabelle 7.21: Data description report für google _Trends _ETH _Newssearch und google _Trends _ETH _Websearch

Datensatz	Wiki _Page _Views _BTC
Observations	861
Features	7
Dateigröße	57

Feature	Datentyp	Besonderheit
project	text	immer 'en.wikipedia'
article	text	immer 'Bitcoin'
granularity	text	immer 'daily'
timestamp	Datum im Format 'yyyymmddhh'	hh immer 00
access	text	immer 'all-access'
agents	text	immer 'all-agents'
views	numerisch (0)	

Tabelle 7.22: Data description report für Wiki _Page _Views _BTC

Datensatz	Wiki _Page _Views _ETH	
Observations	861	
Features	7	
Dateigröße	57	
Feature	**Datentyp**	**Besonderheit**
project	text	immer 'en.wikipedia'
article	text	immer 'Ethereum'
granularity	text	immer 'daily'
timestamp	Datum im Format 'yyyymmdd00'	hh immer 00
access	text	immer 'all-access'
agents	text	immer 'all-agents'
views	numerisch (0)	

Tabelle 7.23: Data description report für Wiki _Page _Views _ETH

Datensatz	abcnews _Date _Text	
Observations	mehr als 1048576 (Excel Maximum)	
Features	2	
Dateigröße	53480	
Feature	**Datentyp**	**Besonderheit**
publish _date	Datum im Format 'yyyymmdd'	mehrere (hundert) Einträge für einen Tag
headline _text	Text	ohne Satzzeichen; alles in Kleinbuchstaben

Tabelle 7.24: Data description report für abcnews _Date _Text

Datensatz	alle Aktienindizes	
Observations	unterschiedlich; bei vollständigen (1.1.2009 bis 5.11.2017) ca. 2230	
Features	7	
Dateigröße	bei vollständigen (1.1.2009 bis 5.11.2017) 253 bis 102	
Feature	**Datentyp**	**Besonderheit**
Date	Datum im Format 'mmm dd, yyyy' (mmm als Text)	Monat als Text (z.B. 'Jan 02, 2009')
Price	numerisch (2)	Nachkommastellen durch Punkt abgetrennt; Tausender durch Komma
Open	numerisch (2)	Nachkommastellen durch Punkt abgetrennt; Tausender durch Komma
High	numerisch (2)	Nachkommastellen durch Punkt abgetrennt; Tausender durch Komma
Low	numerisch (2)	Nachkommastellen durch Punkt abgetrennt; Tausender durch Komma
Vol.	Mischform aus Zahl und Text	Werte mit 'K' für Tausend (Kilo), 'M' für Millionen (engl. Millions) und 'B' für Milliarden (engl. Billions); Fehlende Werte mit Strich (-) gekennzeichnet
Change %	Prozentzahl mit Prozentzeichen (%)	negative und positive Werte

Tabelle 7.25: Data description report für alle Aktienindizes

Datensatz	STLFSI _history	
Observations	461	
Features	2	
Dateigröße	9	
Feature	**Datentyp**	**Besonderheit**
DATE	Datum im Format 'dd/mm/yyyy'	in wöchentlichem Abstand
STLFSI	numerisch (3)	im Intervall von [-1,586;3,246]

Tabelle 7.26: Data description report für STLFSI _history

Datensatz	alle Währungen
Observations	unterschiedlich; INR: 996, BRL: 2314, JPY: 2615, CNY: 2314, GBP: 2339, EUR: 2323

Features	6	
Dateigröße	INR: 51; BRL, CNY: 110; EUR, GBP: 111; JPY: 134	
Feature	**Datentyp**	**Besonderheit**
Date	Datum im Format 'mmm dd, yyyy' (mmm als Text)	Monat als Text (z.B. 'Jan 02, 2009')
Price	numerisch (4)	Nachkommastellen durch Punkt abgetrennt
Open	numerisch (4)	Nachkommastellen durch Punkt abgetrennt
High	numerisch (4)	Nachkommastellen durch Punkt abgetrennt
Low	numerisch (4)	Nachkommastellen durch Punkt abgetrennt
Change %	Prozentzahl mit Prozentzeichen (%)	negative und positive Werte

Tabelle 7.27: Data description report für alle Währungen

Datensatz	alle natürlichen Ressourcen	
Observations	unterschiedlich; Gold: 2289, Silber: 2692 Brent: 2290, WTI: 2283	
Features	7	
Dateigröße	unterschiedlich; Gold: 139, Silber: 142, Brent: 121, WTI: 118	
Feature	**Datentyp**	**Besonderheit**
Date	Datum im Format 'mmm dd, yyyy' (mmm als Text)	Monat als Text (z.B. 'Jan 02, 2009')
Price	numerisch (2)	Nachkommastellen durch Punkt abgetrennt
Open	numerisch (2)	Nachkommastellen durch Punkt abgetrennt
High	numerisch (2)	Nachkommastellen durch Punkt abgetrennt
Low	numerisch (2)	Nachkommastellen durch Punkt abgetrennt
Change %	Prozentzahl mit Prozentzeichen (%)	negative und positive Werte

Tabelle 7.28: Data description report für alle natürlichen Ressourcen

Datensatz	ETH _Price _Volume _Full _Daily	
Observations	822	
Features	7	
Dateigröße	51	
Feature	**Datentyp**	**Besonderheit**

Date	Datum im Format 'mmm dd, yyyy' (mmm als Text)	Monat als Text (z.B. 'Jan 02, 2009')
Open	numerisch (2)	Nachkommastellen durch Punkt abgetrennt
High	numerisch (2)	Nachkommastellen durch Punkt abgetrennt
Low	numerisch (2)	Nachkommastellen durch Punkt abgetrennt
Close	numerisch (2)	Nachkommastellen durch Punkt abgetrennt
Volume	numerisch (0)	Kommas zwischen Tausendern
Market Cap	numerisch (0)	Kommas zwischen Tausendern

Tabelle 7.29: Data description report für ETH _Price _Volume _Full _Daily

Datensatz	BTC _Price _Volume _Full _Daily	
Observations	2247	
Features	8	
Dateigröße	148	
Feature	**Datentyp**	**Besonderheit**
Date	Datum im Format 'dd/mm/yyyy HH:MM'	HH:MM ungenutzt
Open	numerisch (2)	Nachkommastellen durch Punkt abgetrennt
High	numerisch (2)	Nachkommastellen durch Punkt abgetrennt
Low	numerisch (2)	Nachkommastellen durch Punkt abgetrennt
Close	numerisch (2)	Nachkommastellen durch Punkt abgetrennt
Volume (BTC)	numerisch (0)	Kommas zwischen Tausendern
Volume (Currency)	numerisch (0)	Kommas zwischen Tausendern
Weighted Price	numerisch (0)	Kommas zwischen Tausendern

Tabelle 7.30: Data description report für BTC _Price _Volume _Full _Daily

Datensatz	bitcoinDataset	
Observations	2920	
Features	24	
Dateigröße	728	
Feature	**Datentyp**	**Besonderheit**
Date	Datum im Format 'dd/mm/yyyy HH:MM'	HH:MM ungenutzt
btc _market _price	numerisch (9)	
btc _total _bitcoins	numerisch (1)	nur Werte mit '.0' oder '.5' am Ende
btc _market _cap	numerisch (5)	
btc _trade _volume	numerisch (4)	Zu Beginn immer 0; enthält Lücken

btc _blocks _size	numerisch (4)	Zu Beginn immer 0; Nachkommastellen erst ab 20.4.2016
btc _avg _block _size	numerisch (16?)	Bis auf einen Wert (2.10.2017) immer < 0
btc _n _orphaned _blocks	numerisch (0)	Nur Werte im Intervall [0;7]
btc _n _transactions _per _block	numerisch (8)	Nachkommastellen erst ab 20.4.2016
btc _median _confirmation _time	numerisch (11)	Manche Werte scheinen Perioden darzustellen (7,86666666667 oder 7,93333333333)
btc _hash _rate	numerisch (17)	Große Spanne von 2.04e−6 bis 1.0e12
btc _difficulty	numerisch (11)	maximal 12 Stellen (Vorkommastellen + Nachkommastellen = 12)
btc _miners _revenue	numerisch (4)	
btc _transaction _fees	numerisch (8)	
btc _cost _per _transaction _percent	numerisch (11)	Werte werden nach unten (zeitlich später) kleiner
btc _cost _per _transaction	numerisch (11)	maximal 12 Stellen (Vorkommastellen + Nachkommastellen = 12)
btc _n _unique _addresses	numerisch (0)	
btc _n _transactions	numerisch (0)	
btc _n _transactions _total	numerisch (0)	Werte kumulativ (stetig steigend)
btc _n _transactions _excluding _popular	numerisch (0)	
btc _n _transactions _excluding _chains _longer _than _100	numerisch (0)	Bis zum 18.4.2010 identisch mit btc _n _transactions _excluding _popular
btc _output _volume	numerisch (7)	viele Anfangswerte sind glatte Zehner
btc _estimated _transaction _volume	numerisch (6)	Nachkommastellen erst ab 20.4.2016
btc _estimated _transaction _volume _usd	numerisch (4)	Nachkommastellen erst ab 20.4.2016

Tabelle 7.31: Data description report für bitcoinDataset

Datensatz	ethereumDataset
Observations	797
Features	18
Dateigröße	115

Feature	Datentyp	Besonderheit
Date(UTC)	Datum im Format 'mm/dd/yyyy'	
UnixTimeStamp	Datum im Format UnixTimeStamp	
eth _etherprice	numerisch (2)	bbb
eth _tx	numerisch (0)	
eth _address	numerisch (0)	Werte stetig steigend
eth _supply	numerisch (4)	Werte stetig steigend
eth _marketcap	numerisch (9)	
eth _hashrate	numerisch (4)	
eth _difficulty	numerisch (3)	die ersten drei Werte sind < 0, was der Definition der Schwierigkeit widerspricht
eth _blocks	numerisch (0)	immer vierstellig
eth _uncles	numerisch (0)	
eth _blocksize	numerisch (0)	
eth _blocktime	numerisch (2)	Nur Werte im Intervall [4,46;30,31]
eth _gasprice	numerisch (0)	
eth _gaslimit	numerisch (0)	vermutlich stufenhafter Anstieg/Abfall
eth _gasused	numerisch (0)	
eth _ethersupply	numerisch (5)	
eth _ens _register	numerisch (0)	von 30.7.2015 bis 3.5.2017 keine Daten; von 4.5.2017 bis 8.5.2017 immer Nullwerte

Tabelle 7.32: Data description report für ethereumDataset

Obwohl die Qualität der Daten erst in einem nachfolgenden Schritt genauer betrachtet wird, können nach den Data description reports schon potentielle Stolperfallen identifiziert werden. Diesen Herausforderungen muss sich angenommen werden:

- Die Formate des Datums sind unterschiedlich. Auch sind für einige Datensätze sieben Observations pro Woche verfügbar (z.B. für den Bitcoin-Kurs), für Andere nur fünf (z.B. für den Dow Jones Industrial Average). Daraus ergibt sich das Problem, wie die Daten am besten aneinander gesetzt (gejoint) werden.
- In den Daten befinden sich Lücken. Obwohl diese in der Regel im Verhältnis zum gesamten Datensatz klein sind, gibt es auch größere Lücken (z.B. die Anzahl der Registrierungen beim Ethereal Name Service pro Tag).

- Es tauchen Attribute doppelt auf. So beinhaltet sowohl der Datensatz BTC _Difficulty _Daily _Full das Schwierigkeitsmaß für das Minen, als auch der Datensatz bitcoinDataset.
- Ein besonderen Fall stellt abcnews _Date _Text dar. Hierbei handelt es sich einerseits um einen Datensatz, der zu groß ist, um komplett in einem Programm geöffnet zu werden, andererseits beinhaltet er Informationen (Überschriften) in Textform.
- Am Ende von Abschnitt 7.2.1 wurde bereits angesprochen, dass das Feature 'Change' (z.B. bei den Aktienindizes) keinen Mehrwert für die Analyse bietet.

7.2.3 Untersuchung der Daten durch Visualisierung (Explore the Data)

Der User Guide empfiehlt, um die bekannten Daten weiter zu untersuchen, „Abfrage-, Visualisierungs- und Reportingtechniken anzuwenden"(Chapman et al., 2000, S. 40; eigene Übersetzung). Es ist sicherlich aufschlussreich, alle Features auf einen Graphen mit dem Kryptowährungskurs zu plotten. Da dies jedoch sehr aufwendig ist, werden hier nur einige Daten betrachtet. Außerdem ist es wahrscheinlich, dass komplexere Zusammenhänge nicht trivial visuell erkennbar sind.

Betrachtet man das Interesse am Bitcoin (Google Websuchen und Google Newssuchen) mit dem R Statement in Listing 7.1, so zeigt sich, dass der erste Ausschlag Mitte 2011 zu erkennen ist und der Nächste erst Anfang 2013 (Abbildung 7.1).

```
1  # Read BTC datasets and convert dates
2  google_Trends_BTC_Websearch.csv <- read.csv("google_Trends_BTC_Websearch.csv");
3  google_Trends_BTC_Websearch.csv[,1] <-
       as.Date(google_Trends_BTC_Websearch.csv[,1])
4  google_Trends_BTC_Newssearch.csv <-
       read.csv("google_Trends_BTC_Newssearch.csv");
5  google_Trends_BTC_Newssearch.csv[,1] <-
       as.Date(google_Trends_BTC_Newssearch.csv[,1])
6  attach(google_Trends_BTC_Websearch.csv)
7  attach(google_Trends_BTC_Newssearch.csv)
8
9  #Plot the data in line graphs
10 ggplot(google_Trends_BTC_Websearch.csv) +
11   geom_line(data = google_Trends_BTC_Websearch.csv, aes(x = Monat, y =
         bitcoin...Weltweit., color = "BTC␣Websearch")) +
12   geom_line(data = google_Trends_BTC_Newssearch.csv, aes(x = Monat, y =
         bitcoin...Weltweit., color = "BTC␣Newssearch")) +
```

```
13    xlab('Date') +
14    ylab('GoogleSearchIndex')+
15    scale_colour_manual("",
16                        breaks = c("BTC Websearch", "BTC Newssearch"),
17                        values = c("green", "blue"))
```

Listing 7.1: Google Websuchen und Newssuchen für „Bitcoin“ im zeitlichen Verlauf in R

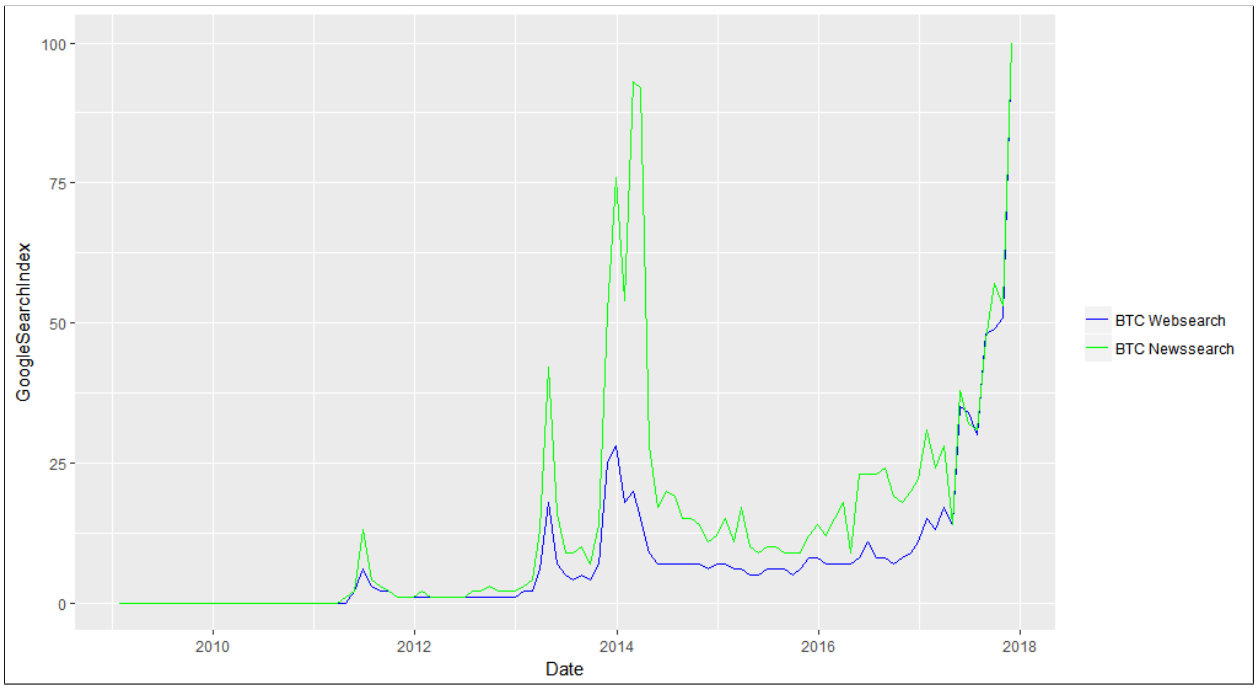

Abbildung 7.1: Google Websuchen und Newssuchen für „Bitcoin“ im zeitlichen Verlauf

Interpretiert man die Googlesuchen als Indikator für den Bekanntheitsgrad des Bitcoins, so lässt sich daraus schließen, dass die Währung bis 2011 sehr unbekannt und erst nach 2013 wirklich bekannt war. Zieht man nun den Bitcoinkurs heran (Listing 7.2), so lässt sich ein Zusammenhang erkennen (Abbildung 7.2).

```
1  #Read historical BTC price data and convert date
2  btcPrice.csv <- read.csv2("btcPrice.csv");
3  btcPrice.csv[,1] <- as.Date(btcPrice.csv[,1],"%d/%m/%Y")
4  attach(btcPrice.csv)
5
6  #Plot the data
7  ggplot(google_Trends_BTC_Websearch.csv) +
8    geom_line(data = google_Trends_BTC_Websearch.csv, aes(x = Monat, y =
         bitcoin...Weltweit., color = "BTC Websearch")) +
9    geom_line(data = google_Trends_BTC_Newssearch.csv, aes(x = Monat, y =
         bitcoin...Weltweit., color = "BTC Newssearch")) +
```

```
10  geom_line(data = btcPrice.csv, aes(x = i..Date, y = btc_market_price/50,
        color = "Bitcoinkurs")) +
11  xlab('Date') +
12  ylab('GoogleSearchIndex')+
13  scale_colour_manual("",
14                      breaks = c("BTC Websearch", "BTC Newssearch",
                            "Bitcoinkurs"),
15                      values = c("red", "green", "blue"))
```

Listing 7.2: Google Websuchen und Newssuchen für „Bitcoin“ und Bitcoinkurs im zeitlichen Verlauf in R

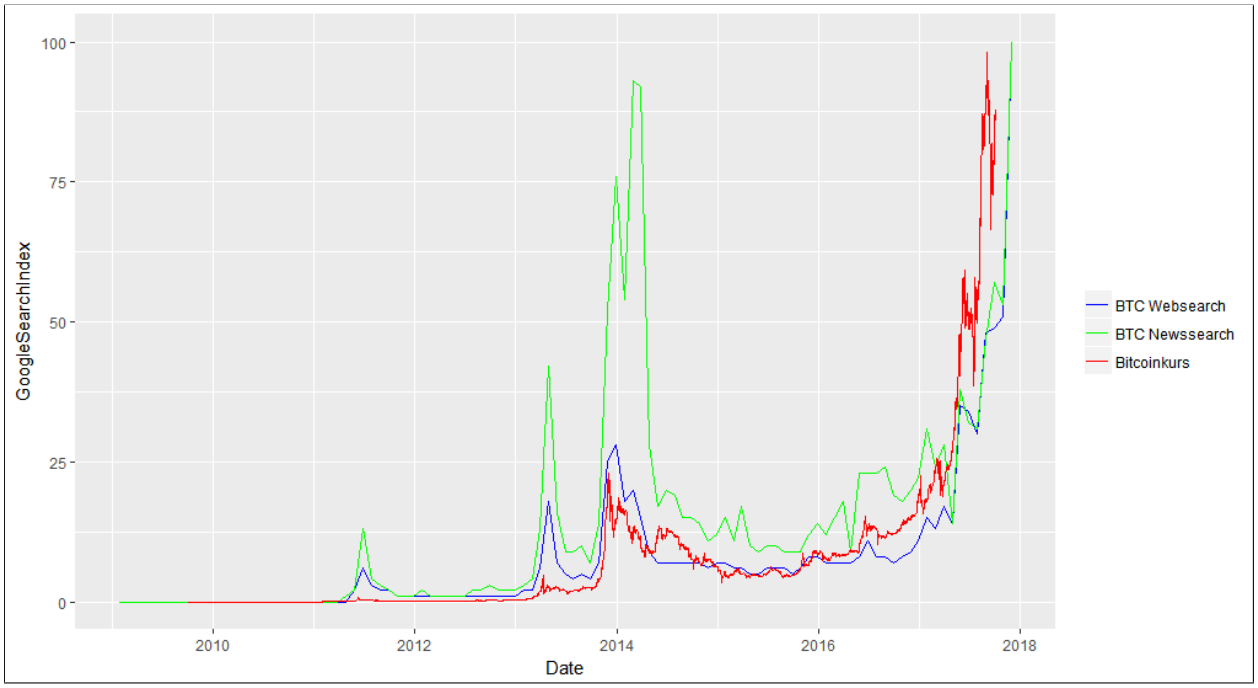

Abbildung 7.2: Google Websuchen und Newssuchen für „Bitcoin“ im zeitlichen Verlauf

(Anzumerken ist, dass die historischen Kursdaten hier in keinem Verhältnis zur y-Achse stehen und nur der grafischen Darstellung dienen.) Anders verhält es sich bei Ethereum (siehe Abbildung 7.3). Hier besteht schon vor dem offiziellen Start des Ethereumnetzwerks(Tual, 2015) ein gewisses Interesse.

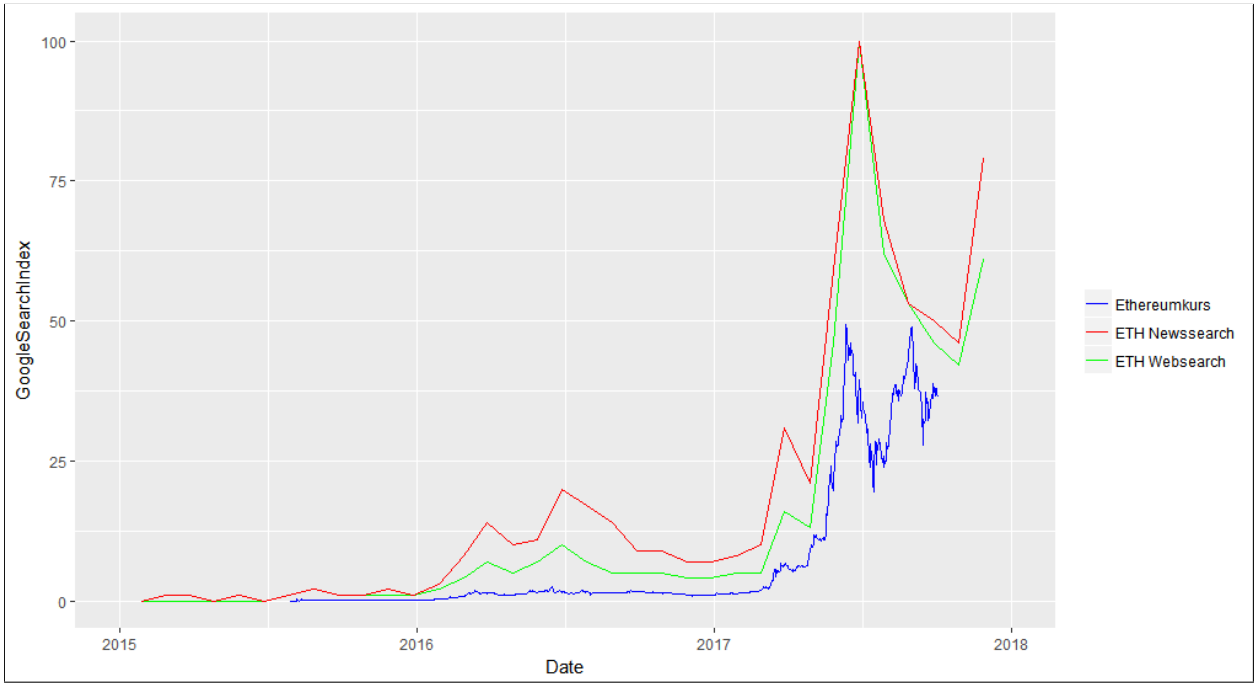

Abbildung 7.3: Google Websuchen und Newssuchen für „Ethereum“ im zeitlichen Verlauf

Als Schlussfolgerung lässt sich daraus ableiten, dass es sehr unwahrscheinlich ist, dass Indikatoren wie Aktienindizes oder der Ölpreis den Kryptowährungskurs beeinflussen, wenn sie noch unbekannt ist. Deswegen empfiehlt sich für die Analyse der Zeitraum, seit die Währungen Bekanntheit erlangt haben.

Output	Beschreibung
Data exploration report	Analyse des Bitcoinkurses erst ab 1.1.2011; des Ethereumkurses ab 30.7.2015

Tabelle 7.33: Output des Schrittes „Explore the Data“

7.2.4 Problemen mit den Daten und mögliche Lösungen (Verify Data Quality)

Bei der Sicherung der Datenqualität fällt auf, dass die Daten oberflächlich eine hohe Qualität aufweisen. Beispielsweise sind alle Datensätze in USD angegeben und strukturiert. Sie

enthalten pro Datei nicht viele Features und besitzen kaum lückenhafte Spalten. Jedoch sind einige Dinge zu beachten (siehe Tabelle 7.34).

Problem	**Lösung**
Die verschiedenen Datensätze haben unterschiedliche Datumsformate. Dies birgt Risiken beim Zusammenführen der Daten.	Vor dem Joinen der Daten müssten die Formate uniformiert werden. Dies kann beispielsweise mit der R-Methode `as.Date(data$DateColumn, format='%Y%m%d')` geschehen, solange das Datum in einem gültigen Format ist.
Beim Zusammensetzen der Datensätze entstehen Lücken, da einige Datensätze (z.B. Kryptowährungs-eigene Eigenschaften) sieben Observations pro Woche festhalten, Andere (z.B. Aktienindizes) nur fünf. Außerdem gibt es Datensätze mit Feiertagen (z.B. Neujahr) oder solche mit nur einer Observation pro Woche (z.B. STLFSI).	Azure ML Studio bietet das Experiment Item 'Clean Missing Data' an. Neben den Möglichkeiten, fehlende Daten mit dem Modus, Median oder Mittelwert auszufüllen, kann das Verfahren MICE(Azur et al., 2011) oder die Probablilistic PCA(Tipping and Bishop, 1999) genutzt werden.
Durch die unterschiedlichen Formatierungen gibt es inkonsistente Separatoren (Komma, Semikolon, Tab), Dezimaltrennzeichen (Punkt, Komma) und Tausendertrennzeichen (Punkte, ohne Trennzeichen).	Vor dem Zusammenführen müssen die Separatoren und Trennzeichen untersucht und eventuell umgeändert werden. Dies kann mit Excel ('Speichern als...' oder Notepad++ ('Find and Replace') geschehen. Nach dem Joinen muss nachgeprüft werden.
Die Schwierigkeit für das Ethereumining wird im Datensatz ethereumDataset anfangs mit kleiner als 1 angeben. Dies ist per Definition unmöglich.	Entweder es handelt sich hier um einen undokumentierte Sonderfall zu Beginn des Netzwerks oder es ist ein Fehler im Datensatz. Trotz dieser Unstimmigkeit, ist die Tatsache insignifikant und kann vernachlässigt werden.

Tabelle 7.34: Data quality report des Schrittes „Verify data quality“

7.3 Data Preperation

7.3.1 Inkludieren und Expludieren von Daten (Select Data)

Die Datensätze sind nun gut beschrieben und es wurde auf Herausforderungen und Probleme hingewiesen. Die folgenden Punkte befassen sich damit, die richtigen Daten und Features weiter zu reduzieren („Select Data“), sie zu reinigen („Clean Data“) und schließlich zum endgültigen Analyse-Datensatz zusammenfassen („Construct Data“, „Integrate Data“ und „Format Data“).

Ausgenommen der jetzt begründet ausgeschlossenen Daten, werden alle Vorgestellten zum bilden des Model genutzt. Tabelle 7.36 liefert einen Überblick.

In Punkt 7.2.1 wurde festgestellt, dass der Datensatz 'BTC _Total _Volume _Daily _Full' große Lücken enthält. Darüber hinaus enthält der Datensatz 'bitcoinDataset' ebenfalls die Anzahl der Bitcoins als Feature 'btc _total _bitcoins'. Ein stichprobenhafter Konsistenzcheck ergibt, dass die vorhanden Daten sich decken. Aus diesem Grund wird der Lückenhafte Datensatz exkludiert. 'BTC _Difficulty _Daily _Full' enthält wie 'bitcoinDataset' die Miningschwierigkeit (siehe Punkt 5). Obwohl der einzelne Datensatz minimal genauere Werte enthält, fällt die Wahl erneut auf das 'bitcoinDataset', da es Daten über eine längere Zeitspanne enthält. Genauso verhält sich bei 'BTC _Transaction _Number _Fully _Daily' und 'BTC _Price _Multiple _Daily'. Analog wird bei den Ethereumdatensätzen vorgegangen. 'ETH _Total _Volume _Daily _Full' enthält die gleichen Daten wie 'ethereumDataset'. Allerdings um Faktor 100000 erhöht. Vermutlich handelt es sich hierbei um einen Konvertierungs- oder Kopierfehler, da das derzeit mögliche Maximum bei 100 Millionen Ether liegt.(Buterin, 2016; Hawksby-Robinson, 2017) Die Werte für die Schwierigkeit und Anzahl der Transaktionen stimmen bei 'ETH _Difficuly _Daily _Full' bzw. 'ETH _Transaction _Number _Fully' mit 'ethereumDataset' genau überein. Der Bitcoinkurs soll vom 1.1.2011 an analysiert werden. Für einige Aktienindizes liegen für diese Zeit noch keine Daten vor. Um die über 50 Indizes auszudünnen, werden nur solche für das Machine Learning genutzt, für die Daten vorhanden sind. Dadurch wird versucht, Datenlücken zu vermeiden. Damit der Arbeitsaufwand reduziert wird, wird die Auswahl für die Analyse des Ethereumpreises übernommen. Die Google Trends Daten werden beibehalten. Anders verhält es sich bei den Wikipedia-Seitenaufrufen. Für die Bitcoinanalyse ist die Zeitspanne des Datensatzes zu kurz (es fehlen 4 1/2 Jahre: von 1.1.2011 bis 1.7.2015). Eine mögliche Korrelation (siehe Abbildung 7.4) zwischen Seitenaufrufen und Bitcoinkurs müsste gesondert untersucht werden.

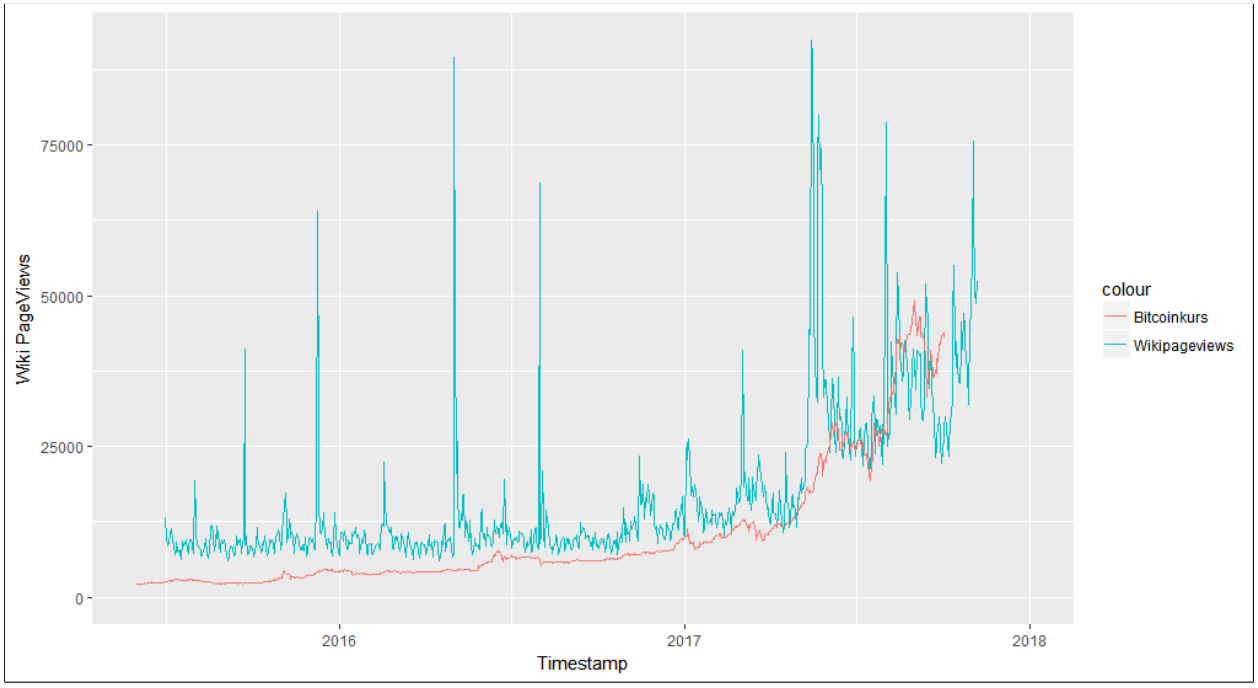

Abbildung 7.4: Wikipedia Seitenaufrufe „Bitcoin“ und der Bitcoinkurs im zeitlichen Verlauf

Für die Ethereumkursanalyse werden die Seitenaufrufe herangezogen, da sich die Daten zeitlich decken (siehe Abbildung 7.5). (Erneut ist bei beiden Diagrammen der Kryptowährungskurs nur relativ dargestellt und nicht als absolute Größe.)

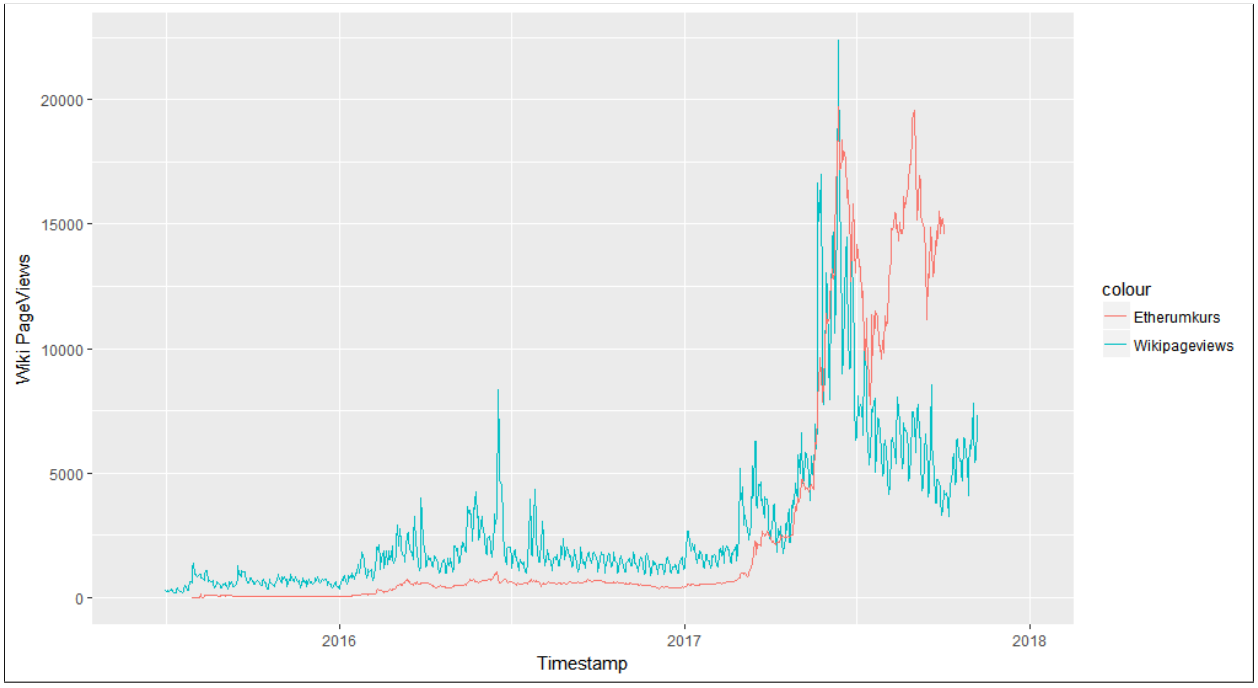

Abbildung 7.5: Wikipedia Seitenaufrufe „Ethereum“ und der Ethereumkurs im zeitlichen Verlauf

Obwohl für den STLFSI nur wöchentlich - nicht täglich - berechnet wird, wird er beibehalten. Bei den Währungen wird nur der Kurs INR/USD exkludiert, da hier nur Daten ab dem 9.9.2014 vorliegen. Alle Daten zu natürlichen Ressourcen (Gold, Silber, Öl) werden inkludiert. Dem Datensatz 'ETH _Price _Volume _Full _Daily' fehlt die erste Woche an Daten, ist sonst aber reicher an Informationen über den Ethereum/USD-Kurs als das 'ethereumDataset', da es nicht nur den Durchschnittspreis eines Tages enthält, sondern sowohl den ersten und letzten Kurs in einem 24-Stunden-Fenster, als auch den Höchsten und den Niedrigsten. Dem 'BTC _Price _Volume _Full _Daily' hingegen fehlt das erste halbe Jahr an Daten und weist erst ab Ende (18.12.) 2011 keine Lücken mehr auf. In allen Datensätzen (Aktienindizes, Währungen, natürliche Ressourcen) wird das Feature '%Change' gestrichen, da in 7.2.2 festgestellt wurde, dass es redundant ist. abcnews _Date _Text wird beibehalten. Auch wenn die Datensätze an dieser Stelle aussortiert werden, war ihre Beschreibung keine Verschwendung. Sie hat dazu beigetragen, das Domainwissen zu vertiefen und die Plausibilität der Daten zu prüfen.

Datensatz	**Inkludiert**	**Exkludiert**
BTC _Total _Volume _Daily _Full		X
BTC _Difficulty _Daily _Full		X

BTC _Transaction _Number _Fully _Daily		X
BTC _Price _Multiple _Daily		X
ETH _Total _Volume _Daily _Full		X
ETH _Difficuly _Daily _Full		X
ETH _Transaction _Number _Fully _Daily		X
google _Trends _BTC _Websearch	X	
google _Trends _ETH _Websearch	X	
google _Trends _BTC _Newssearch	X	
google _Trends _ETH _Newssearch	X	
Wiki _Page _Views _BTC		X
Wiki _Page _Views _ETH	X	
abcnews _Date _Text	X	
FTXIN9, PSI20, AEX, BFX, XU100, BVSP, VIX, CSE, GDAXI, DJI, FTSE, FTMIB, HSI, IBEX, MXX, JKSE, KSE, KS11, MCX, IXIC, NSEI, N225, OMXC20, OMXS30, IRTS, SPX, AXJO, GSPTSE, SSEC, SSMI, TA35, TASI, TRX50CAP, US2000, WIG20	X	
ATX, BSESN, BUX, NZDOW, DJSH, STOXX50E, STI, HNX30, PSI, SETI, SZSC1, TWII		X
STLFSI	X	
CNY _USD _history	X	
JPY _USD _history	X	
EUR _USD _history	X	
GBP _USD _history	X	
INR _USD _history		X
BRL _USD _history	X	
gold _history	X	
silver _history	X	
oil _brent _history	X	
oil _wti _history	X	
ETH _Price _Volume _Full _Daily	X	
BTC _Price _Volume _Full _Daily		X
bitcoinDataset	X	
ethereumDataset	X	

Tabelle 7.35: Inkludierte und exkludierte Datensätze für die Analyse

7.3.2 Aufbereitung der Daten (Clean, Construct, Integrate, Format Data)

Die Schritte 'Clean Data', 'Construct Data', 'Integrate Data' und 'Format Data' gehen Hand in Hand. Deswegen werden die Punkte zusammengefasst. Bei der tatsächlichen Durchführung wurden die Schritte iterativ und inkrementell durchlaufen.
Die Menge aller Daten für die Untersuchung ist heterogen. Es existieren jedoch Blöcke (z.B. alle Währungskurse oder alle natürlichen Ressourcen), die innerhalb eine homogene Struktur aufweisen. Aus diesem Grund ist es ratsam, zuerst die homogenen Daten zusammenzufassen und diese großen Blöcke dann zusammenzufügen. Durch dieses Buttom-Up-Zusammensetzen wird die Integration vereinfacht. Das Vorgehen ist dabei folgendes:

- Die einzelnen Datensätze werden so bearbeitet, dass sie für die Zusammenführung in ihren Block bereit sind.
- Die bearbeiteten Daten werden ihrem Datenblock hinzugefügt. Das Ergebnis wird untersucht und eventuelle Unreinheiten beseitigt.
- Alle Blöcke werden in einen abschließenden Datensatz integriert. Dieser wird wiederum inspiziert und bereinigt.

Nachfolgend wird das Bereinigen und Zusammensetzten beschrieben. Zuerst werden die Aktienindizes bearbeitet (Listing 7.4) und gejoint (Abbildung 7.6).

```
#clear all environment data
rm(list=ls())

#use anytime for date conversions
library(anytime)

#Function to format the data and rename columns
processDataSet <- function(fileName){

  #Read Data as csv
  dataSet <- read.csv2(fileName)

  #Name for columns
  name <- basename(fileName)
  dateName <- paste(name, "Date")
  priceName <- paste(name, "Price")
  openName <- paste(name, "Open")
  highName <- paste(name, "High")
  lowName <- paste(name, "Low")
  volumeName <- paste(name, "Volume")
  colNames <- c(dateName, priceName, openName, highName, lowName, volumeName)

```

```
23    #attach and rename columns
24    colnames(dataSet)[c(1, 2, 3, 4, 5, 6)] <- colNames
25
26    #select what to keep --> drop column Change
27    keep <- colNames
28    dataSet <- dataSet[,keep]
29
30    #remove all thousand-seperator-commas
31    dataSet[,priceName] <- gsub(",", "", dataSet[,priceName])
32    dataSet[,openName] <- gsub(",", "", dataSet[,openName])
33    dataSet[,highName] <- gsub(",", "", dataSet[,highName])
34    dataSet[,lowName] <- gsub(",", "", dataSet[,lowName])
35
36    #change type Factor to correct one
37    dataSet[,dateName] <- anytime(dataSet[,dateName])
38    dataSet[,priceName] <- as.numeric(as.character(dataSet[,priceName]))
39    dataSet[,openName] <- as.numeric(as.character(dataSet[,openName]))
40    dataSet[,highName] <- as.numeric(as.character(dataSet[,highName]))
41    dataSet[,lowName] <- as.numeric(as.character(dataSet[,lowName]))
42    dataSet[,volumeName] <- pseudoNumeric.as.numeric(dataSet[,volumeName])
43
44    #missing values to NA (missing Data)
45    dataSet[, volumeName][dataSet[, volumeName] == "-"] <- NA
46
47    #save as csv; not csv2, because azure ML studio can't read semicolons as
          seperators
48    write.csv(dataSet, file = paste(name, "_processed.csv"), append = FALSE,
          quote = TRUE, sep = ",",
49                eol = "\n", na = "", dec = ".", row.names = TRUE,
50                col.names = TRUE, qmethod = c("escape", "double"),
51                fileEncoding = "")
52  }
53
54  #get all files to convert
55  files <- list.files(path=".", pattern="*.csv", full.names=T, recursive=FALSE)
56
57  #convert
58  lapply(files, processDataSet)
```

Listing 7.4: Aufbereiten der Indizes-Datensätze

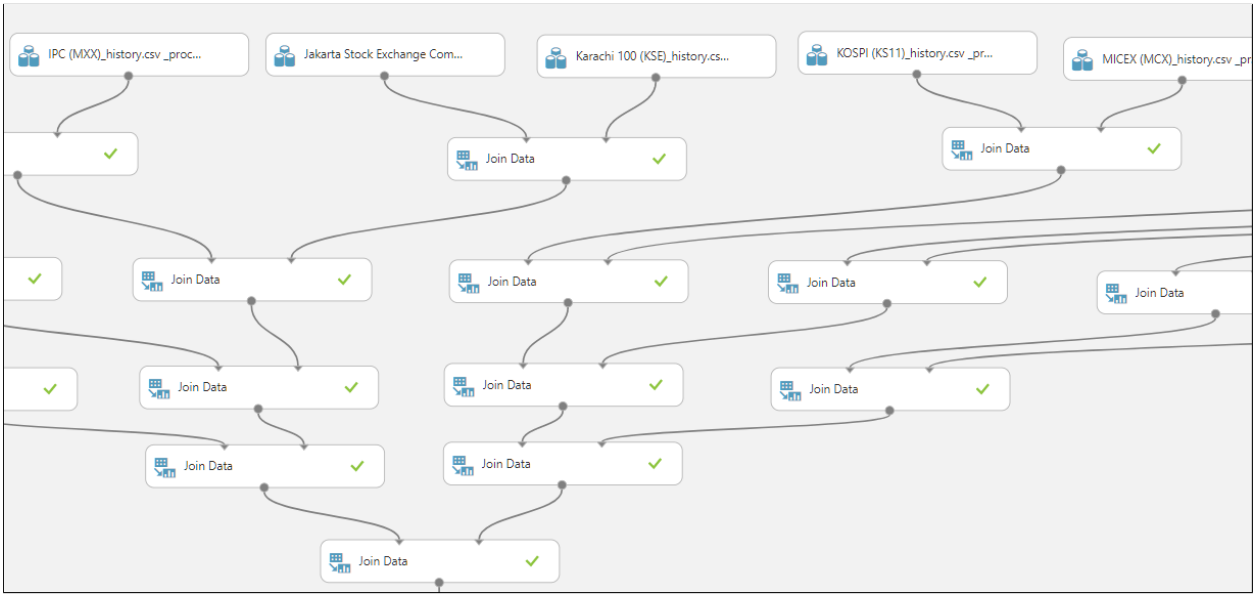

Abbildung 7.6: Zusammenführen der Aktienindizes in Azure Machine Learning Studio

Daraufhin werden sie nachbearbeitet (die Beschreibung bezieht sich auf Abbildung 7.7): In jedem Ursprungsdatensatz existiert eine Spalte, die die Zeilen durchnummeriert. Dies diente zum beheben von Fehlern und wird nun entfernt (1. Select Columns in Dataset). Ebenso enthält das Ergebnis noch 35 Datums-Spalten (von jedem Ursprungsdatensatz eine). Diese werden bis auf Eine entfernt (2. Select Columns in Dataset und Edit Metadata).

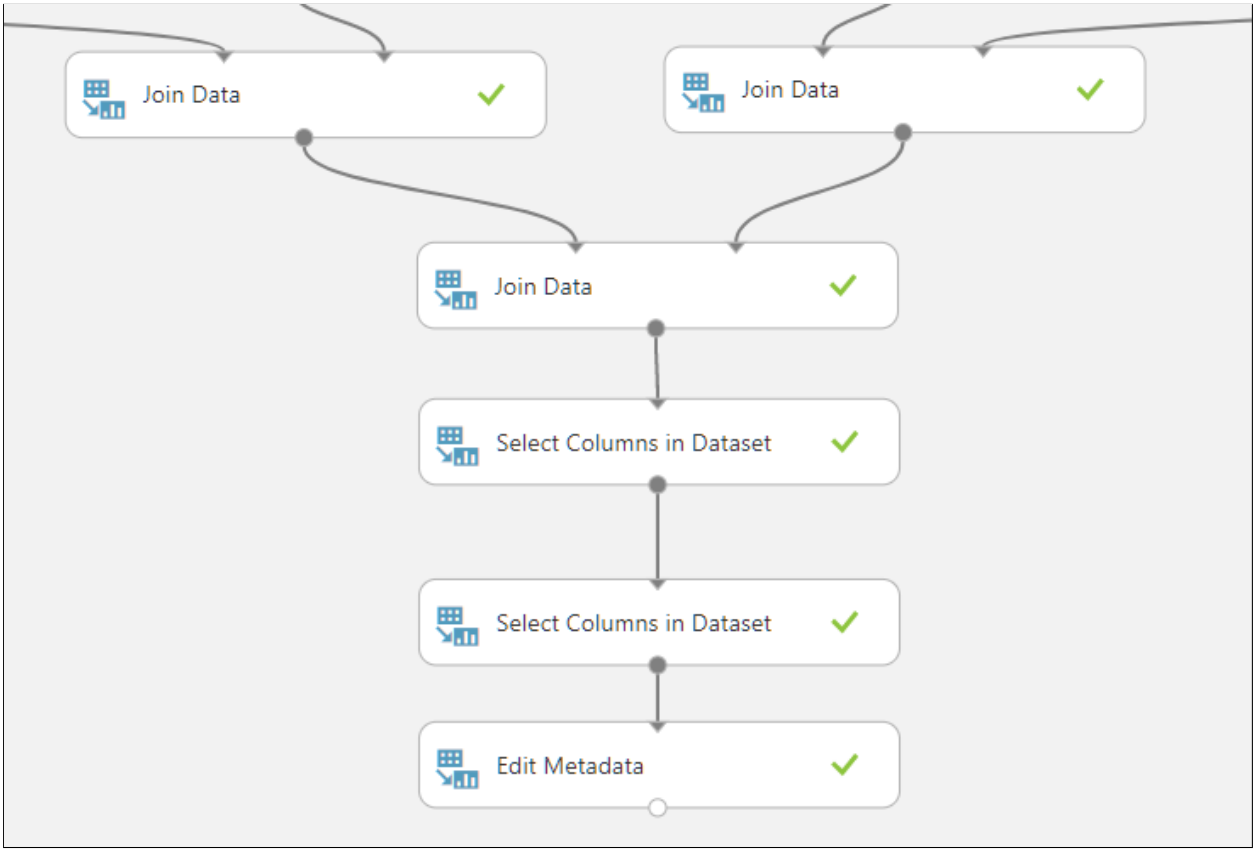

Abbildung 7.7: Nachbearbeiten des Ergebnisdatensatzes in Azure Machine Learning Studio

Analog werden die natürlichen Ressourcen zu einem Datenblock zusammengefasst. Ähnlich wird mit den Währungskursen vorgegangen. Die Verarbeitung in R ist jedoch einfacher, da sie keine Tausendertrennzeichen, keine pseudo-numerischen Werte und keine Lücken enthalten. Spezieller ist der Datensatz 'abcnews _Date _Text' (siehe Abbildung 7.8). Er wird mit dem Experiment Item 'Preprocess Text' vorbearbeitet, nachdem alte Überschriften (vor dem 1.1.2011) entfernt wurden (Execute R Script; siehe Listing 7.5 Zeile 6).

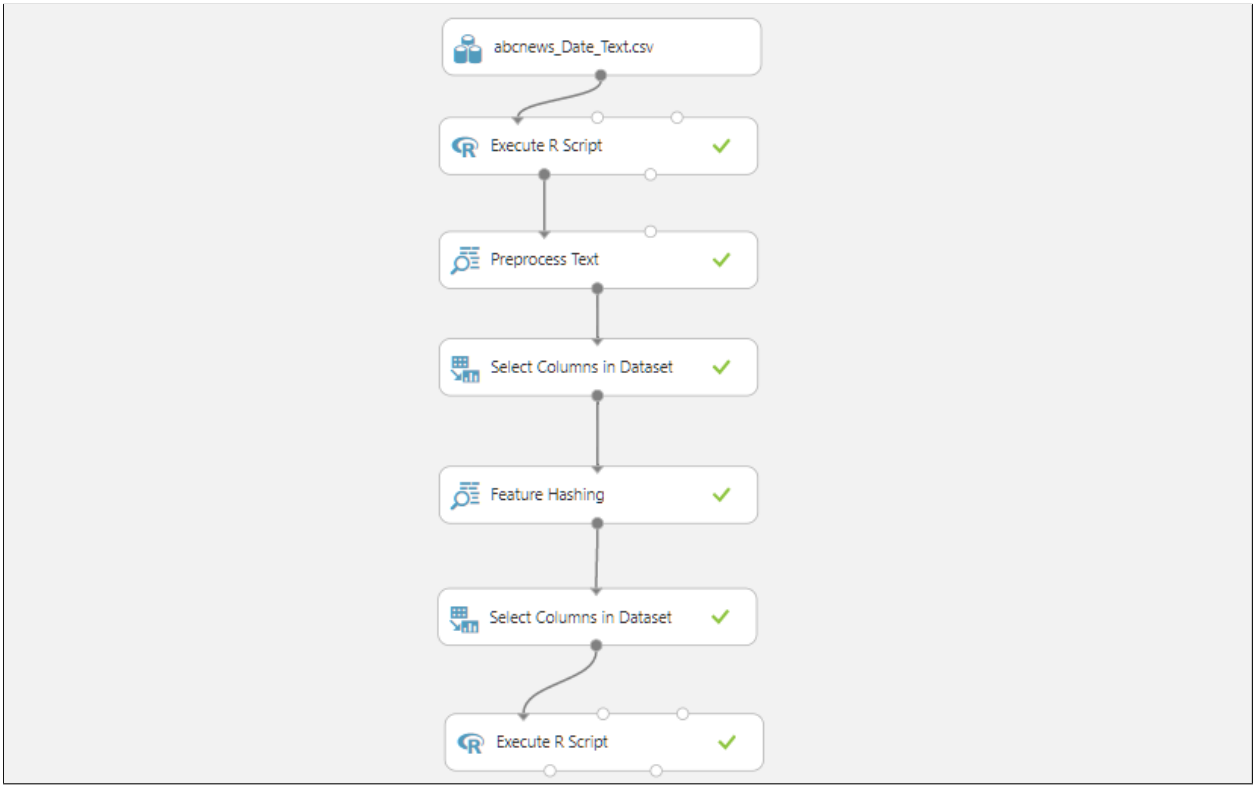

Abbildung 7.8: Feature Hashing der abcnews in Azure Machine Learning Studio

```
# Map 1-based optional input ports to variables
abcnews <- maml.mapInputPort(1) # class: data.frame
attach(abcnews)

#remove old news
abcnews <- abcnews[abcnews$publish_date > 20110101,]

#convert text to date
abcnews$publish_date <- as.Date(as.character(abcnews$publish_date), "%Y%m%d")

# Select data.frame to be sent to the output Dataset port
maml.mapOutputPort("abcnews");
```

Listing 7.5: Entfernen der alten Überschriften

Das Preprocessing führt unter anderem eine Lemmatisierung durch und entfernt Zahlen und Sonderzeichen. Anschließend wird ein Feature Hashing durchgeführt. Dabei handelt es sich um die Überführung von Texttupeln in Zahlwerte, damit eine Verarbeitung durch Computer möglich wird. Für jede Überschrift existiert eine einzelne Zeile im Datensatz. Diese werden mit dem R-Script in Listing 7.6 in Azure aggregiert.

```
# Map 1-based optional input ports to variables
```

```
2 abcnews <- maml.mapInputPort(1) # class: data.frame
3 attach(abcnews)
4
5 #aggregate by date
6 abccompact <- aggregate(. ~ publish_date, abcnews, sum)
7
8 # Select data.frame to be sent to the output Dataset port
9 maml.mapOutputPort("abccompact");
```

Listing 7.6: Aggregieren der Hashing-Ergebnisse

Zuletzt werden noch die Google News- und Websuchen zusammengefasst. Der abschließende Data cleaning report ist in Tabelle 7.36 zu finden.

Problem	**Lösung**	**Werkzeug**
unterschiedliche Datumsformate	Zuerst werden die Daten in Text und anschließend mit dem Packet 'anytime' in R konvertiert.	R + Package 'anytime' (siehe Listing 7.4 Zeile 37)
Lücken durch joinen der Daten	Es handelt sich um 'Missing not at random'-Fehler (MNAR)(Graham, 2009, S. 553). Das bedeutet, dass die Lücken absichtlich sind (weil an diesen Tagen nicht gehandelt wird). Da es kein statistisches Verfahren zum schließen solcher Lücken gibt, werden sie beibehalten. (Leonhart, 2014, S. 1109)	
unterschiedliche Separatoren, Trennzeichen etc.	Die Daten werden mit Excel von '.tsv' in '.csv' konvertiert. Eine programmatische Änderung oder 'Find and Replace' ist schwer, da es einen Tab als Trennzeichen zwischen Tag, Monat und Jahr gibt. Kommas als Tausendertrennzeichen werden entfernt.	Excel 'Save As...' und R-Methode 'gsub' (siehe Listing 7.4 Zeilen 30-34)
pseudo-numerische Werte (z.B. „1.5M“ oder „0.07k“)	Bei betroffenen Werten wird der Buchstabe entfernt und die Zahl mit dem entsprechenden Wert multipliziert (z.B. 1.3×1000 für 1.3K)	R-Methoden 'gsup' und 'grepl', sowie mathematische Operationen (siehe Listing 7.7)

Fehlende Werte	Fehlende Werte im Volumen werden als '-' angegeben, nicht als 'Nichts'. Durch ersetzen des '-' in R wird das Problem behoben.	R: Listing 7.4 Zeile 45

Tabelle 7.36: Data cleaning report

```
1  #clear environment
2  rm(list = ls())
3
4  #converts a vector of pseudo numerics to a vector of numerics
5  pseudoNumeric.as.numeric <- function(initialVector) {
6
7    #the new vector to fill with values
8    newVector <- c()
9
10   #convert from factors to characters
11   initialVector <- as.character(initialVector)
12
13   #convert each value
14   for(x in initialVector){
15
16     #see which case
17     #(1) k or K for 1000
18     #(2) m or M for 1000000
19     #(3) b ir B for 1000000000
20     if (grepl("k", x) || grepl("K", x)) {
21
22       #remove character
23       x <- gsub("K", "", x)
24       x <- gsub("k", "", x)
25
26       #convert to numeric
27       x <- as.numeric(x)
28
29       #multiply to ge correct number
30       x <- x * 1000
31     }
32     if (grepl("m", x) || grepl("M", x)) {
33       #remove character
34       x <- gsub("m", "", x)
35       x <- gsub("M", "", x)
36
37       #convert to numeric
38       x <- as.numeric(x)
39
40       #multiply to ge correct number
41       x <- x * 1000000
42
43     }
44     if (grepl("b", x) || grepl("B", x)) {
```

```
45        #remove character
46        x <- gsub("b", "", x)
47        x <- gsub("B", "", x)
48
49        #convert to numeric
50        x <- as.numeric(x)
51
52        #multiply to ge correct number
53        x <- x * 1000000000
54      }
55
56      #add the newly converted value to the return vector
57      newVector <- c(newVector, as.character(x))
58    }
59
60    #return when finished
61    return(newVector)
62  }
```

Listing 7.7: Konvertierung von pseudo-numerischen Werten

Der Schritt 'Construct data' enthält die Möglichkeit, neue Features hinzuzufügen. Hier ist kein Bedarf dafür. Außerdem handelt es sich teilweise schon um stark aggregierte Features, wie den STLFSI (siehe 7.1.2). Damit sind die Konvertierungen, Formatierungen und Zusammenführungen der Datenblöcke abgeschlossen. Die Ergebnisse sind in den Tabellen 7.37 und 7.38 festgehalten.

Output	**Beschreibung**
Hashed _news	Die abcsnews nach dem Feature Hashing und der Aggregation.

Tabelle 7.37: Output: Generated records

Output	**Beschreibung**
Joined _Indices	Alle Aktienindizes in einem Datensatz.
Joined _Currencies	Alle Währungen in einem Datensatz.
Joined _Resources	Alle natürlichen Ressourcen in einem Datensatz.
Joined _Public	Google Suchen und Google Newssuchen in einem Datensatz.

Tabelle 7.38: Output: Merged data

Syntaktische Änderungen an den Datensätzen, also verändern der Spaltenreihenfolgen oder der Spaltenbezeichnungen sind Teil des Schrittes 'Format data'. Es kann beispielsweise in einem eingesetzten Werkzeug notwendig sein, dass die erste Spalte einer Tabelle ein eindeutiger Schlüssel sein muss.(Chapman et al., 2000, S. 46) Obgleich dieser Fall hier nicht eintritt, werden syntaktische Änderungen vorgenommen: Wie oben bereits erwähnt, entstehen beim Zusammenführen der Tabellen doppelte Spalten. Werden zwei Tabellen gejoint, so existieren zwei Datumsspalten, die die gleichen Werte beinhalten (rechte Datumsspalte und linke Datumsspalte). Diese Redundanzen werden entfernt. Außerdem wird die beibehaltene Spalte in 'Date' umbenannt.
Alle erzeugten Blöcke und einzelnen Datensätze werden nun im Azure Machine Learning Studio zu einem Datensatz zusammengefasst. Für das Machine Learning werden aus diesem die benötigten Spalten für BTC bzw. ETH ausgewählt. Da das Joinen wie bisher verläuft, wird auf eine extra Beschreibung von Abbildung 7.9 verzichtet.

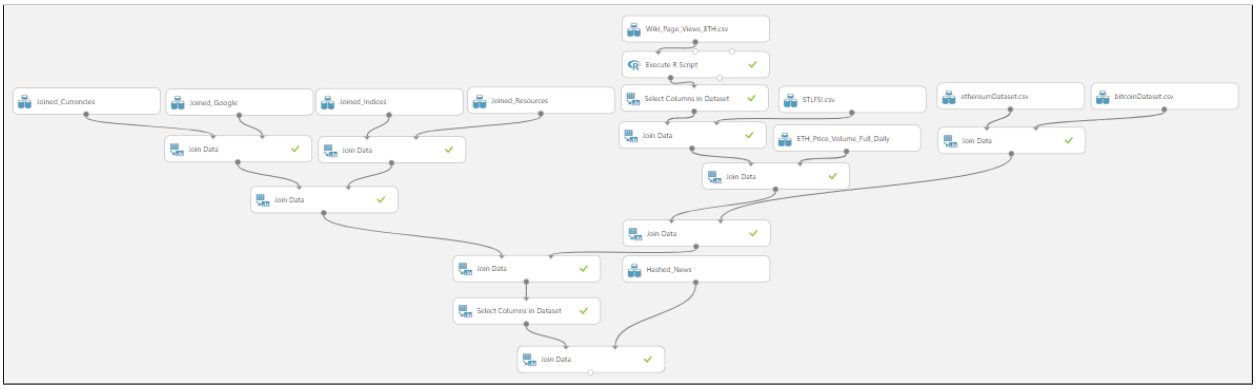

Abbildung 7.9: Erzeugen des finalen Datensatzes für das Machine Learning

Der letzte Schritt vor dem Machine Learning ist das Auswählen des Analysezeitraums und das bereinigen fehlender Daten (siebe Abbildung 7.10).

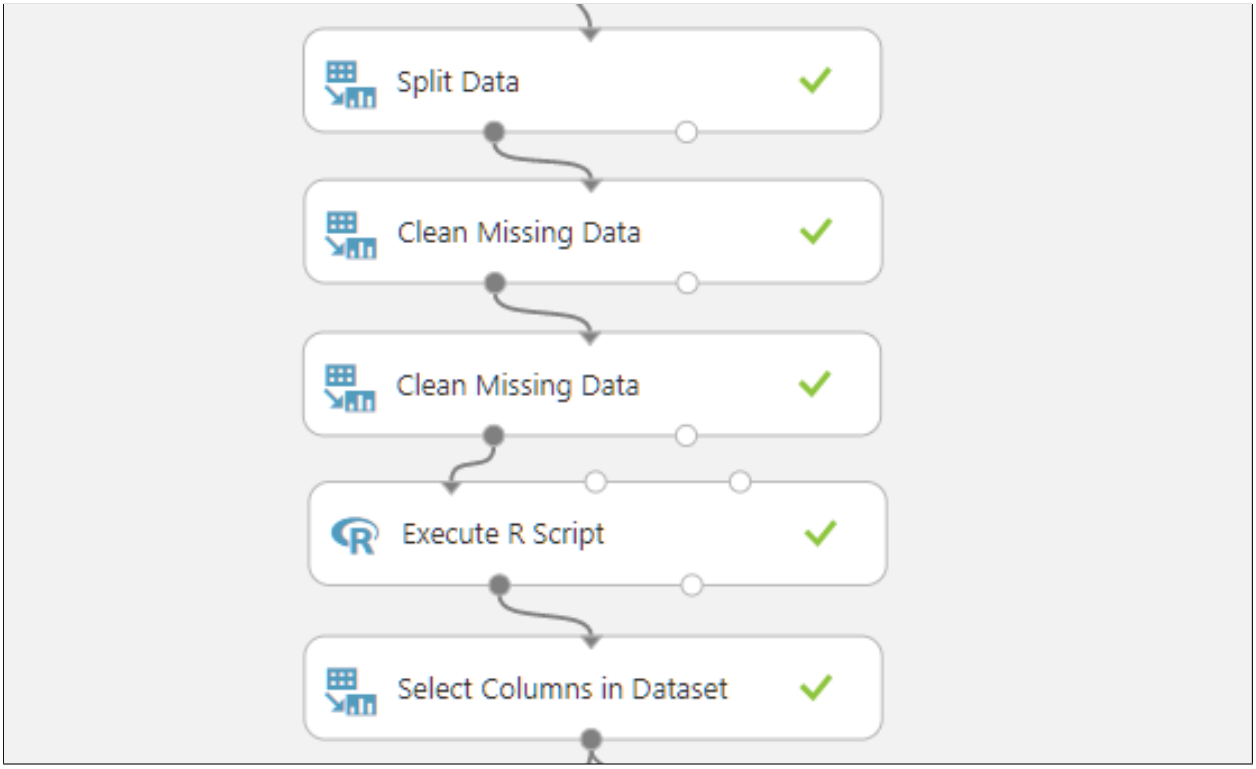

Abbildung 7.10: Auswahl des Analysezeitraums und bereinigen fehlender Daten in Azure Machine Learning Studio

Um nur die Zeilen zur Analyse zu benutzen, die im ausgewählten Zeitintervall (siehe Punkt 7.2.3) liegen, wird das Datenset noch aufgesplittet (Experiment Item 'Split Data' mit Relative Expression \„Date“ >= 2011-01-01T00:00:00 für BTC und \„Date“ >= 2015-07-30T00:00:00 für ETH). Daraufhin werden mit dem Item 'Clean Missing Data (Probabilistic PCA))' fehlende Daten ersetzt, die nicht unter die Kategorie MNAR fallen. Bei neun Spalten ist dies nicht möglich (keine Fehlermeldung in Azure Machine Learning; vermutlich zu wenige Werte für Probabilistic PCA). Diese werden mit einem erneuten 'Clean Missing Data (Remove entire column)' entfernt. Als Abschluss wird mit dem R-Script in Listing 7.8 eine neue Spalte angefügt, die angibt, ob der Kurs zum Vortag gestiegen ist (Wert: 1) oder nicht (Wert: 0). Dieser wird im Nachfolgenden als Vorhersagewert genutzt.

```
1 # Map 1-based optional input ports to variables
2 data <- maml.mapInputPort(1) # class: data.frame
3
4 #add column increase and default to NA
5 data[, "increase"] <- NA
6
7 #get the column with the price
8 priceVector <- data[, "btc_market_price"]
```

```
9
10 #returns 0 if there was no increase in value; else returns 1
11 checkForIncrease <- function(prev, cur) {
12   #skip na values
13   if ((is.na(prev)) || (is.na(cur))) {
14     return(0)
15   } else {
16     #set to 1
17     increase <- 1
18
19     #if there was no increase --> set to zero
20     if (prev >= cur) {
21       increase <- 0
22     }
23
24     #return increase
25     return(increase)
26   }
27 }
28
29 #for each value
30 for (i in 1:length(priceVector)) {
31   #get values
32   prev <- data[i, "btc_market_price"]
33   cure <- data[i + 1, "btc_market_price"]
34
35   #fill values
36   data[i + 1, "increase"] <-
37     checkForIncrease(prev, cure)
38 }
39
40 # Select data.frame to be sent to the output Dataset port
41 maml.mapOutputPort("data")
```

Listing 7.8: Hinzufügen und Popularisieren der Spalte „increase“ in Azure Machine Learning Studio

7.4 Modeling

7.4.1 Auswahl von Klassifizierungs- und Regressionsalgorithmen (Select the Modeling Technique)

Wie in Punkt 7.1.3 gefordert, wird einerseits versucht, einen Wert für den Kurs vorherzusagen (mit Hilfe von Regressionen) und andererseits, ob der Kurs steigt oder nicht (Two-Class classification). In beiden Fällen handelt es sich um Supervised Learning, da für jeden Input

ein Output vorhanden ist (siehe Punkt 4.1). In Azure verfügbare Two-Class classification Algorithmen sind:

- Two-Class Support Vector Machine
- Two-Class Neural Network
- Two-Class Logistic Regression
- Two-Class Locally-Deep Support Vector Machine
- Two-Class Decision Jungle
- Two-Class Decision Forest
- Two-Class Boosted Decision Tree
- Two-Class Bayes Point Machine
- Two-Class Averaged Perceptron

Bei den Regressionen sind es:

- Bayesian Linear Regression
- Boosted Decision Tree Regression
- Decision Forest Regression
- Fast Forest Quantile Regression (*)
- Linear Regression (*)
- Neural Network Regression
- Ordinal Regression (*)
- Poisson Regression (*)

Ausgeschlossen (mit * gekennzeichnet) werden Ordinal Regression (nur geeignet, um Rangfolgen aufzustellen), Poisson Regression (nur geigten, um vorherzusagen, wie oft etwas passiert), Linear Regression (nur für sehr simple Modelle) und Ordinal Regression (zwar geeignet, nutzt aber andere Metriken und ist nicht vergleichbar). Die Algorithmen besitzen konfigurierbare Parameter. Die Güte der erzeugten Modelle hängt unter anderem von diesen einstellbaren Werten ab. Azure bietet die Möglichkeit, die Parameter automatisch zu optimieren. Dazu wird das Experiment Item 'Tune Model Hyperparameter' eingesetzt. Es verbessert das Ergebnis des Models hinsichtlich eines bestimmten Wertes (Genauigkeit, Präzision etc.). Im vorliegenden Fall werden als Zielparameter die Metriken aus den Data Mining Goals (Punkt 7.1.3) gewählt: F1-Score für Klassifikationen und Bestimmtheitsmaß (Coefficient of determination) für die Regressionen.

7.4.2 Festlegung des Verhältnis von Trainings- und Testdaten (Generate Test Design)

Es muss überlegt werden, in welchem Verhältnis die zu analysierenden Daten aufgeteilt werden (Trainingsdaten : Testdaten). Es gibt kein allgemeingültiges, optimales Verhältnis. In der Praxis sind jedoch Verhältnisse wie 70:30 oder 80:20 üblich. Es existieren auch andere Ansätze, die wiederum auf Algorithmen basieren.(Crowther and Cox, 2005) Beachtet werden muss jedoch, dass nicht einfach obere und untere Hälfte des Datensatzes genutzt werden, sondern die Aufteilung nach dem Zufallsprinzip geschieht. Da das Experiment Item 'Tune Model Hyperparameter' eingesetzt wird, gibt es die Möglichkeit, ein zusätzliches Validation Dataset zu nutzen. Das für diese Analyse gewählte Verhältnis (Training 56 : Validierung 14 : Test 30) ist in Abbildung 7.11 visualisiert.

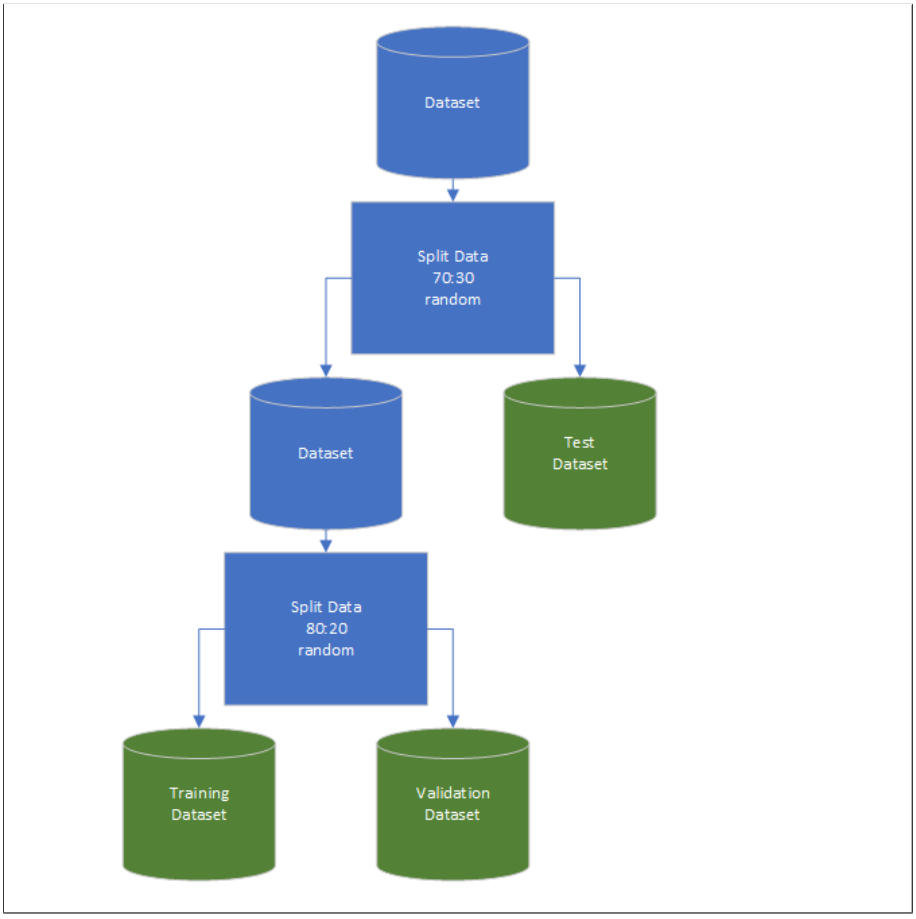

Abbildung 7.11: Verhältnis der Aufteilen in Trainings-, Test- und Validierungsdatensatz (Output Testdesign)

7.4.3 Durchführung des Machine Learning (Build the Model)

Nachdem alle Vorbereitungen abgeschlossen sind, wird jetzt das eigentliche Machine Learning durchgeführt. Dazu werden vier Experimente in Azure Machine Learning Studio modelliert. Je eines für:

- die Kurspreisvorhersage für ETH (Regression für ETH),
- die Kurspreisvorhersage für BTC (Regression für BTC),
- die Vorhersage, ob der ETH Kurs steigt (Two-class classification für ETH) und
- die Vorhersage, ob der BTC Kurs steigt (Two-class classification für BTC).

Die Experimente gleichen sich sehr, deswegen sind hier nur die Kurspreisvorhersage für ETH (Abbildung 7.12) und die Klassifikation für BTC (Abbildung 7.13) abgebildet.

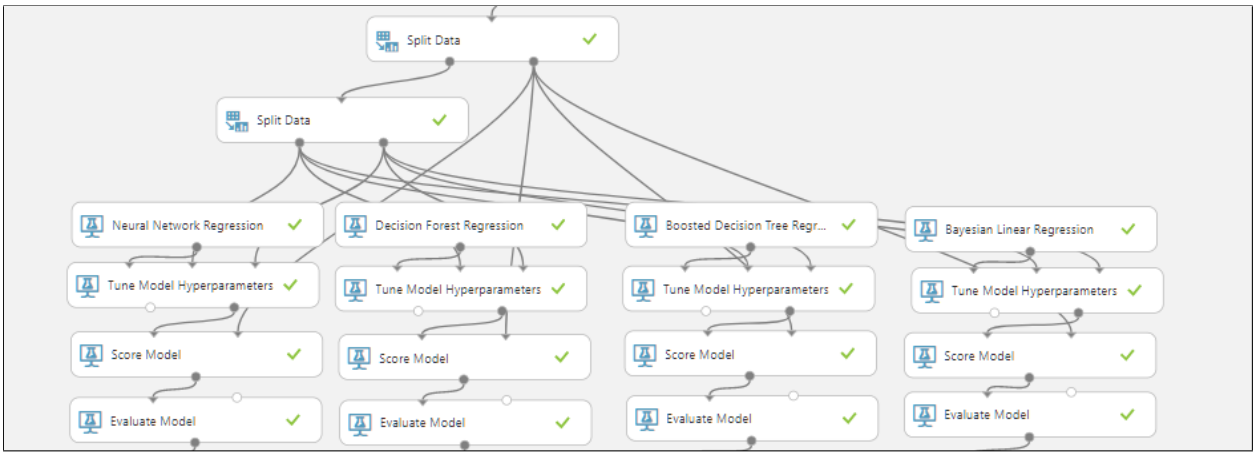

Abbildung 7.12: Ethereum Regressionen in Azure Machine Learning Studio

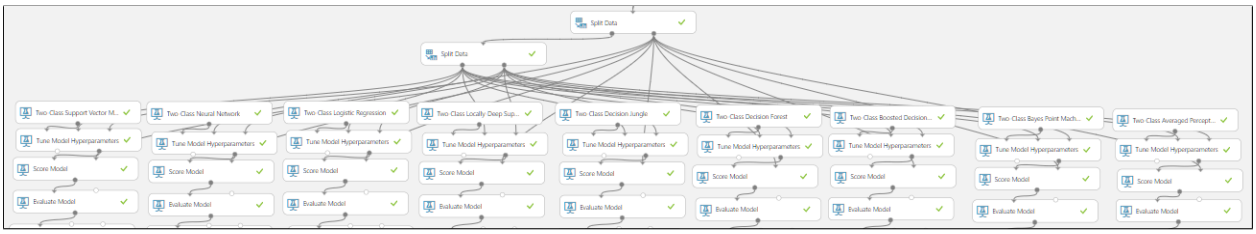

Abbildung 7.13: Bitcoin Klassifikation in Azure Machine Learning Studio

Mit den zwei 'Split Data' Experiment Items wird der Analysedatensatz, wie im vorherigen Punkt spezifiziert, in Trainings-, Validierungs- und Testdaten aufgeteilt. Zu sehen ist auch das Item 'Tune Model Hyperparameter (entire grid)', das sowohl das Training des Models, als auch das optimieren der Algorithmenparameter übernimmt (deshalb an dieser Stelle kein Output „Parameter Settings“ wie im Referenzmodell verlangt). Die Option 'entire grid' beim Verbessern der Parameter ist eine inperformante Option, die aber sehr genau arbeitet. Dies ist verkraftbar, da die Gesamtlaufzeit des Experiments 30 Minuten nicht deutlich überschreitet.

7.4.4 Betrachtung der Ergebnisse (Assess the Model)

Ein Durchlauf liefert folgende Ergebnisse:

Algorithmus	**F1-Score**	**Accuracy**	**Precision**	**Recall**	**AUC**
Two-Class Support Vector Machine	0.565445	0.538889	0.568421	0.562500	0.552703
Two-Class Neural Network	0.685259	0.561111	0.554839	0.895833	0.571181
Two-Class Logistic Regression	0.559585	0.527778	0.556701	0.562500	0.568204
Two-Class Locally-Deep Support Vector Machine	0.547264	0.494444	0.52381	0.572917	0.517361
Two-Class Decision Jungle	0.698413	0.577778	0.564103	0.916667	0.553571
Two-Class Decision Forest	0.556818	0.566667	0.6125	0.510417	0.573289
Two-Class Boosted Decision Tree	0.514620	0.538889	0.586667	0.458333	0.570188
Two-Class Bayes Point Machine	0.684211	0.533333	0.535294	0.947917	0.595982
Two-Class Averaged Perceptron	0.522727	0.533333	0.575000	0.479167	0.573289

Tabelle 7.39: Ergebnisse des Machine Learning: Ethereum Two-class Classification

Algorithmus	**F1-Score**	**Accuracy**	**Precision**	**Recall**	**AUC**
Two-Class Support Vector Machine	0.662937	0.552045	0.578049	0.777049	0.499951
Two-Class Neural Network	0.706199	0.594796	0.599542	0.859016	0.612805
Two-Class Logistic Regression	0.623946	0.585502	0.642361	0.606557	0.597031
Two-Class Locally-Deep Support Vector Machine	0.647555	0.611524	0.666667	0.629508	0.643130
Two-Class Decision Jungle	0.659236	0.602230	0.640867	0.678689	0.639724
Two-Class Decision Forest	0.656958	0.605948	0.648562	0.665574	0.650770
Two-Class Boosted Decision Tree	0.668750	0.605948	0.638806	0.701639	0.649595
Two-Class Bayes Point Machine	0.734082	0.604089	0.592742	0.963934	0.583691
Two-Class Averaged Perceptron	0.566957	0.537175	0.603704	0.534426	0.564160

Tabelle 7.40: Ergebnisse des Machine Learning: Bitcoin Two-class Classification

Algorithmus	R^2	**MAE**	**RMSE**	**RAE**	**RSE**
Neural Network Regression	-0.232605	69.99189	124.834987	0.737855	1.232605
Boosted Decision Tree Regression	0.999326	1.372442	2.918441	0.014468	0.000674
Decision Forest Regression	0.994616	3.514041	8.250093	0.037045	0.005384
Bayesian Linear Regression	0.999994	0.206625	0.272996	0.002178	0.000006

Tabelle 7.41: Ergebnisse des Machine Learning: Ethereum Regression

Algorithmus	R^2	**MAE**	**RMSE**	**RAE**	**RSE**
Neural Network Regression	-0.105524	496.619518	717.756528	1.225332	1.105524
Boosted Decision Tree Regression	0.995827	10.095783	44.099144	0.02491	0.004173
Decision Forest Regression	0.995791	13.301142	44.290157	0.032819	0.004209
Bayesian Linear Regression	0.999946	2.273447	5.032831	0.005609	0.000054

Tabelle 7.42: Ergebnisse des Machine Learning: Bitcoin Regression

Betrachtet man zuerst die beiden Regressionen (Tabellen 7.41 und 7.42), so fällt auf, dass R^2 sehr hoch ist (Maximum ist $R^2 = 1$). Auf den ersten Blick zeugt das von einer nahezu perfekten Kurspreisvorhersage (Anmerkung: Ein R^2 von -0.105524 entspricht $R^2 = 1 - 0.105524 = 0.894476$). Untersucht man das Ergebnis genauer, so sieht man, dass die vorhergesagten Werte sehr nahe an den Tatsächlichen liegen (siehe Tabelle 7.43).

Vorhergesagter Ethereumpreis in USD	**tatsächlicher Preis in USD**
44.4758	44.16
11.3525	11.41
306.6813	306.72
5.2086	5.38

Tabelle 7.43: Ausschnitt des Ergebnisses der Vorhersage des Ethereumkurses mit einer Bayesian Linear Regression (nicht chronologisch; auf vier Nachkommastellen gerundet)

Bei der Beschreibung der Bewertungsmetriken in Punkt 7.1.3 wurde jedoch darauf hingewiesen, dass große Werte für R^2 misstrauisch machen sollten. Dieses Phänomen kann mehre Gründe haben. Hier liegen folgende Vermutungen vor:

- Es liegt ein Fall von Überanpassung (engl. overfitting) vor. „Ein überangepasstes Modell [...] ist zu kompliziert für die [darunterliegenden] Daten“(Frost, 2015, eigene Übersetzung). Das bedeutet, das Modell ist zu speziell angepasst und besitzt zu viele Variablen.

- Wenn zu viele Variablen im Verhältnis zu den Beobachtungen vorliegen, kann es zu einer Zufallskorrelation (engl. chance correlation) kommen.(Lohninger, 1999)
- Wahrscheinlich ist auch, dass die Trends (steigende Kurse) „sehr hohe R^2 Werte erzeugen“.(Frost, 2016, eigene Übersetzung)

Das CRISP-DM Referenzmodell sieht ein Ranking der Modelle vor. An dieser Stelle wird jedoch darauf verzichtet, da die Ergebnisse noch Mängel aufweisen. Bei den Klassifikationen gibt es in dieser Hinsicht keine offensichtlichen Merkmale.
Würde man versuchen, einen Münzwurf mit einem mathematischen Modell vorherzusagen, so würde die statistische Genauigkeit (accuracy) bei 50% liegen. In Tabelle 7.39 und 7.40 ist zu sehen, dass die Accuracy nicht deutlich über diesem Wert liegt. Bei einigen Algorithmen ist jedoch ein Wert von ca. 60% (0.6) erkennbar. Die besten Modelle nach F1-Score sind für Ethereum in Tabelle 7.44 und für Bitcoin in 7.45 aufgestellt.

Algorithmus	**F1-Score**
Two-Class Decision Jungle	0.698413
Two-Class Neural Network	0.685259
Two-Class Bayes Point Machine	0.684211

Tabelle 7.44: Rangliste der besten Ethereum Two-Class Classification Algorithmen

Algorithmus	**F1-Score**
Two-Class Bayes Point Machine	0.734082
Two-Class Neural Network	0.706199
Two-Class Boosted Decision Tree	0.668750

Tabelle 7.45: Rangliste der besten Bitcoin Two-Class Classification Algorithmen

Anzumerken ist hier die interessante Tatsache, dass der Kurs sehr genau vorhergesagt wurde (wenn auch durch Fehler in Modell), eine Aussage darüber, ob der Kurs steigt oder nicht,

jedoch kaum möglich ist. Der Bitcoinkurs scheint stärker von den Faktoren beeinflusst zu werden (höherer F1-Score). Nach der Ergebnisbetrachtung folgt nun die Bewertung hinsichtlich der business objectives und der business success criteria.

7.5 Evaluation

7.5.1 Bewertung der Ergebnisse (Evaluate Results)

Es sollte einerseits geprüft werden, ob die Kurse von Kryptowährungen mit den Mitteln des Machine Learning vorhergesagt werden können. Andererseits sollte eine Einarbeitung in Azure Machine Learning Studio erfolgen und Herausgearbeitet werden, welchen Restriktionen das Werkzeug unterliegt. Schlussfolgern lässt sich, dass der Kurspreis von Bitcoin und Ethereum mit den hier angewandten Mitteln nicht vorhersagbar ist. Es kann aber auch festgehalten werden, dass die Kursschwankungen nicht rein zufällig sind. Eine genauere Betrachtung mit den gewonnen Erkenntnissen (weniger Einflussfaktoren, eventuell eine Time-Series-Analysis) ist sinnvoll (siehe Punkt Punkt 8.3). Der Umgang mit Azure Machine Learning Studio wurde vertraut. Eine Aussage über den Reifegrad und Grenzen des Tools kann gemacht werden (siehe Punkt 8.1).

7.5.2 Rückblick auf den CRISP-DM-Prozess (Review Process)

Neben der technischen Interpretation der Ergebnisse und dem beurteilen der business objectives, folgt ein Rückblick auf die Schritte des CRISP-DM Referenzmodells. In Tabelle 7.46 wird der Projektplan (Tabelle 7.10) aus Punkt 7.1.4 aufgegriffen und mit dem tatsächlichen Verlauf verglichen. Abweichungen werden erläutert und Dinge, die beim nächsten mal verbessert werden können, werden festgehalten (Lessons Learned).

Prozessschritt	**gesch. Aufwand**	**tats. Aufwand**	**Grund für Abweichung**	**Verbesserungen**
Business understanding gesamt	5 Tage	5 Tage	-	Die Ziele genauer beschreiben. Bereits hier sollte sich mit Metriken beschäftigt werden.
Data understanding gesamt	20 Tage	16 Tage	Die Struktur der Daten war eher homogen und selbsterklärend. Der Explorationsschritt war kurz.	Mehr Aufwand für das erforschen der Daten aufbringen und mehr Zusammenhänge suchen.
Data preparation gesamt	10 Tage	29 Tage	Die Transformation erforderte Programmierung (Skripte) in einer nicht vertrauten Sprache. Viele Probleme sind erst beim tatsächlichen Durchführen aufgetaucht. Der Modellierungsaufwand im Azure ML Studio war nicht unerheblich.	Beim Sammeln der Daten auf Formatierung, Trennzeichen und Dateiformate achten. Stücke der Modellierung können als Vorlagen vorbereitet werden (Split Data, Tune Hyperparameter, Score, Evaluate).
Modeling gesamt	5 Tage	10 Tage	Nicht alle Algorithmen können verwendet werden. Deswegen muss die Spezifikation gelesen und eingeschätzt werden. Die Durchlaufzeit der Experimente (vor allem das Modelltrainings) wurde unterschätzt.	Es sollte ein Modell und Modifikationen davon als Kopien erstellt und nachts trainiert werden. Dem Werkzeug nicht zu sehr vertrauen (schnelleres Abbrechen bei langen Laufzeiten).
Evaluation gesamt	5 Tage	5 Tage	-	-

Tabelle 7.46: Rückblick auf den Projektplan

Die Übereinstimmung der Aufteilung nach Shearer mit dem tatsächlichen Aufwand wird in Abbildung 7.14 erkennbar.

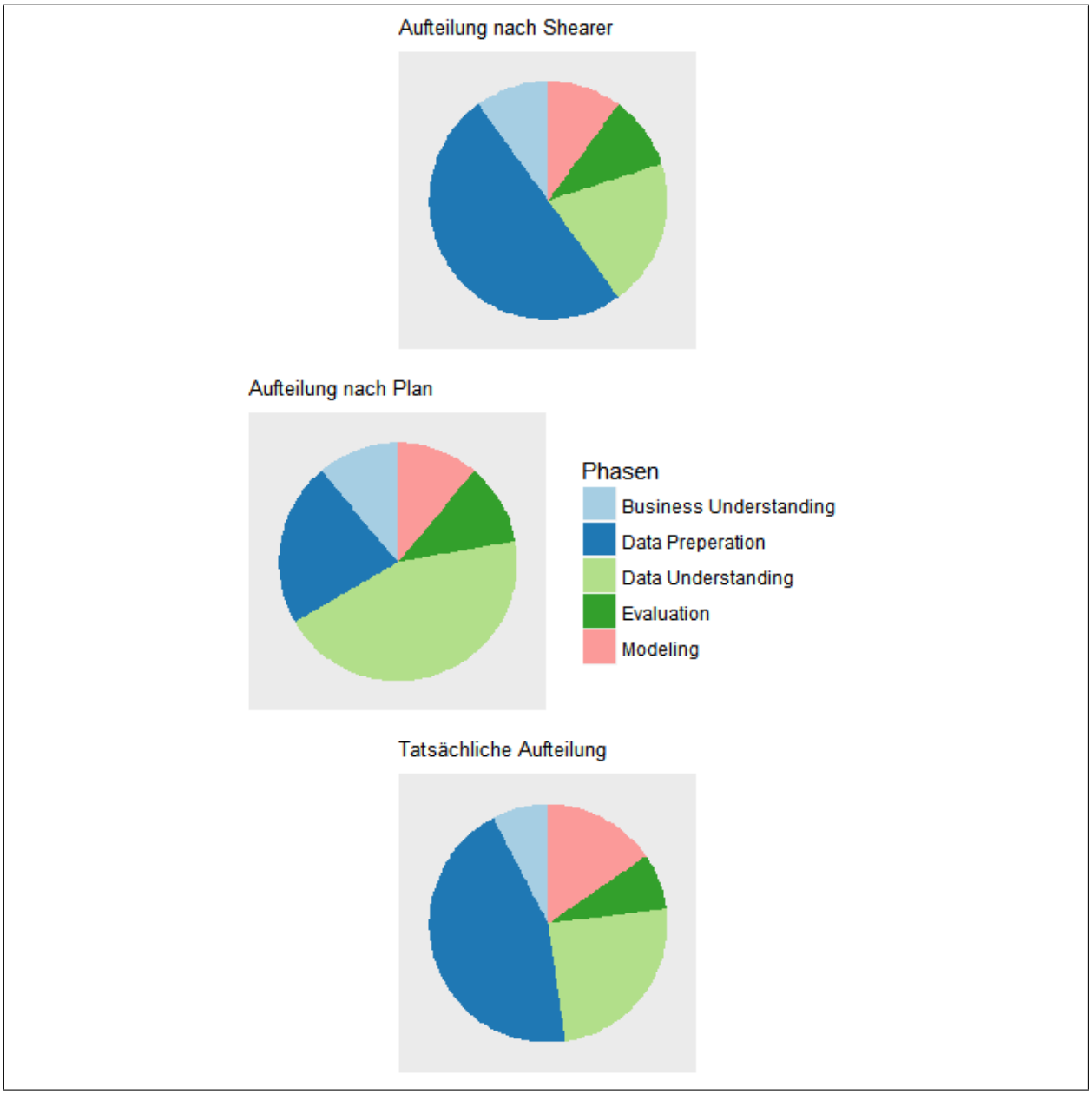

Abbildung 7.14: Tortendiagramme des Projektaufwands nach Shearer, nach Plan und des tatsächlicher Aufwands

7.5.3 Umgang mit den Erkenntnissen (Determine Next Steps)

Im Referenzmodell folgt nach der Evaluation das Deployment. Im Falle dieser Untersuchung wird jedoch darauf verzichtet. Einerseits besitzen die erstellten Modelle noch nicht den Reifegrad, um genutzt zu werden, andererseits würde der Aufwand den Umfang der Arbeit übersteigen. Im weiteren wird wie oben (Punkt 7.5.1) angesprochen, das Werkzeug Azure Machine Learning Studio bewertet (Punkt 8.1), eine Aussage über das Referenzmodell CRIPS-DM getroffen (Punkt 8.2) und die Ergebnisse des Machine Learning interpretiert (Punkt 8.3).

8 Interpretation, Schlussbetrachtung und Fazit

8.1 Bewertung von Azure Machine Learning Studio

Bei Azure Machine Learning Studio handelt es sich um Werkzeug mit klarer Oberfläche. Das Einlernen und Navigieren zwischen Experimenten und Projekten ist sehr einfach. Die Funktionen sind auf dem ersten Blick verständlich. Es bietet die Standardoperationen des Supervised Learning an (Read, Clean, Transform, Split, Train, Tune, Score, Evaluate). Dadurch gelingt es, schnell zu Ergebnissen zu kommen.
Das Werkzeug vermittelt den Eindruck, dass nicht viel Hintergrundwissen benötigt wird. Dies stimmt bei genauerem Hinsehen jedoch nicht. Bei der Auswahl der Algorithmen lassen sich alle Algorithmen auswählen, obwohl sie nicht auf das Problem passen. So lässt sich beispielsweise eine 'Two-Class Bayes Point Machine' auf ein offensichtlich nicht-lineares Problem anwenden. Dies ist kein Fehler des Werkzeugs, es zeigt nur auf, dass nicht auf Hintergrundwissen verzichtet werden kann. Dies Fällt auch auf, wenn das Experiment Item 'Clean Missing Data' genutzt wird: Es muss zwischen acht Möglichen Bereinigungsmethoden entscheiden werden, zu denen auch Probabilistic PCA und MICE gehören. Allein diese Auswahl erfordert Recherchearbeit. Andererseits bietet sich auch die Möglichkeit, verschiedene Einstellungen schnell und einfach auszuprobieren und so über 'trial und error' zu lernen.
Ein großer Kritikpunkt ist, dass es passieren kann, dass Experimente beim Durchlaufen nicht terminieren. Sie laufen ewig oder werden nach einer Stunden ohne Fehlermeldung beendet. In der Free-Version ist eine parallele Verarbeitung mehrerer Experimente nicht möglich, deswegen führen diese Fehler zum Stillstand.
Die Dokumentation ist angemessen und gibt einen guten Überblick über alle Features. An manchen Stellen wäre etwas mehr Information jedoch wünschenswert. Die Untersuchung ist an die Limitation von 100 Experiment Items pro Experiment und die maximale Experimentdauer

von einer Stunde gestoßen.
Azure Machine Learning bietet nur Supervised Learning an.
Grundsätzlich ist der Erweiterbarkeit durch eigene Skripte (Python oder R) keine Grenzen gesetzt, es kann im Tool jedoch kein Debugging durchgeführt werden. Nicht-trivialer Code muss zuvor lokal getestet werden. Ebenfalls gibt es keine Versionsverwaltung, was für Analysten mit Software Engineering Hintergrund sicherlich ein Manko darstellt.
Als Fazit kann festgehalten werden, dass Azure Machine Learning Studio ein gutes Werkzeug ist, das alle Grundfunktionen beinhaltet. Es ist gut dokumentiert und gut für Standardprobleme einsetzbar. Es lässt sich einerseits für Vorstudien in Projekten empfehlen, wenn herausgefunden werden muss, ob ein Problem mit den Mitteln des Machine Learning lösbar ist. Andererseits bietet es durch viele integrierte Beispiele einen perfekten Rahmen für das Lernen des Vorgehens beim Machine Learning.

8.2 Bewertung des CRISP-DM Referenzmodells

Das CRISP-DM Referenzmodell ist ein sehr guter Leitfragen für alle Beteiligten an einem Data Mining Projekt. Für allem für Analysten mit wenig Erfahrung bietet der User Guide eine gewaltige Hilfestellung. Das Modell legt viel Wert auf Dokumentation und Nachvollziehbarkeit. Es versucht, zwischen den betriebswirtschaftlichen, fachlichen und technischen Aspekten zu unterscheiden. Dies gelingt nicht immer ganz. Durch diese Betrachtung aus verschiedenen Blickwinkeln bietet der User Guide viele Tipps und stellt sicher, dass nichts vergessen oder übersehen wird.
Empfehlenswert ist eine Anwendung des Modells als roter Faden in Data Mining, Machine Learning und anderen Data Science oder Analytics-Projekten. Es muss nur so abgeändert werden, dass es auf das Problem passt. Da es sehr umfangreich ist, kann - wie in der vorliegenden Arbeit - etwas weggelassen oder hinzugefügt werden.

8.3 Bewertung der Ergebnisse des Machine Learning

Der Analyseansatz in der Arbeit war, mit der Recherche nach Einflussfaktoren zu beginnen. Dabei wurden viele Faktoren ausgewählt. Diese wurden genauer untersucht, ausgedünnt und bearbeitet. Anschließend wurde das Machine Learning durchgeführt. Wenn dieser Ansatz

verfolgt wird, ist es empfehlenswert, im Vorfeld bereits einen oder zwei Algorithmen auszuwählen (z.B. 'Two-Class Bayes Point Machine', da er widerstandsfähig gegen Overfitting ist), die genutzt werden sollen. Dadurch kann mehr Zeit aufgewendet werden, die Daten in eine bessere Struktur zu bringen.
Mit dem gewonnene Wissen empfiehlt sich jedoch eher die Auswahl von einem oder zwei Faktoren (z.B. Kurs des Dow Jones (DJIA) und des Goldpreises). Dadurch können mehr Hilfsanalysen (z.B. Auseinanderschneiden (engl. slicing) der Daten und Suchen nach Zusammenhängen in bestimmten Zeitperioden) durchgeführt werden. Auch wird weniger Zeit für das Beschreiben und Zusammenführen der Daten verwendet als für detaillierte Untersuchungen.
Es kann festgehalten werden, dass die Arbeit vorherige Untersuchungen unterstreicht, dass der Kursverlauf von Kryptowährungen nicht zufällig ist, sondern mit Faktoren wie dem öffentlichen Interesse (z.B. Google Suchanfragen) zusammenhängt.
Eine andere Erkenntnis ist, dass die Kurse im Analysezeitraum tendenziell steigend. Aus diesem Grund besitzen Regressionen einen sehr hohen Wert für R^2. Eventuell muss eine andere Metrik hier zum Einsatz kommen.
Im realen Betrieb eines Systems zur Vorhersage, müsste darauf geachtet werden, dass die, zur Analyse verwendeten Daten, parallel erhoben werden müssen. So kann sich beispielsweise die Größe „Tageshoch des WTI-Ölpreises“ ständig ändern. Außerdem handelt es sich bei BTC und ETH um die wahrscheinlich prominentesten Kryptowährungen. Kleinere Altcoins unterliegen möglicherweise anderen Einflüssen. Auch die Wechselwirkung zwischen den Währungen darf nicht vernachlässigt werden.
Zum Schluss muss noch erwähnt werden, dass es sich definitiv um ein sehr komplexes Problem handelt, das eventuell mit Standardvorgehen und -mitteln nicht lösbar ist.

Literaturverzeichnis

Androulaki, E., Karame, G. O., Roeschlin, M., Scherer, T., and Capkun, S. (2013). Evaluating User Privacy in Bitcoin. In *Financial Cryptography and Data Security*, Lecture Notes in Computer Science, pages 34–51. Springer, Berlin, Heidelberg.

Appelrath, H.-J., Kagermann, H., and Krcmar, H. (2014). *Future Business Clouds: Ein Beitrag zum Zukunftsprojekt Internetbasierte Dienste für die Wirtschaft*. Herbert Utz Verlag.

Azur, M. J., Stuart, E. A., Frangakis, C., and Leaf, P. J. (2011). Multiple Imputation by Chained Equations: What is it and how does it work? *International journal of methods in psychiatric research*, 20(1):40–49.

Bajpai, P. (2014). Altcoin. http://www.investopedia.com/terms/a/altcoin.asp.

Baur, D. G., Lee, A. D., and Hong, K. (2015). Bitcoin: Currency or Investment? SSRN Scholarly Paper ID 2561183, Social Science Research Network, Rochester, NY.

BitcoinProject (2017a). Difficulty, Network Difficulty - Bitcoin Glossary. https://bitcoin.org/en/glossary/difficulty.

BitcoinProject (2017b). Orphan Block - Bitcoin Glossary. https://bitcoin.org/en/glossary/orphan-block.

BitcoinProject (2017c). Stale Block - Bitcoin Glossary. https://bitcoin.org/en/glossary/stale-block.

Bitkom and KPMG (2017). Welche Public-Cloud-Anwendungen als Software-as-a-Service nutzen Sie?

Brandt, M. (2017). Infografik: Die Top 10 der Kryptowährungen.

Buterin, V. (2016). Let's talk about the projected coin supply over the coming years.. • r/ethereum.

Chandola, V., Banerjee, A., and Kumar, V. (2009). Anomaly detection: A survey. *ACM Computing Surveys*, 41(3):1–58.

Chapelle, O., Schölkopf, B., and Zien, A., editors (2006). *Semi-supervised learning*. Adaptive computation and machine learning. MIT Press, Cambridge, Mass. OCLC: ocm64898359.

Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., and Wirth, R. (2000). CRISP-DM 1.0 Step-by-step data mining guide.

Christidis, K. and Devetsikiotis, M. (2016). Blockchains and Smart Contracts for the Internet of Things. *IEEE Access*, 4:2292–2303.

CoinDesk (2017). Anzahl der Altcoins weltweit in ausgewählten Monaten von Dezember 2015 bis September 2016.

CoinMarketCap (2017). Ranking der größten virtuellen Währungen nach Marktkapitalisierung im Juli 2017 (in Millionen US-Dollar). https://de.statista.com/statistik/daten/studie/296205/umfrage/marktkapitalisierung-digitaler-zahlungsmittel/.

Crowther, P. S. and Cox, R. J. (2005). A Method for Optimal Division of Data Sets for Use in Neural Networks. In *Knowledge-Based Intelligent Information and Engineering Systems*, Lecture Notes in Computer Science, pages 1–7. Springer, Berlin, Heidelberg.

Dannen, C. (2017). *Introducing Ethereum and Solidity*. Apress, Berkeley, CA. DOI: 10.1007/978-1-4842-2535-6.

DeepMind (2017a). AlphaGo -The story of AlphaGo so far.

DeepMind (2017b). DeepMind.

Dhar, V. (2013). Data Science and Prediction. *communications of the acm*, vol. 56 no. 12:10. doi:10.1145/2500499.

Dr. Sayad, S. (2017). Model Evaluation. http://www.saedsayad.com/model_evaluation_r.htm.

EnterpriseMiner, S. (2012). SAS | SEMMA. aus dem Webarchiev. https://web.archive.org/.

Ericson, G. and Rohm, W. A. (2017a). How to choose machine learning algorithms.

Ericson, G. and Rohm, W. A. (2017b). Microsoft Azure Machine Learning: Algorithm Cheat Sheet.

Ericson, G. and Rohm, W. A. (2017c). What is Azure Machine Learning Studio?

Fayyad, U., Piatetsky-Shapiro, G., and Smyth, P. (1996). From data mining to knowledge discovery in databases. *AI magazine*, 17(3):37.

Fraunhofer, I. (2010). Vorteile von SaaS-Angeboten | IT-Anbieter Umfrage.

Fritsch, W. (2013). Salesforce.com überholt im CRM-Markt SAP. http://www.crn.de/software-services/artikel-99222.html.

Frost, J. (2015). The Danger of Overfitting Regression Models | Minitab.

Frost, J. (2016). Five Reasons Why Your R-squared Can Be Too High | Minitab. http://blog.minitab.com/blog/adventures-in-statistics-2/five-reasons-why-your-r-squared-can-be-too-high.

Fund, T. I. M. (2017). World Economic Outlook Database.

FusionMediaLimited (2017). Major Stock Indices. https://www.investing.com/indices/major-indices.

Garcia, D., Tessone, C. J., Mavrodiev, P., and Perony, N. (2014). The digital traces of bubbles: feedback cycles between socio-economic signals in the Bitcoin economy. *Journal of The Royal Society Interface*, 11(99):20140623–20140623.

Gartner (2017). Umsatz mit Software-as-a-Service (SaaS) weltweit von 2010 bis 2016 und Prognose bis 2020 (in Milliarden US-Dollar).

GoogleTrends (2017). GoogleTrends Vergleich: Bitcoin, Ethereum, Cryptocurrency.

Graham, J. W. (2009). Missing Data Analysis: Making It Work in the Real World. *Annual Review of Psychology*, 60(1):549–576.

Hawksby-Robinson, S. (2017). What is Ethereum's inflation rate? (how quickly will new ether be created) - Ethereum Stack Exchange.

Hertle, J. (2016). Datenanalyse - Vorlesung Master, Hochschule München, SS 2016.

HochschuleMünchen (2017). Hochschule München - Bibliothek - Recherche - Übersicht.

IBM (2017). IBM Watson. https://www.ibm.com/watson/.

Jdebunt (2017). What are Ethereum Uncles? themerkle.com.

Kauchak, D. (2016). Neural Networks.

Kim, P. (2017). *MATLAB Deep Learning*. Apress, Berkeley, CA. DOI: 10.1007/978-1-4842-2845-6.

Kraker, P. and Dennerlein, S. (2013). Towards a Model of Interdisciplinary Teamwork for Web Science: What can Social Theory Contribute? *Web Science 2013 Workshop: Harnessing the Power of Social Theory for Web Science*.

Kristoufek, L. (2013). BitCoin meets Google Trends and Wikipedia: Quantifying the relationship between phenomena of the Internet era. *Scientific Reports*, 3:srep03415.

Kristoufek, L. (2015). What Are the Main Drivers of the Bitcoin Price? Evidence from Wavelet Coherence Analysis. *PLOS ONE*, 10(4):e0123923.

Kubat, M. (2017). *An Introduction to Machine Learning*. Springer International Publishing, Cham. DOI: 10.1007/978-3-319-63913-0.

Kumar, raj, s. (2017). Cryptocurrency Historical Prices.

Larose, D. T. (2014). *Discovering Knowledge in Data: An Introduction to Data Mining*. John Wiley & Sons. Google-Books-ID: UGu8AwAAQBAJ.

Leonhart, R. (2014). *Dorsch – Lexikon der Psychologie*. M. A. Wirtz (Hrsg.), Bern: Hogrefe Verlag, 18. edition.

Lison, P. (2012). An introduction to machine learning. http://folk.uio.no/plison/pdfs/talks/machinelearning.pdf.

liveindex.org (2017). Live Index. https://liveindex.org/.

Lohninger, H. (1999). *Teach/Me Data Analysis*. Springer-Verlag, Berlin-New York-Tokyo.

Lohninger, H. (2013). *Grundlagen der Statistik*. Web-version edition.

Mircosoft (2017). Evaluate Model. https://msdn.microsoft.com/library/en-us/Dn905915.aspx.

Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System Bitcoin: A Peer-to-Peer Electronic Cash System.

Olsson, F. (2009). A literature survey of active machine learning in the context of natural language processing.

Paluszek, M. and Thomas, S. (2017). *MATLAB Machine Learning*. Apress, Berkeley, CA. DOI: 10.1007/978-1-4842-2250-8.

Piatetsky, G. (2014). CRISP-DM, still the top methodology for analytics, data mining, or data science projects.

ProjectJupyter (2017a). About Project Jupyter. http://www.jupyter.org.

ProjectJupyter (2017b). The Jupyter Notebook. http://www.jupyter.org.

Ramasubramanian, K. and Singh, A. (2017). *Machine Learning Using R*. Apress, Berkeley, CA. DOI: 10.1007/978-1-4842-2334-5.

Reid, F. and Harrigan, M. (2013). An Analysis of Anonymity in the Bitcoin System. In Altshuler, Y., Elovici, Y., Cremers, A. B., Aharony, N., and Pentland, A., editors, *Security and Privacy in Social Networks*, pages 197–223. Springer New York. DOI: 10.1007/978-1-4614-4139-7_10.

Settles, B. (2010). Active learning literature survey. *University of Wisconsin, Madison*, 52(55-66):11.

Shearer, C. (2000). The CRISP-DM Model: The New Blueprint for Data Mining. *JOURNAL OF DATA WAREHOUSING*, 5(4):13–22.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., Chen, Y., Lillicrap, T., Hui, F., Sifre, L., van den Driessche, G., Graepel, T., and Hassabis, D. (2017). Mastering the game of Go without human knowledge. *Nature*, 550(7676):354–359.

Sowa, A. (2017). *Management der Informationssicherheit*. Springer Fachmedien Wiesbaden, Wiesbaden. DOI: 10.1007/978-3-658-15627-5.

Statista. Root Mean Square Error (RMSE; dt.: Wurzel der mittleren Fehlerquadratsumme) - Statista Definition.

Suthaharan, S. (2016). *Machine Learning Models and Algorithms for Big Data Classification*, volume 36 of *Integrated Series in Information Systems*. Springer US, Boston, MA. DOI: 10.1007/978-1-4899-7641-3.

Swamynathan, M. (2017). *Mastering Machine Learning with Python in Six Steps*. Apress, Berkeley, CA. DOI: 10.1007/978-1-4842-2866-1.

Tipping, M. E. and Bishop, C. M. (1999). Probabilistic Principal Component Analysis. *Journla of the Royal Statistical Society*, 61(3):611–622.

TSYS (2016). Kennen oder nutzen sie virtuelle Währungen wie Bitcoin? https://de.statista.com/statistik/daten/studie/605593/umfrage/bekanntheit-und-nutzung-von-virtuellen-waehrungen-in-deutschland/.

Tual, S. (2015). Ethereum Launches. https://blog.ethereum.org/2015/07/30/ethereum-launches/.

Vermeulen, R., Hoeberichts, M., Vašíček, B., Hájková, D., Šmídková, K., and de Haan, J. (2014). Financial stress indexes and financial crises. *Journal of Economic Literature*, JEL-code: E5, G10.

Wikimedia, F. (2016). Brent (Öl). Page Version ID: 156366434.

Wikimedia, F. (2017). West Texas Intermediate. Page Version ID: 169755263.

WikiTrends (2017). Compare popularity of Bitcoin, Cryptocurrency, Ethereum on Wikipedia | Wiki Trends.

Wood, G. (2014). Ethereum: A secure decentralised generalised transaction ledger. *Ethereum Project Yellow Paper*, 151.

YahooFinance (2017). Major world indices - Yahoo Finance. https://in.finance.yahoo.com/world-indices/.