Seminararbeit

im Modul

Wissen in der modernen Gesellschaft

über das Thema

Maschinelles Lernen als Software-Ökosystem im Spannungsfeld zwischen Entwickler und Anwender

vorgelegt von:	Tobias Zschietzschmann
Matrikelnummer:	2863311
Studiengang:	B. Sc. Informatik
Dozent:	Prof. Dr. H.-G. Gräbe
Semester:	Sommersemester 2017

Inhaltsverzeichnis

1 Einleitung 1
2 Einordnung der wichtigsten Begriffe 2
2.1 Maschinelles Lernen 2
2.1.1 Grundsätzliche Funktionsprinzipien 3
2.1.2 Kategorien und Unterteilung von Lernalgorithmen 4
2.1.3 Genereller Ablauf einer Anwendung maschinellen Lernens 5
2.2 Software-Ökosysteme als Forschungsfeld 6
2.2.1 Herkunft des Terminus und Definitionsansätze 6
2.2.2 Unterscheidungskriterien und Kernkomponenten 7
2.3 Spannungsfelder 9
3 Das Software-Ökosystem des maschinellen Lernens 9
3.1 Maschinelles Lernen als zentrale Technologie 10
3.2 Ökosysteme im Ökosystem 11
3.2.1 Open Source Communitys 11
3.2.2 Unternehmen und Industrie 13
3.2.3 Forschungsbereich und Organisationen 15
3.3 Rollenverteilung innerhalb des Ökosystems 16
3.4 Das Spannungsfeld Entwickler – Anwender 18
4 Zusammenfassung und Ausblick 19
5 Literaturverzeichnis 21

1 Einleitung

Wer die aktuellen Entwicklungstrends in der IT-Branche aufmerksam verfolgt, wird in den Fachmagazinen und Blogs nicht um den Begriff des maschinellen Lernens herumkommen. Es häufen sich Meldungen zu Verbesserungen bei der automatischen Spracherkennung oder dem Erkennen von Bildinhalten, wobei die Systeme scheinbare Wunder vollbringen. So erzielen sie bei der Erkennung von Tumoren der Haut hohe Trefferraten und ziehen bei diesen mit menschlichen Experten gleich oder halten als persönliche Assistenten, die auf Zuruf gesteuert werden, Einzug in die Wohnzimmer der Menschen. Maschinelles Lernen erfährt einen neuen Popularitätsschub, angefacht von der Verfügbarkeit billigen Datenspeichers und sehr großer Datensammlungen sowie den schnelleren Verarbeitungsmöglichkeiten mit modernen Systemen [1, S. 52]. Erst wer tiefer in die Thematik einsteigt, realisiert dabei, dass die Algorithmen, welche in den Anwendungen arbeiten, keine neuen Entdeckungen sind, sondern seit vielen Jahren existieren und ständigen Verbesserungen unterliegen. So arbeiten solche Algorithmen seit Jahren in unseren E-Mail-Postfächern als Spam-Filter[1, S. 53] und bilden die Basis für Empfehlungsdienste beim Onlineshopping oder für die Erkennung von Handschriften auf entsprechenden Eingabegeräten für den PC. Die mathematisch-statistischen Methoden, Algorithmen und Systeme haben eine ständige Verbesserung seit dem Aufkommen der ersten theoretischen Modelle zum maschinellen Lernen in den 1950er-Jahren bis in unsere heutige Zeit erfahren haben. Parallel dazu sind auch Communitys für die Erforschung, Entwicklung und Anwendung, Geschäftsmodelle und Unternehmenszweige, Programmbibliotheken und Projekte in verschiedenen Programmiersprachen und sogar eigene Studienfächer und ganze Studiengänge [2] rund um das Thema entstanden und gewachsen. Gemeinsam sind sie Teil des Software-Ökosystems des maschinellen Lernens (ML-Ökosystem), welches der zentrale Untersuchungsgegenstand dieser Seminararbeit ist.

Ziel der Arbeit ist es, einen aktuellen Überblick über dieses dynamische Software-Ökosystem zu geben, wichtige Komponenten und Hauptakteure zu identifizieren und so einen möglichen Grundstein für umfassendere Analysen des Software-Ökosystems herauszuarbeiten. Dabei soll auch auf die Perspektiven von Entwicklern und Anwendern eingegangen werden, um eventuell vorhandene Probleme und unterschiedliche Interessen zu verdeutlichen.

Nach der Einleitung werden in Kapitel 2 wichtige Begriffe für das Verständnis der Arbeit erläutert. Dabei soll zuerst ein kurzer Überblick zum Begriff des maschinellen Lernens gegeben werden, ohne dabei tiefgreifender auf Funktionsweisen und den mathematischen Hintergrund einzugehen. Es wird eine kurze Einführung in das junge Forschungsfeld der Software-Ökosysteme gegeben und kurz auf den Begriff des Spannungsfelds eingegangen. Kapitel 3 stellt den Hauptteil der Arbeit dar. Dort findet eine Annäherung an das ML-Ökosystem statt und Ansätze aus Kapitel 2 werden mit einbezo-

gen. Dabei sollen auch erste Schlüsselrollen identifiziert werden. Kapitel 3.4 ist dem Spannungsfeld Entwickler - Anwender bezogen auf das maschinelle Lernen gewidmet. Danach folgt in Kapitel 4 noch einmal eine kurze Zusammenfassung und es werden Vorschläge für weiterführende Arbeiten unterbreitet.
Zum maschinellen Lernen selbst soll nur ein sehr kurzer Überblick gegeben werden, um spätere Aussagen über das Software-Ökosystem einordnen zu können. Es ist nicht das Ziel dieser Arbeit, bestimmte Teilbereiche maschinellen Lernens oder die Lernalgorithmen und deren Anwendung umfassend zu erläutern. Vielmehr wird eine entferntere Perspektive eingenommen, welche ermöglichen soll, das Zusammenspiel der Komponenten des Ökosystems zu erfassen.

2 Einordnung der wichtigsten Begriffe

Da verschiedene Begriffe, die in dieser Seminararbeit verwendet werden sollen, in der Literatur und den Forschungsbereichen sehr unterschiedlich verstanden werden, ist es notwendig, zunächst ein gemeinsames Verständnis zu schaffen, bevor das ML-Ökosystem näher bestimmt werden kann. Im folgenden Teil der Arbeit sollen deshalb die Begriffe maschinelles Lernen, Software-Ökosystem und Spannungsfeld erläutert werden.

2.1 Maschinelles Lernen

Der Begriff des Lernens wird in diversen Forschungsbereichen, wie beispielsweise der Kognitionswissenschaft, der KI-Forschung, der Philosophie und der Psychologie, breit diskutiert und in viele verschiedene Subkategorien unterteilt. Der Duden beschreibt *Lernen* unter anderem als das Erlangen eines bestimmten Verhaltens oder einer bestimmten Einstellung durch Erfahrungen oder Einsichten über einen bestimmten Zeitraum hinweg [3]. Aspekte dieser Beschreibung finden sich auch in Definitionsansätzen für *maschinelles Lernen*. Da es nicht Gegenstand dieser Arbeit ist, die Trefflichkeit des Lernbegriffs im Kontext des *maschinellen Lernens* zu erörtern, soll an dieser Stelle nur ein kurzer Definitionsansatz aus dem Forschungsbereich der Künstlichen Intelligenz das Verständnis für diese Arbeit erläutern. Russell und Norvig beschreiben unter Verwendung des Begriffs Agent[1] *Lernen* wie folgt:

[1] Ein Agent stellt einen abstrakten Akteur in einem System dar, welcher über Sensoren und Aktoren verfügt, über die er mit der Umgebung kommuniziert – dieses Konzept erweitern sie auf Softwareagenten [1].

> Ein Agent lernt, wenn er seine Leistung für zukünftige Aufgaben verbessert, nachdem er Beobachtungen über die Welt gemacht hat. Lernen reicht vom Trivialen – wie es beispielsweise beim Notieren einer Telefonnummer geschieht – bis zum Tiefgründigen – wie wir es von Albert Einstein kennen, der eine neue Theorie des Universums hergeleitet hat. [1 S. 809]

Der Begriff maschinelles Lernen wird in der populärwissenschaftlichen Literatur oft mit dem der Künstlichen Intelligenz gleichgesetzt und synonym verwendet. Unter Künstlicher Intelligenz ist allerdings vielmehr ein Forschungsgebiet zu verstehen, welches eine übergeordnete Ebene darstellt und das maschinelle Lernen, neben vielen anderen, als Unterkategorie einschließt. Neben dieser ungenauen Verwendung wird die Bezeichnung auch synonym für das derzeit populäre *Deep Learning* verwendet, welches jedoch wiederum nur eine Teilkategorie des maschinellen Lernens bzw. eine Kategorie von Lernalgorithmen darstellt.
Tatsächlich gibt es eine Vielzahl verschiedener Lernalgorithmen, welche nach verschiedenen Kriterien unterteilt werden. In Kapitel 2.1.2 wird nur ein kurzer Überblick über diese Kategorien gegeben, um dem Leser ein besseres Verständnis des dritten Kapitels und eine Einordnung der dort beschriebenen Komponenten des ML-Ökosystems zu ermöglichen.

2.1.1 Grundsätzliche Funktionsprinzipien

Gemeinsames Ziel von Systemen maschinellen Lernens ist die Bildung eines abstrakten Modells als Ausschnitt der realen Welt. Für dieses Modell sind auf der Basis von Daten eine Zielfunktion zu maximieren, neue Erkenntnisse in eine simulierte Wissensbasis aufzunehmen (welche auf zu dem Erlernten ähnliche Szenarien anwendbar sind) oder eine vorhandene Wissensbasis oder Bewertungsparameter anzupassen, basierend auf Ergebnissen der Lernalgorithmen.
Es stellt sich die Frage, warum überhaupt Interesse daran besteht, dass Maschinen bzw. genauer gesagt deren Software lernt. Die Ursprüngliche Motivation basiert auf dem Bestreben der KI-Forschung, menschliche kognitive Eigenschaften und Problemlösungsvarianten nachzuahmen [1 S. 22-26]. Später hat sich beispielsweise für Software-Entwickler gezeigt, dass viele Probleme nicht auf vordefinierte und einprogrammierte Weise nach festen Regeln lösbar sind. Man kann beispielsweise nicht von vornherein alle eventuellen Probleme berücksichtigen, auf die eine Software für automatisiertes Fahren stoßen könnte, welche aktiv am Verkehr teilnimmt. Gerade für solche Anwendungsszenarien eignet sich Software, welche die Fähigkeit besitzt selbstständig aus bereits bekannten Situationen und vorhandenem Wissen Rückschlüsse zu ziehen und eine Entscheidung für eine unbekannte Situation zu treffen, die nicht vorher vom Entwickler vorausgesehen werden musste.

2.1.2 Kategorien und Unterteilung von Lernalgorithmen

Maschinelles Lernen ist auch ein Oberbegriff, unter dem eine Vielzahl unterschiedlicher Algorithmen zusammengefasst wird, welche das Lernen simulieren. Die meisten dieser Lernalgorithmen lassen sich einer der nachfolgenden Kategorien zuordnen oder nutzen Aspekte aus mehreren davon. Da diese Klassifizierung weit verbreitet und üblich ist, sollen diese Begriffe kurz erläutert werden.

Überwachtes Lernen (Supervised Learning)

Die am häufigsten verwendete Kategorie von Lernalgorithmen ist die des überwachten Lernens, welche laut SAS Institute generell etwa 70 % der angewandten Methoden des maschinellen Lernens ausmachen. Eines der Hauptunterscheidungskriterien zu anderen Kategorien ist die Art des Trainierens der Algorithmen mit vorgegebenen, bereits bewerteten Beispielen. Dem Algorithmus werden Eingabe/Ausgabe-Paare bereitgestellt, wobei die mitgelieferte gewünschte Ausgabe als Vergleichswert für die eigene Ausgabe des Algorithmus dient. Bei identifizierten Fehlern passt das Model seine Gewichtungen entsprechend an [4, 1 S. 811].

Unüberwachtes Lernen (Unsupervised Learning)

Im Gegensatz zum überwachten Lernen wird dem Lernalgorithmus nicht mitgeteilt, ob eine Einordnung oder Entscheidung richtig oder falsch war. Muster werden vom Lernalgorithmus erkannt, „selbst wenn kein explizites Feedback bereitgestellt wird“, so Russell und Norvig [1 S. 811]. Diese Lernmethoden bilden laut dem SAS Institute den Bereich der am zweithäufigsten verwendeten Algorithmen mit einem Anteil von etwa 10-20 % [4].

Halbüberwachtes Lernen (Semi-Supervised Learning)

Das halbüberwachte Lernen bildet eine Mischform und kommt vor allem dort zur Anwendung, wo auch überwachtes Lernen verwendet wird. Es werden sowohl Daten für das Training genutzt, die eine gewünschte bewertete Ausgabe besitzen, als auch solche, die unbewertet sind, bei denen der Lernalgorithmus die Muster selber erkennen soll. Dabei ist die Menge der Beispieldaten, die unbenannt oder unbewertet sind, wesentlich größer. Diese Algorithmen und Methoden kommen vor allem zum Einsatz, wenn es zu aufwendig wäre, alle Trainingsdaten vorher zu bewerten [4, 1 S. 811].

Selbstverstärkendes Lernen (Reinforcement Learning)

Diese Methode setzt vor allem das Konzept von Versuch und Irrtum um. Es wird aus einer Reihe von Verstärkungen gelernt, die positiv oder negativ sein können in Form von Belohnung oder Bestrafung. Die Gewichtung der Faktoren, welche für die Belohnung oder Bestrafung verantwortlich waren, liegen beim Lernalgorithmus selbst [4, 1 S. 811].

Je nach Anwendungsfall ist immer wieder erneut zu entscheiden, welche der Lernmethoden die erfolgversprechendste ist. Auch durch diese Diversität ist das ML-Ökosystem breit gefächert. In Kapitel 3 wird verstärkt darauf eingegangen. Allerdings sind nicht nur die gewählten Lernalgorithmen entscheidend für die erfolgreiche Anwendung maschinellen Lernens, sondern auch Vor- und Nachbereitung, Zielstellung und andere Punkte, wie das folgende Unterkapitel verdeutlicht.

2.1.3 Genereller Ablauf einer Anwendung maschinellen Lernens

Nachfolgend soll kurz der generelle Ablauf einer Anwendung maschinellen Lernens auf ein bestimmtes Problem erläutert werden, um besser nachzuvollziehen, an welchen Stellen Akteure im Software-Ökosystem ansetzen können. In der folgenden Abbildung von Wagstaff vom Jet Propulsion Laboratory in Kalifornien wird der Ablauf übersichtlich in drei Phasen dargestellt.

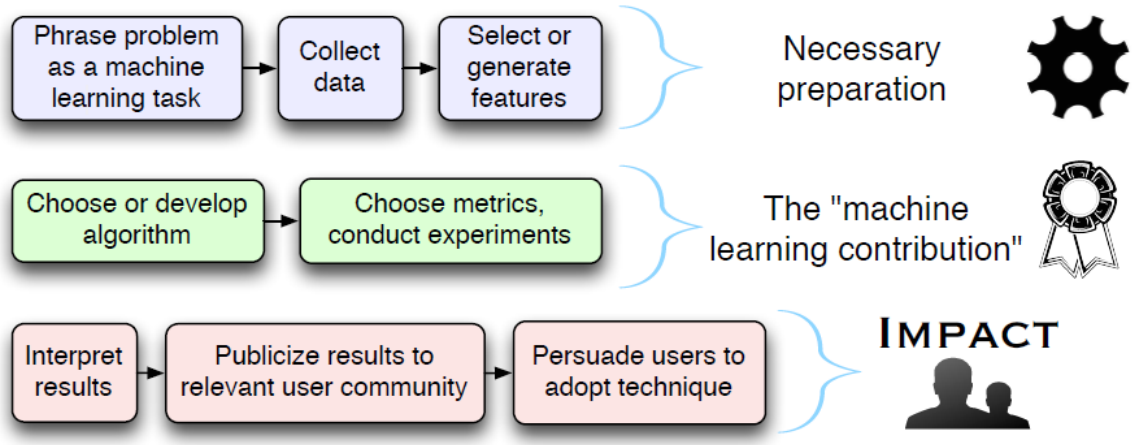

Abbildung 1. Die drei Hauptphasen maschinellen Lernens mit Teilschritten [5 S. 3]

Bevor Methoden des maschinellen Lernens angewendet werden, muss zunächst das eigentliche Problem identifiziert werden und abgeschätzt werden, ob die Anwendung von maschinellem Lernen sinnvoll ist. Danach gilt es, Daten zu beschaffen bzw. vorhandene Daten entsprechend vorzubereiten und eventuell gewichtige Punkte als solche zu kennzeichnen. Anhand der Art des Problems, dem vorhandenen Vorwissen, der Struktur der Daten, der Zielstellung und der Art und Weise, wie die Daten letztendlich dargestellt werden sollen, wird dann ein passender Lernalgorithmus gewählt oder neu entwickelt[5 S. 3]. Unter Verwendung der zuvor identifizierten gewichtigen Punkte werden eventuelle Metriken gesetzt und dann wird das vom Algorithmus erzeugte abstrakte Modell trainiert bis bestmögliche Resultate erzielt werden. Dabei gilt es auch, sogenannte Überanpassung zu vermeiden, bei der durch zu viele betrachtete Parameter relativ zu den zu beobachtenden Ände-

rungen, die Leistungsfähigkeit der Algorithmen wieder sinkt [6]. Schließlich müssen die erhaltenen Ergebnisse von Experten aus der Domäne des Problems interpretiert und die gewonnenen Erkenntnisse publiziert werden.

2.2 Software-Ökosysteme als Forschungsfeld

Damit in Kapiteln 3 genauer auf das ML-Ökosystem eingegangen werden kann, muss vorher geklärt werden, was unter Software-Ökosystemen zu verstehen ist. Um diesen Begriff hat sich ein noch relativ junger Forschungsbereich gebildet. Grundlegend dafür war ein im Jahr 2003 von Messerschmitt und Szyperski veröffentlichtes Buch mit dem Titel „Software Ecosystems: Understanding an Indispensable Technology and Industry“. Darin beschreiben sie wie sich die Software-Industrie technologisch, organisatorisch und sozial von anderen Industrien unterscheidet und gehen dabei auf verschiedene Perspektiven zum Thema Software und Software-Erstellung ein [7]. Besonders interessant für diese Arbeit sind dabei die Perspektiven des Anwenders sowie die des Software-Entwicklers, auf welche in Kapitel 4 dieser Arbeit noch einmal eingegangen wird.
Die Motivation Software-Ökosysteme zu untersuchen, ergibt sich aus der veränderten Situation auf dem Software-Markt, auf welchem eine starke Vernetzung zwischen Verkäufern, Bereitstellern von Diensten, Kunden etc. stattgefunden hat, im Gegensatz zur klareren Trennung in traditionelleren wirtschaftlichen Ökosystemen, so Jansen et al. [8].

2.2.1 Herkunft des Terminus und Definitionsansätze

Zunächst scheint es befremdlich, den Terminus Ökosystem auf Software anzuwenden. Ursprünglich kommt er aus der Biologie und wird z. B. in der Brockhaus-Enzyklopädie beschrieben als „Wirkungsgefüge zwischen Lebewesen verschiedener Arten und ihrem Lebensraum“ [9 S. 281]. Dabei werden auch intermediäre Konsumenten zwischen Produzent und Destruent erwähnt.
Hervorzuheben sind diese Konzepte des Wirkungsgefüges sowie des intermediären Konsumenten, da sie sich auch im Bereich der Software-Ökosysteme wiederfinden lassen. Einer von vielen Definitionsansätzen kommt von Hanssen und Dybå. Sie beschreiben den Terminus wie folgt:

> Software ecosystems is a recent term, referring to a networked community of organizations or actors, which base their relations to each other on a common interest in the development and use of a central software technology. [10 S. 8]

Dabei bauen sie auf Definitionsansätzen von Jansen et al. [8] und Bosch [10 S. 8] auf. Bosch bezieht sich in seiner Definition vor allem auf das gemeinsame Software- bzw. Technologie-Interesse. Jansen dagegen hebt vor allem den Aspekt des geteilten Marktes hervor, welcher im übertragenen

Sinne als der Lebensraum aus der obigen Definition angesehen werden könnte. Die Gemeinschaft von Organisationen und Akteuren, die ein Netzwerk bilden, könnte als das Wirkungsgefüge zwischen den Lebewesen angesehen werden. Der Aspekt der intermediären Konsumenten kommt insofern zur Geltung, dass Akteure und Organisationen vom Software-Ökosystem profitieren können und ihre Beiträge dann der Gemeinschaft wieder zur Verfügung stellen, in Form von Weiterentwicklungen oder Feedback.

Die Verwendung des Begriffs Ökosystem in diesem Kontext wird jedoch auch kritisch betrachtet. So wird beispielsweise auf den Seiten der Free Software Foundation zum GNU-Projekt angemerkt, dass die Verwendung des Begriffs Ökosystem für die Beschreibung einer menschlichen Gemeinschaft ungeeignet sei, da dadurch das Fehlen eines ethischen Urteils impliziert würde [11]. Nach ihrer Auffassung drücke der Begriff auch die „Haltung einer werturteilsfreien Betrachtung“ aus, welche „nur untersuchen und verstehen“ wolle, nicht aber hinterfrage, ob dies recht oder unrecht ist, analog zur Betrachtung ökologischer Phänomene in der Natur. Dagegen könnten „Wesen[,] […] die eine ethische Haltung gegenüber ihrer Umgebung einnehmen“ regulierend, fördernd oder erhaltend einwirken.

Neben den erwähnten Ansätzen, welche sich aus einer eher wirtschaftlichen Perspektive dem Thema Software-Ökosysteme nähern, existieren auch andere Herangehensweise. Als ebenfalls interessant für diese Arbeit zu nennen, ist ein eher sozio-technischer Definitionsansatz von Lungu:

> A **software ecosystem** is a collection of software projects which are developed and which co-evolve together in the same environment. [12 S. 27 Hervorhebung im Original]

Als Umgebungen identifiziert Lungu in seiner Arbeit Unternehmen, Forschungsbereich und Open Source Communitys. Des Weiteren erwähnt er einen „associated social context which emerges as a result of the collaboration between the different contributors to the projects in the ecosystem” und stellt die folgende Behauptung auf: „Each software ecosystem is characterized by its associated social structure” [12 S. 27].

In Kapitel 3.2 soll zunächst über diesen Ansatz eine Näherung an das Ökosystem des maschinellen Lernens erfolgen. Danach werden Kriterien von Hanssen und Dybå einbezogen.

2.2.2 Unterscheidungskriterien und Kernkomponenten

Eine Möglichkeit Software-Ökosysteme zu unterscheiden, stellen also nach Lungu die schon erwähnten Umgebungen Unternehmensbereich, Forschungsbereich und Open Source Communitys dar. Die Grenze zwischen diesen verläuft jedoch nicht immer eindeutig - je nachdem wie sich die Software-Ökosysteme entwickeln.

Weitere Möglichkeiten zeigen die Arbeiten von Hanssen und Dybå auf. Neben den beiden fundamentalen Aspekten der Gemeinschaftsbildung mit netzwerkartigem Charakter von Organisationen und Akteuren und dem gemeinsamen Interesse der Weiterentwicklung einer zentralen Software-Technologie gibt es ihrer Ansicht nach weitere wichtige Faktoren, welche ein Software-Ökosystem ausmachen [10 S. 11]. Dazu gehören:

- der Aspekt der Selbstregulierung, durch häufige Interaktion und Rückkopplung zwischen den Akteuren,
- das Vorhandensein zentraler Keystone-Organisationen (Organisationen, welche die Entwicklung der zentralen Software-Technologie maßgeblich vorantreiben),
- eine Basis gemeinsamer Werte als Motivation aller Akteure für aktives Beitragen zum Ökosystem und
- die Nutzung von Informations- und Kommunikationstechnologien.

Weiterhin werden verschiedene Kriterien erläutert, welche Funktionen und Eigenschaften von Software-Ökosystemen beschreiben und auch ermöglichen, verschiedene Arten von diesen zu unterscheiden. Im Folgenden wird kurz aufgelistet, auf welche davon in dieser Arbeit besonderer Wert gelegt wurde, um das ML-Ökosystem ansatzweise zu beschreiben.

Rollen im Software-Ökosystem und Beziehungen zwischen diesen

Unterschiedliche Beziehungen der Organisationen und Akteure zur Technologie, welche im Mittelpunkt des Ökosystems steht, sind laut Hanssen und Dybå der Grund für die Herausbildung von Rollen in Software-Ökosystemen. Dabei beschreiben sie die drei Schlüsselrollen Keystone-Organisationen, Endnutzer und Third-Party-Organisationen. Außerdem verweisen sie darauf, dass je nach Software-Ökosystem weitere Rollen existieren können. Wichtige Rollen innerhalb des ML-Ökosystems werden in Kapitel 3 näher erläutert [10 S. 9-10].

Keystone-Organisationen und deren Kontrolle des Ökosystems

Zu diesem Kriterium gehören die Identifikation von Keystone-Organisationen, insofern es diese im entsprechenden Software-Ökosystem gibt, und die Beschreibung von deren Einwirken auf das jeweilige Software-Ökosystem. Keystone-Organisationen treiben die Entwicklung der zentralen Software-Technologie maßgeblich voran [10 S. 9-10].

Software-Ökosysteme bilden sich laut Jansen et al. [9] häufig auch um Technologie-Plattformen und nicht nur um konkrete Technologien. Hanssen und Dybå nennen hier als bekannte Beispiele die App-Store Plattform für das Apple iPhone und die Open Source Entwicklungsumgebung Eclipse

mit dem Eclipse-Marketplace [10 S. 8]. Ersteres könne dabei als ein Beispiel für ein geschlossenes, letzteres für ein offenes Software-Ökosystem gelten. Dabei sei entscheidend, ob andere Organisationen Zugang zu Quellcode und Dokumentation haben [10 S. 8]. Diese Offenheit oder Geschlossenheit stellt ein weiteres interessantes Unterscheidungsmerkmal dar.

2.3 Spannungsfelder

Der Begriff Spannungsfeld wird in der Literatur häufig in Titeln verwendet, um ein sonst allgemeineres Thema zu spezifizieren oder einen Betrachtungsaspekt deutlich zu machen. Im Duden wird er beschrieben als ein „Bereich mit unterschiedlichen, gegensätzlichen Kräften, die aufeinander einwirken, sich gegenseitig beeinflussen und auf diese Weise einen Zustand hervorrufen, der wie mit Spannung […] geladen zu sein scheint“ [13]. Dabei könnte man in Anlehnung an ein elektrisches Spannungsfeld von Spannungspolen sprechen, welche Einfluss auf ihre unmittelbare Umwelt nehmen. An ihnen richtet sich das Spannungsfeld aus.
In dieser Arbeit werden Entwickler und Anwender als Pole gegenübergestellt, welche durch unterschiedliche Interessen und Ziele ein Spannungsfeld bilden. Auf diesen Punkt wird weiter in Kapitel 3.4 eingegangen.

3 Das Software-Ökosystem des maschinellen Lernens

Vorweg ist anzumerken, dass das ML-Ökosystem eher einem losen Verbund mehrerer kleiner Ökosysteme ähnelt, die sich jeweils um bestimmte spezifischere Technologien in den jeweils verwendeten Programmiersprachen gebildet haben, wie in Kapitel 3.2 noch weiter erläutert wird. Durch unterschiedliche Zielstellungen und Erwartungen sowie verschiedene Ansätze in der Forschung weist es eine hohe Diversität auf, was auch an der Vielzahl von Programmbibliotheken und Softwarewerkzeugen [14] oder auch an Publikationen in diesem Bereich erkennbar ist. Diese reichen von einer unüberschaubaren Menge an Blogs und Webseiten, Artikeln in populärwissenschaftlichen Magazinen bis hin zu Büchern und Forschungsaufsätzen auf hohem Niveau in wissenschaftlichen Magazinen, wie z. B. dem *Journal of Machine Learning Research* oder *Machine Learning*. Im Wirkungsgefüge des ML-Ökosystems können viele dieser Organisationen unterschiedlichen Rollen zugeordnet werden. Einige Zuordnungsvorschläge sollen in Kapitel 3.2 vorgenommen und erläutert werden. Es ist hervorzuheben, dass Zuordnungsvorschläge zu Rollen und Beschreibungen eines so dynamischen Software-Ökosystems nur Momentaufnahmen sein können, da es sich schnell weiter-

entwickelt. Vorher soll kurz das Geltungsrecht des ML-Ökosystems als Software-Ökosystem untersucht werden.

3.1 Maschinelles Lernen als zentrale Technologie

Einige der in Kapitel 2.2.2 genannten Grundmerkmale für Software-Ökosysteme sind im ML-Ökosystem leicht zu identifizieren. Beispielsweise stellt das maschinelle Lernen an sich, also die entsprechenden Algorithmen und Methoden, die zentrale Software-Technologie des Ökosystems dar. Auch der netzwerkartige Charakter ist klar erkennbar an der Vernetzung der Akteure über gemeinsam genutzte Lernmethoden oder Programmiersprachen und Programmbibliotheken. Die Nutzung von Informations- und Kommunikationstechnologien ist mit dem Internet als verwendetem Medium für das Anbieten von Dienstleistungen und für den Austausch von Informationen gegeben. Dazu werden verschiedene Mittel wie E-Mails, Webseiten von Organisationen oder aber auch Austauschplattformen, wie z. B. reddit, GitHub oder Stack Exchange, genutzt [15, 16].
Schwerer zu erkennen ist dagegen eine Selbstregulation im ML-Ökosystem. Diese kann durch vorhandene Abhängigkeiten zwischen Akteuren im Ökosystem beschrieben werden. Zum Beispiel hängen Unternehmen, welche *Machine Learning as a Service*[2] anbieten, von genutzten Bibliotheken bzw. von deren Anbietern und Communitys ab, wenn ihre eigenen Systeme darauf aufsetzen. Weiterhin kann auch das Übernehmen erfolgreicher Techniken von einer Programmbibliothek oder einem Projekt in ein anderes innerhalb des ML-Ökosystem als selbstregulatorische Funktion herausgestellt werden. Dabei könnte die Regulation in dem Ausgleich der Leistungsfähigkeit verschiedener Teilnehmer im ML-Ökosystem gesehen werden. Eine weitere selbstregulatorische Funktion könnte die Tatsache darstellen, dass nur erfolgreiche Techniken und Methoden für bestimmte Szenarien weiterentwickelt werden. Beispielsweise haben sich je nach Anwendungszweck bestimmte Lernmethoden als erfolgreicher gegenüber anderen erwiesen.
Eine einzelne Keystone Organisation für das gesamte ML-Ökosystem zu finden, wie Hanssen und Dybå als Merkmal von Software-Ökosystemen vorschlagen (siehe Kapitel 2.2.2), ist wegen des dezentralen Aufbaus aus Teilökosystemen, wie in Kapitel 3.2 näher erläutert, vermutlich nicht möglich. Allerdings gibt es eine Reihe zentraler Akteure, die als Keystone Organisation für ihr jeweiliges Teilökosystem auftreten. So übernimmt z. B. Google diese Rolle im TensorFlow-Ökosystem oder die Apache Foundation im Apache Ökosystem, welches mit seiner Programmbibliothek Apache Spark MLlib, mit den Projekten Apache SINGA, Apache Mahout, Apache MXnet und Apache SystemML Teil des ML-Ökosystems ist.

[2] *Machine Learning as a Service* stellt eine Reihe von Diensten dar, welche Methoden des maschinellen Lernens im Rahmen von Cloud Computing Services anbieten

Der Aspekt der geteilten Werte innerhalb des ML-Ökosystems lässt sich aufgrund der unterschiedlichen Umgebungen (Open Source Communitys, Unternehmensbereich, Forschungsbereich) schwer aufzeigen, da bei den Gemeinschaften dieser Umgebungen oft unterschiedliche Interessen im Vordergrund stehen. Eventuell ist diesen das Bestreben zu leistungsfähigeren Algorithmen und Methoden gemeinsam. Bei anderen Punkten, wie z. B. die freie Verfügbarkeit der Software und Erkenntnisse oder finanzielle Interessen gehen die Wertevorstellungen jedoch weit auseinander.
Der geteilte Markt des ML-Ökosystems, wie in der in Kapitel 2.2.2 gegebenen Definition, ist aufgrund der Vielschichtigkeit der Anwendungsgebiete und Geschäftsbereiche nicht so klar erkennbar. Wenn die Hauptkriterien für die Zugehörigkeit zu diesem Markt aber auf die Nutzung von Lernalgorithmen und das Anbieten analytischer Dienste in jeglicher Form reduziert werden, trifft dieser Aspekt vermutlich auf die meisten Akteure des ML-Ökosystems zu.
Eine Zuordnung zu einer bestimmten Umgebung nach Lungus Definition ist nur in Teilen möglich, da das ML-Ökosystem in allen der drei genannten Umgebungen aktive Communitys hat. Einzelne Teile lassen sich jedoch klarer zuordnen, wie im nachfolgenden Kapitel beschrieben wird.

3.2 Ökosysteme im Ökosystem

Es gibt eine Vielzahl kleinerer Communitys im ML-Ökosystem, was sich in verschiedenen Voraussetzungen und Präferenzen der Akteure begründet. Daraus resultiert, dass die Lernalgorithmen und die statistisch-mathematischen Methoden in unterschiedlichen Programmiersprachen umgesetzt oder aufsetzend auf anderen Technologien entwickelt wurden [14]. Die Communitys bilden kleinere Software-Ökosysteme, wie es in diesem Bereich nicht selten der Fall ist [9 S. 2]. Diese sind vollständig oder zum Teil im ML-Ökosystem enthalten. Um dies zu zeigen, eignet sich die Unterscheidung von Lungu nach der Umgebung, wie sie in Kapitel 2.2 erwähnt wurde.

3.2.1 Open Source Communitys

Viele Projekte innerhalb des ML-Ökosystems sind Open Source Projekte. Sie stellen als eigenständige Software-Ökosysteme vermutlich den größten und nutzerreichsten Bereich des ML-Ökosystems dar.
Deutlich wird dies beispielsweise, wenn auf die verwendeten Programmiersprachen und Programmbibliotheken eingegangen wird. In der Literatur und im Internet finden sich verschiedene Quellen zu der Frage, welche Programmiersprache die am häufigsten verwendete innerhalb des ML-Ökosystems ist. Generell kann behauptet werden, dass es je nach Anwendungsbereich eine der beiden Sprachen Python oder R ist [17, 18]. Dies ist auch erkennbar an der großen Anzahl von Büchern zum Thema maschinelles Lernen, in welchen meist eine der Sprachen für Beispiele verwen-

det wird. Beide Sprachen haben eine große Community, bilden jeweils Software-Ökosysteme [19, 20] und sind wiederum mit ihren jeweiligen Programmbibliotheken auch Teil des ML-Ökosystems. Die Grenzen sind dabei nicht immer klar zu ziehen, da es z. B. auch verschiedene Möglichkeiten gibt, Programmbibliotheken von R in Python zu nutzen [21, 22]. Folgende Übersicht zeigt dabei eine Auswahl diverser Programmbibliotheken, welche dem Thema maschinelles Lernen gewidmet sind:

Programmiersprache	**Python**	**R**
Auswahl an Programmbibliotheken die auf maschinelles Lernen ausgerichtet sind	scikit-learn, auto_ml, PyTorch, ML-From-Scratch, TensorFlow, Caffe, BigML, metric-learn, Apache Spark MLlib, Featureforge, pattern, NuPIC, PyBrain, Brainstorm, Crab, Bolt, Shogun, Pyevolve, pydeep, pgmpy, TFLearn, skbayes, keras, neuropredict, imbalanced-learn	ahaz, arules, biglasso, bigrf, bigRR, bmrm, Boruta, CORElearn, caret, caretEnsemble, CoxBoost, Cubist, elasticnet, ElemStatLearn, evtree, forecast, forecastHybrid, GAMBoost, glmnet, glmpath, h2o, kernlab, LiblineaR, klaR, LogicReg, maptree, RWeka, MXNet, randomForest

*Tabelle 1.*Auswahl von Programmbibliotheken zum Thema maschinelles Lernen in R und Python

Diese Programmbibliotheken sind unterschiedlich ausgerichtet. Einige nutzen nur einzelne Kategorien von Lernmethoden oder nur einen einzelnen Algorithmus und spezialisieren sich dabei, während andere versuchen eine Art Sammlung von Lernalgorithmen, eine besonders einfache Schnittstelle oder Methoden für gute visuelle Aufbereitung zu bieten.

Besonders hervorzuheben sind auch Bibliotheken, die für mehrere Programmiersprachen verfügbar sind. Als Beispiel zu nennen wäre hier TensorFlow, was als Entwicklung des Google Brain Teams[3] startete und seit November 2015 dem Open Source Bereich zuzuordnen ist [23, 24]. TensorFlow hat als Programmbibliothek seine eigene Community inklusive Entwicklerkonferenzen etc. und bildet auch ein Software-Ökosystem nach bereits genannten Definitionsansätzen von Hanssen und Dybå[4]. TensorFlow ist in der Entwicklungssprache der Programmbibliothek Python verwendbar, bietet daneben aber auch API's[5] für Java, Go und C an und ist über Umwege auch in R verwendbar [24, 25].

Auch das Unternehmen Facebook hat seine am häufigsten genutzte Plattform für maschinelles Lernen, den FBLearner Flow, unter Open Source Lizenz gestellt und nutzt so die Effekte dieses Software-Ökosystems für sich [26].

[3]Das *Google Brain Team* ist ein Team innerhalb des Unternehmens Google, das sich vor allem mit Technologien rund um das Thema Künstliche Intelligenz befasst und in diesen Bereichen Forschung betreibt.

[4]Die Keystone Organisation stellt Google dar, gemeinsames Interesse der Teilnehmer des Ökosystems ist die Bibliothek weiter zu entwickeln, zentrale Software-Technologie ist die Programmbibliothek selbst, Netzwerkartige Gemeinschaft etc.

[5] *Application Providing Interface* ist die Bezeichnung für Programmierschnittstellen zur Nutzung von bereitgestellten Methoden.

Eine weitere umfangreiche Ansammlung von Open Source Werkzeugen zum Thema maschinelles Lernen bietet die Apache Foundation, welche nochmal in Kapitel 3.3 genannt wird und einen großen Teil der Open Source Projekte zur Verarbeitung großer Datenmengen vereint. Aufsetzend auf unterschiedlichen Datenbank- und Clustertechnologien haben sich dort verschiedene Projekte zum maschinellen Lernen herausgebildet, die von ihrer jeweiligen Open Source Community weiterentwickelt werden. Es finden sich beispielsweise Apache Singa, Apache Mahout, Apache SystemML, Apache MXnet und Apache Spark MLlib, deren Methoden ebenfalls für mehrere Programmiersprachen bereitgestellt werden[6]. In Bezug auf den generellen Ablauf einer Anwendung maschinellen Lernens stellen die zugrundeliegenden Datenbanktechnologien Teile der Phase 1 dar, während die genannten Projekte hauptsächlich in Phase 2 zur Anwendung kommen.

Die bisher erwähnten Technologien richten sich vor allem an Entwickler. Dagegen bietet das in Java geschriebene *Waikato Environment for Knowledge Analysis* (WEKA), einen anderen Ansatz. Die Software ist frei verfügbar, plattformunabhängig und der Quelltext liegt offen. Es stellt Schnittstellen für die Programmiersprache Java bereit, allerdings richtet sich WEKA auch an Anwender, die nicht selbst programmieren wollen [27]. Diesen bietet es eine Reihe von Methoden des maschinellen Lernens, nutzbar mit einer Benutzeroberfläche. Eine andere Ausrichtung ist dabei auch in Bezug auf den Ablauf einer Anwendung maschinellen Lernens erkennbar. Während die vorher genannten Programmbibliotheken vor allem in Phase 2 einzuordnen sind, ist WEKA eine übergreifende Implementierung, welche in allen drei Phasen unterstützt.

WEKA stellt aber vermutlich eher einen Teil des ML-Ökosystems dar, welcher nicht selber als Software-Ökosystem gelten kann. Der netzwerkartige Gemeinschaftscharakter und andere wichtige Merkmale fehlen.

3.2.2 Unternehmen und Industrie

Typische kleinere Software-Ökosysteme im ML-Ökosystem werden neben den Open Source Projekten auch um einige erfolgreiche Technologien, Programmbibliotheken und Dienste von verschiedenen Unternehmen gebildet. Anzumerken ist jedoch, dass viele große Unternehmen ihre genutzten Technologien ganz oder zumindest anteilig in den Open Source Bereich auslagern, wie beispielsweise Google mit TensorFlow oder Facebook mit FBLearner Flow. Daher lassen sich diese Software-Ökosysteme sowohl dem Unternehmensbereich als auch dem Open Source Bereich zuordnen.

Andererseits existieren auch Closed Source Angebote, die andere Formen des Zugangs bieten. Viele davon lassen sich unter der Bezeichnung Machine Learning as a Service im Rahmen von Cloud

[6] Java, Scala, Python und R [28]

Computing-Services zusammenfassen, wie z. B. Googles Cloud Machine Learning oder die Amazon AWS API-Services für maschinelles Lernen [29, 30]. Amazon setzt dabei auf verschiedene Möglichkeiten, sodass ihre Angebote auch über andere häufig genutzte Frameworks und Bibliotheken nutzbar sind. Da ihre eigenen Angebote aber nur über API's nutzbar und nicht quelloffen sind, lassen diese sich eher den Software-Ökosystemen des Unternehmensbereichs zuordnen. Auch IBM Watson Machine Learning Services gehört zu dieser Gruppe und wird zusammen mit anderen Clouddiensten auf der IBM Bluemix Plattform angeboten [31]. Um diese Bluemix-Plattform hat sich ein Software-Ökosystem gebildet, in welchem IBM die Keystone Organisation darstellt. Unter der Nutzung von SPSS, der proprietären Statistik Software von IBM, werden Modelle für die Datenverarbeitung entworfen und dann nach diesem Modell von der Bluemix-Plattform generiert. Nutzer dieser Plattform müssen also einen Überblick über ihre Daten haben und Statistikkenntnisse mitbringen, benötigen aber weniger Programmier- und Entwicklungserfahrung. Diesen Ansatz verfolgt auch Microsoft Azure Machine Learning [32]. Einen Eindruck solcher Modellierungswerkzeuge gibt folgende Abbildung vom Azure Machine Learning Service:

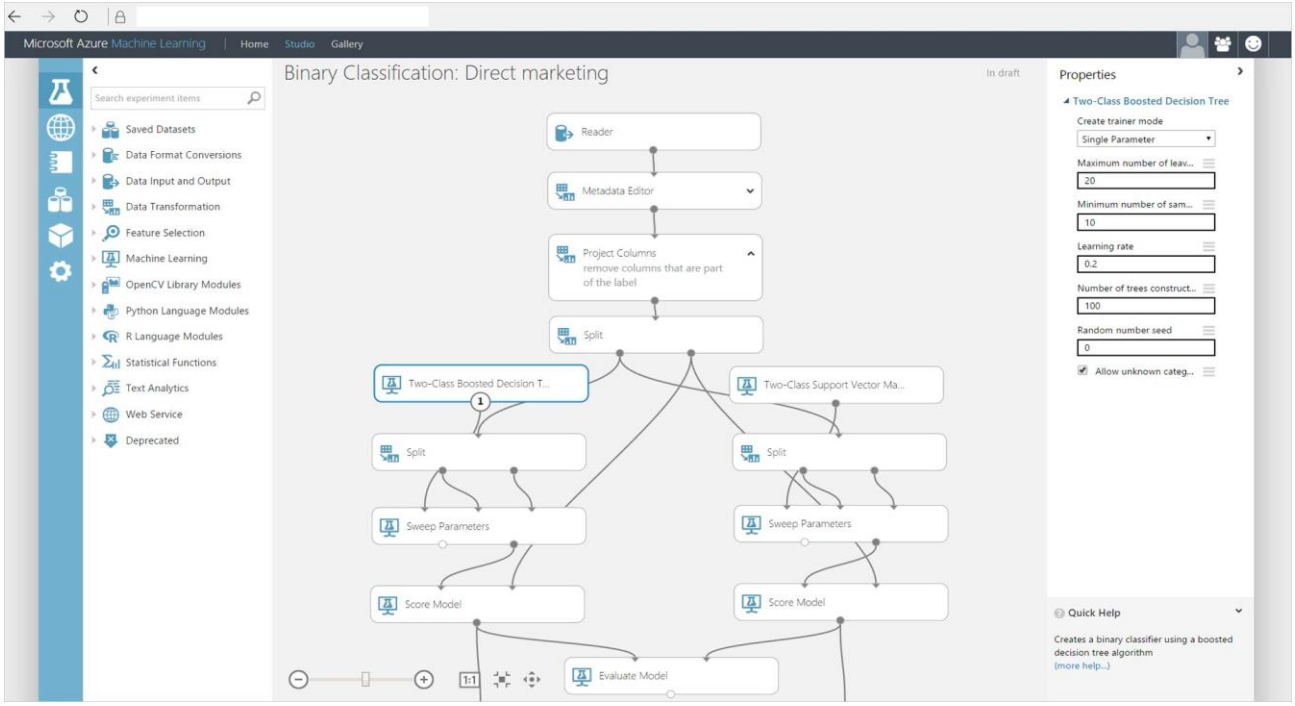

Abbildung 2 Beispiel eines Azure Machine Learning Modells [33]

Weitere ähnliche Beispiele stellen SAP Leonardo Machine Learning im SAP Software-Ökosystem [34] oder die Angebote vom SAS Institute rund um dessen Analyse-Software und Statistik Programmiersprache SAS dar [4]. Ähnlich wie WEKA aus dem Open Source Bereich unterstützen diese Angebote in allen drei Phasen der Anwendung maschinellen Lernens (siehe Kapitel 2.1.3) und legen sich auf keine bestimmten Lernmethoden oder Kategorien des maschinellen Lernens fest.

Neben den erwähnten großen Anbietern existieren unzählige andere und kleinere Anbieter, teils mit neuen Geschäftsideen oder spezialisiert auf bestimmte Branchen.

3.2.3 Forschungsbereich und Organisationen

Aufgrund der hohen Komplexität der Thematik [5 S. 5-6] und der Relevanz für andere Wissenschaften ist es naheliegend, dass auch der Forschungsbereich einen Teil des ML-Ökosystems einnimmt. Nahezu jede größere Forschungsgesellschaft, -organisation oder -unternehmung hat Institute oder Abteilungen, die direkt auf Künstliche Intelligenz oder speziell maschinelles Lernen ausgerichtet sind, wie beispielsweise die 1979 gegründete Association for the Advancement of Artificial Intelligence als eine „nonprofit scientific society“ [35], das Fraunhofer-Institut für Intelligente Analyse- und Informationssysteme oder auch Forschungsunternehmen wie das Deutsche Forschungszentrum für Künstliche Intelligenz, woran auch Google beteiligt ist [36]. Der Wissensaustausch findet dabei sowohl in wissenschaftlichen Magazinen statt als auch auf vielen Konferenzen und im Internet.

Auch im Forschungsbereich lässt sich die Grenze zu den anderen Umgebungen für Software-Ökosysteme nicht klar ziehen. Überschneidungen zum Unternehmensbereich existieren beispielsweise durch Beteiligung an Forschungsunternehmen oder eigenen Ausgründungen mit Forschungsausrichtung, wie Google Deepmind. Die Grenze zum Bereich der Open Source Communitys verschwimmt bei der Verwendung von wissenschaftlich ausgerichteten Programmbibliotheken, wie z. B. scikit-learn in der Programmiersprache Python. Dabei fließen in die Weiterentwicklung die Erkenntnisse von den Forschern, welche diese Bibliotheken verwenden, mit ein. Nicht nur Forschungsinstitute, sondern auch Unternehmen wie IBM und Google werden aktiv bei der Anmeldung von Patenten. Dazu gibt es Diskussionsansätze auf Austauschplattformen im Internet, inwiefern dadurch Forschung und Wettbewerb beeinflusst werden [37].

Es existieren auch kritische Stimmen zur Ausrichtung der Forschung im Bereich maschinelles Lernen. So merkt z. B. Wagstaff einen gewissen Mangel an Realitätsbezogenheit im Forschungsfeld des maschinellen Lernens an und führt dazu die Ausrichtung auf immer leistungsfähigere Algorithmen anhand von standardisierten DataSets als bedenklich an. Sie fordert zu einer praktischeren Ausrichtung auf, die auch wieder Ergebnisse generiert, welche für Bereiche außerhalb des ML-Forschungsfeldes interessant sind [5 S. 5-6].

Neben den Forschungsgesellschaften und Instituten bildet die universitäre Forschung einen weiteren großen Block innerhalb des ML-Ökosystems. Dort wird in vielen unterschiedlichen Fachbereichen, nicht nur in der Informatik und Mathematik, Forschung betrieben, die Bereiche des maschinellen Lernens untersucht und weiterentwickelt.

3.3 Rollenverteilung innerhalb des Ökosystems

Wie bereits in Kapitel 2.2 beschrieben können drei Schlüsselrollen in Software-Ökosystemen identifiziert werden: Keystone Organisationen, Endnutzer und Third Party Organisationen.
Unternehmen können durch das Bereitstellen einer Technologie zu großen Treiber mit entsprechender Stellung in der IT-Branche aufsteigen oder haben eventuell schon einen Bekanntheitsgrad, der sie in die Rolle der Keystone Organisation erhebt. Große Open Source Communitys steigern ihre Reichweite durch das Einbinden einer großen Anzahl von Entwicklern, die Methoden adaptieren und Open Source Werkzeuge in den Unternehmen, für welche sie arbeiten, anwenden. Einige Vorschläge für Rollenzuordnungen sollen anschließend mit kurzer Begründung aufgelistet werden.

Keystone Organisationen

Da das ML-Ökosystem eher einem losen Verbund mehrerer Ökosysteme ähnelt, ist es auch naheliegend, dass es eine Reihe von Akteuren gibt, die die Rolle einer Keystone Organisation übernehmen. Das Unternehmen Google wurde in den vorherigen Kapiteln mehrfach aufgeführt, und zwar als Keystone Organisation in einem Teilökosystem wie TensorFlow im Open Source Bereich, mit seinem Cloud Machine Learning Service im Unternehmensbereich und auch mit seiner Beteiligung im Forschungsbereich. Auf der Grundlage, dass das Unternehmen in allen dieser Umgebungen aktiv das ML-Ökosystem mitgestaltet und maschinelles Lernen auch für die eigenen Geschäftsmodelle nutzt, ist es anscheinend eine der deutlichsten Keystone Organisationen im ML-Ökosystem.
Auch IBM, Amazon und Facebook sind breit aufgestellt im ML-Ökosystem, binden in ihre Angebote viele verschiedene Open Source Projekte ein und treiben so die Entwicklung des ML-Ökosystems maßgeblich voran. Aufgrund dieser Tatsache und ihrer Bekanntheit sowie Größe sollten auch sie zu den Keystone Organisationen des ML-Ökosystems zugeordnet werden.
Nicht nur Unternehmen können die Rolle einer Keystone Organisation erfüllen. Gerade auch die Apache Foundation kann als Paradebeispiel für eine vernetzte Keystone Organisation angesehen werden. Sie stellt das zentrale Bindeglied für eine Vielzahl von teilnehmenden Unternehmen dar. Diese Unternehmen wiederum bringen sich durch ihre Anforderungen maßgeblich in den Open Source Projekten mit ein und haben so auch Einfluss auf die Entwicklung. Dies ist gut sichtbar auf Collaboration-Seiten wie GitHub, auf denen Entwickler aus diesen Unternehmen durch Anmerkungen, Fehlermeldungen und Anforderungen an die Software zum festen Bestandteil der Contributor-Gemeinschaft übergehen. Dabei nimmt maschinelles Lernen auch Einfluss auf andere Technologien der Apache Foundation, welche in Zusammenhang damit stehen, wie beispielsweise auf den Transport von Daten und deren Speicherung (Apache Kafka, Apache Zookeeper, Apache Hive, Apache Hadoop, Apache Cassandra etc.).

Third Party Organisationen

Neben den Keystone Organisationen kann man Third Party Organisationen im ML-Ökosystem ausmachen, welche die zentrale Technologie für die Entwicklung ihrer eigenen damit zusammenhängenden Technologien verwenden. Sie haben im Einzelnen einen geringeren, als Gruppe aber vermutlich einen entscheidenden Einfluss auf die Weiterentwicklung der zentralen Technologie. Diese Gruppe ist sehr groß und anders als bei den Keystone Organisationen nicht beschränkt auf einige wenige, daher soll die folgende Abbildung nur eine ungefähre Vorstellung liefern. Ein Großteil der aufgelisteten Unternehmen und Firmen kann als Third Party Organisation gesehen werden:

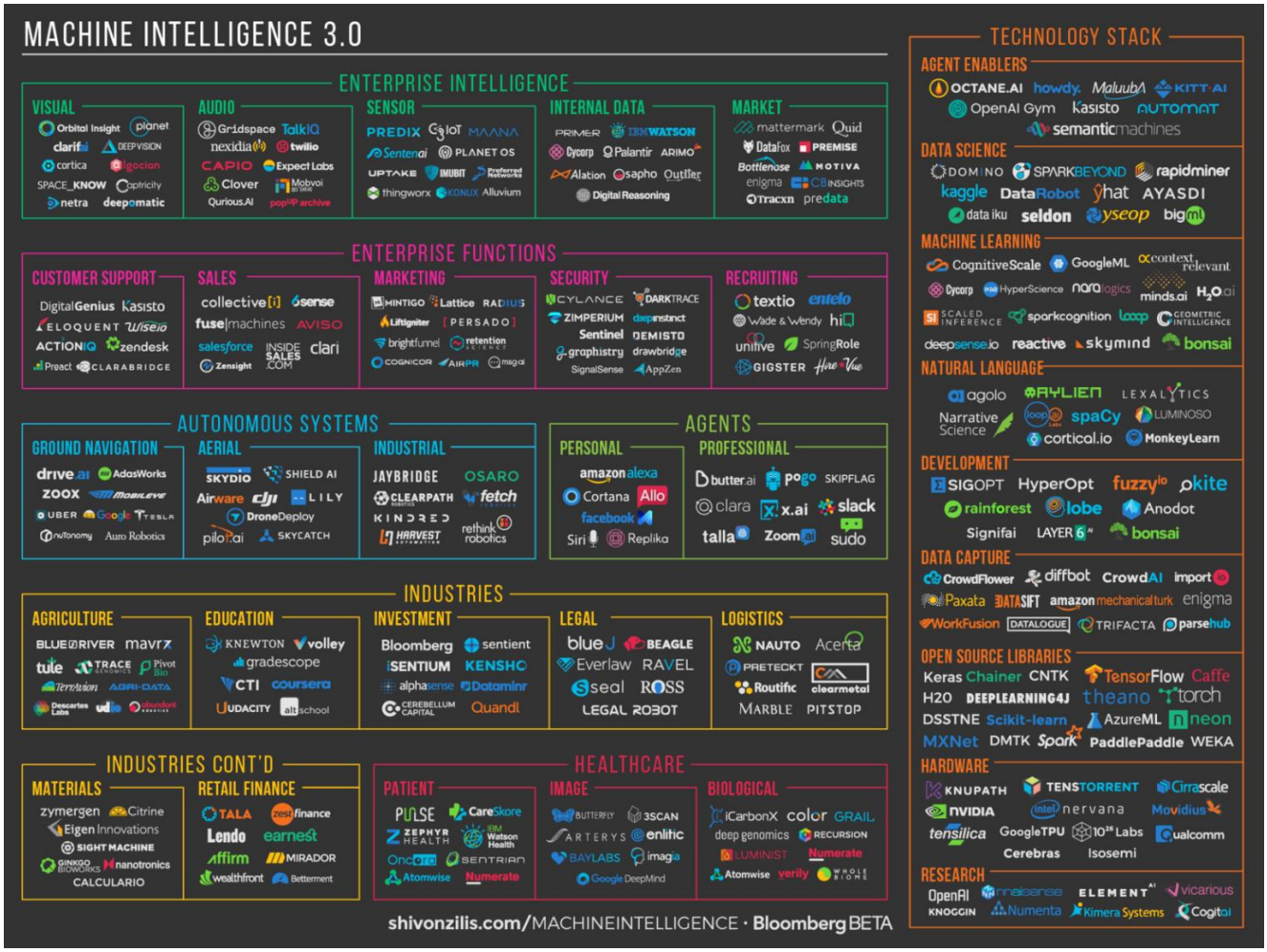

Abbildung 3 Machine Intelligence Landscape [38]

Endnutzer

Die Endnutzerrolle lässt sich nur schwer von der zuvor genannten Rolle der Third Party Organisation trennen. Dort ist eventuell erforderlich eine genauere Definition zu finden. Im ML-Ökosystem lassen sich vermutlich solche Akteure dieser Rolle zuordnen, welche die entwickelten Technologien und angebotenen Dienste im Zusammenhang mit maschinellem Lernen für ihre Industriezweige und Projekte nutzen und nicht direkt mit der Branche der Informationstechnologie zusammenhängen. Es

gibt viele andere Geschäftsfelder und Bereiche, in denen die Verwendung dieser Technologien Einfluss hat. Aufgrund der hohen Anzahl ist es nicht zielführend, eine umfassende Auflistung einzelner Endnutzer zu geben. Einen Eindruck davon, für welche Geschäftsfelder und Industriezweige die Technologie des maschinellen Lernens noch relevant ist, vermittelt folgende Grafik aus dem McKinsey-Report zu Künstlicher Intelligenz und Automation:

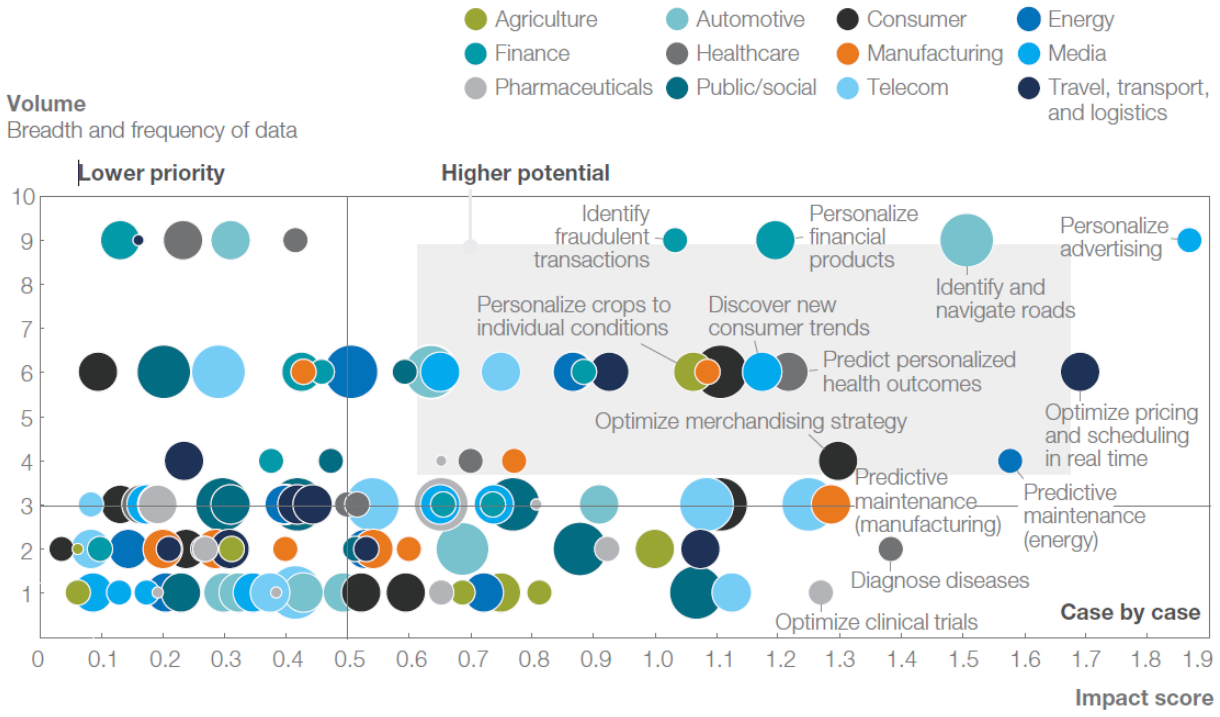

Abbildung 4 Grafik zum Potenzial von maschinellem Lernen in der Industrie [39]

Jeder der Kreise steht für eine andere Art von Daten (nach Quelle oder Verwendungszweck in Datentypen eingeteilt). Die Größe der Kreise beschreibt die Breite und Häufigkeit der Datentypen in diesem Industriezweig. An den Farben sind die unterschiedlichen Industriezweige gut erkennbar. Nahezu jeder Industriezweig hat nach der Grafik von McKinsey Daten, welche ein hohes Potenzial für die gewinnbringende Anwendung von Technologien des maschinellen Lernens bieten. Interessant zu erwähnen ist dabei noch, dass sich kein Kreis für *Public/social* im Bereich mit hohem Potenzial finden lässt.

3.4 Das Spannungsfeld Entwickler – Anwender

Wie bereits in Kapitel 3 ersichtlich, existieren unterschiedliche Angebote für Anwender von Technologien maschinellen Lernens. Die meisten richten sich an Anwender mit tiefgreifenden Kenntnissen in Programmierung und Statistik sowie dem Maschinellen Lernen an sich und bieten verschie-

dene Lernmethoden für Phase 2 des generellen Ablaufs einer Anwendung maschinellen Lernens. Es gibt auch einige Programmbibliotheken und Software-Tools, welche in Phase 3 bei der visuellen Aufbereitung und Darstellung unterstützen [14] oder die bereits spezifischer auf einen Bereich wie z. B. die natürliche Spracheverarbeitung oder Bilderkennung ausgerichtet sind. Für die Phase 1 dagegen gibt es nur wenig Hilfestellung [5 S. 3]. Die Beurteilung und Aufbereitung der Daten bzw. das Formulieren des eigentlichen Problems, welches sich für eine Anwendung von maschinellem Lernen eignet, erfordert umfangreiche Kenntnisse über die zur Verfügung stehenden Methoden und deren Erfordernisse. Daher kommen bei größeren Projekten ganze Teams von Datenexperten und Entwicklern in den Unternehmen zum Einsatz, um das passende Problemmodell zu formulieren.
Daraus kann die Frage abgeleitet werden, ob es nicht notwendig wäre, den Zugang zu dieser Technologie zu vereinfachen. Zum Beispiel merkt Wagstaff an, dass es anderen Wissenschaften noch nicht so einfach möglich ist, Methoden des maschinellen Lernens zu verwenden, wie es mit anderen Methoden der Statistik und Mathematik möglich ist [5 S. 5-6]. Die meisten direkten Anwender sind selbst auch Entwickler und Datenwissenschaftler. Eine Trennung in Entwickler und Anwender, wie diese bei anderen Industriezweigen üblich ist, ist kaum möglich. Vermutlich auch aus diesem Grund wird die Rolle des Endnutzers im ML-Ökosystem hauptsächlich von Unternehmen ausgefüllt, die dann wiederum privaten Anwendern Dienste auf Basis maschinellen Lernens anbieten.
Wenn diese Unternehmen als die Anwender betrachtet werden, kann behauptet werden, dass durch die Anforderungen und Bedürfnisse dieses Anwenders auch die zentrale Technologie des Software-Ökosystems geprägt wird.

4 Zusammenfassung und Ausblick

Anhand der in Kapitel 2 definierten Begriffe und Merkmale wurde gezeigt, dass es ein ML-Ökosystem gibt. Dieses besteht anscheinend aus einem losen Verbund von kleineren Software-Ökosystemen, welche sich um einzelne Technologien innerhalb des ML-Ökosystems gebildet haben. Verschiedene Akteure wurden den in Kapitel 2 definierten Rollen zugeordnet. Somit wurde ein Ansatz für weitere Untersuchungen zu diesem umfangreichen Software-Ökosystem geliefert.
Zusammenfassend kann weiterhin angenommen werden, dass das ML-Ökosystem eher einem offenen Software-Ökosystem als einem geschlossenen entspricht, dass die Mitarbeit und Weiterentwicklung jedem ermöglicht, der bereit ist, sich in dieses komplexe Thema einzuarbeiten. Dabei muss nicht jeder als Software-Entwickler daran mitwirken und alle Lernalgorithmen verstehen, sondern kann nach Belieben sein Teilökosystem wählen, in welchem er mitwirken möchte. Dies kann auf unterschiedliche Weise geschehen, auch als Nutzer mit konstruktivem Feedback. Jedoch

bleibt weiterhin eine gewisse Schranke der Komplexität, welche überwunden werden muss, um aktiv Technologien maschinellen Lernens zu nutzen. Selbst Angebote, welche sich vom reinen Programmieren entfernen, setzen dabei statistisches Verständnis und einen Überblick über die Thematik voraus, um die richtigen Mechanismen für die erfolgreiche Verarbeitung der verwendeten Daten zu wählen.

Auf dieser Seminararbeit aufbauend wäre es im Rahmen einer umfassenderen Bachelor- oder Masterarbeit möglich, anhand von Modellierungsvorschlägen für Software-Ökosysteme von Jansen das ML-Ökosystem zu umreißen. Dabei könnte die momentane Situation der wichtigsten Akteure und Organisationen des Software-Ökosystems Untersuchungsgegenstand sein. Ein solcher Modellierungsversuch hätte den zeitlichen Rahmen dieser Arbeit überstiegen und müsste eventuell auch spezifischer gefasst sein. Dies wäre beispielsweise bezogen auf eine bestimmte Industrie oder Technologie innerhalb des ML-Ökosystems bzw. auf ein Teilökosystem möglich.

5 Literaturverzeichnis

[1] RUSSELL, Stuart J.; NORVIG, Peter (2012). *Künstliche Intelligenz - Ein moderner Ansatz*. Pearson. 3., aktualisierte Auflage. München [u. a.].

[2] Carnegie Melon University. Studienangebote zum maschinellen Lernen. https://www.ml.cmu.edu/academics/index.html zuletzt geprüft: 09.10.2017.

[3] Duden Online-Wörterbuch. s. v. *lernen*. http://www.duden.de/suchen/dudenonline/Lernen zuletzt geprüft: 08.10.2017.

[4] Homepage des SAS Institute. https://www.sas.com/en_id/insights/analytics/machine-learning.html zuletzt geprüft: 08.10.2017.

[5] WAGSTAFF, Kiri L. (2012). „Machine Learning that Matters". http://www.wkiri.com/research/papers/wagstaff-MLmatters-12.pdf zuletzt geprüft: 08.10.2017.

[6] Homepage der StatSoft GmbH. https://www.statsoft.de/glossary/O/Overfitting.htm zuletzt geprüft: 07.10.2017.

[7] MESSERSCHMITT, David G.; SZYPERSKI, Clemens (2003). *Software Ecosystems – Understanding an Indispensable Technology.* 1.Auflage. MIT Press. Cambridge, MA [u. a.].

[8] JANSEN, Slinger et al. (2009). „A Sense of Community: A Research Agenda for Software Ecosystems". *IEEE Machine Learning.* https://slingerjansen.files.wordpress.com/2009/04/ssnniericse.pdf zuletzt geprüft: 08.10.2017.

[9] Der Große Brockhaus: in 30 Bänden. *Band 20 NORDE–PARAK* (2006). 21., völlig neu bearbeitete Auflage. Brockhaus. Leipzig

[10] HANSSEN, Geir K.; DYBÅ, Tore (2012). „Theoretical Foundations of Software Ecosystems". *Proceedings of IWSECO 2012.* http://ceur-ws.org/Vol-879/paper1.pdf zuletzt geprüft: 08.10.2017

[11] Homepage des GNU Projekts. https://www.gnu.org/philosophy/words-to-avoid.html#Ecosystem zuletzt geprüft: 09.10.2017.

[12] LUNGU, Mircea F. . Reverse Engineering Software Ecosystems. Doktorarbeit, Universität von Lugano, 2009 S. http://www.inf.usi.ch/faculty/lanza/Downloads/Lung2009b.pdf zuletzt geprüft: 09.10.2017.

[13] Duden Online Wörterbuch. s v. *Spannungsfeld*.
http://www.duden.de/rechtschreibung/Spannungsfeld
zuletzt geprüft: 07.10.2017.

[14] Kuratierte Sammlung verschiedener Programmbibliotheken zum maschinellen Lernen.
https://github.com/josephmisiti/awesome-machine-learning
zuletzt geprüft: 06.10.2017.

[15] Social News Community reddit. Bereich für maschinelles Lernen.
https://www.reddit.com/r/MachineLearning/#
zuletzt geprüft: 10.10.2017.

[16] Question-and-Answer Website Stack Exchange. Bereich für künstliche Intelligenz.
https://ai.stackexchange.com/
zuletzt geprüft: 10.10.2017.

[17] PIATETSKY-SHAPIRO, Gregory I. . KDNuggets. „Emerging Ecosystem: Data Science and Machine Learning Software, Analyzed".
http://www.kdnuggets.com/2017/06/ecosystem-data-science-machine-learning-software.html
zuletzt geprüft: 09.10.2017.

[18] The Official Blog of Kaggle.com. „Your Year on Kaggle: Most Memorable Community Stats from 2016".
http://blog.kaggle.com/2017/01/05/your-year-on-kaggle-most-memorable-community-stats-from-2016/
zuletzt geprüft: 08.10.2017.

[19] SMITH, David. Slides on the R ecosystem. „Microsoft and the R ecosystem"
https://de.slideshare.net/RevolutionAnalytics/the-r-ecosystem
zuletzt geprüft: 09.10.2017.

[20] MUELLER, John P.; MASSARON, Luca. Webseite der Dummies Buchreihe. „Using the Python Ecosystem for Data Science".
http://www.dummies.com/programming/big-data/data-science/using-the-python-ecosystem-for-data-science/
zuletzt geprüft: 08.10.2017.

[21] rpy2 Repository Webseite. Beispiel für Nutzung von R Programmbibliotheken in Python
https://rpy2.bitbucket.io/
zuletzt geprüft: 08.10.2017.

[22] Python Package Index. s. v. PypeR. Beispiel für Nutzung von R Programmbibliotheken in Python
https://pypi.python.org/pypi/PypeR/1.1.2
zuletzt geprüft: 08.10.2017.

[23] O'Reilly Strata Data Conference. Vortragsankündigung.
https://conferences.oreilly.com/strata/strata-sg/public/schedule/detail/62909
zuletzt geprüft: 09.10.2017.

[24] Homepage des TensorFlow Projekts.
https://www.tensorflow.org/
zuletzt geprüft: 09.10.2017.

[25] Homepage des Projekts R Interface for Tensorflow.
https://tensorflow.rstudio.com/
zuletzt geprüft: 09.10.2017.

[26] DUNN, Jeffrey (2016). „Introducing FBLearner Flow: Facebook's AI backbone".
https://code.facebook.com/posts/1072626246134461/introducing-fblearner-flow-facebook-s-ai-backbone/
zuletzt geprüft: 08.10.2017.

[27] Homepage des WEKA Projekts.
https://www.cs.waikato.ac.nz/ml/weka/
zuletzt geprüft: 09.10.2017.

[28] Homepage der Programmbibliothek MLlib.
https://spark.apache.org/mllib/
zuletzt geprüft: 09.10.2017.

[29] Google Cloud Machine Learning Angebot.
https://cloud.google.com/products/machine-learning/?hl=de
zuletzt geprüft: 09.10.2017.

[30] Amazon Web Services. Bereich für AI-Services.
https://aws.amazon.com/de/amazon-ai/
zuletzt geprüft: 09.10.2017.

[31] IBM Watson Machine Learning Service.
https://developer.ibm.com/clouddataservices/docs/ibm-watson-machine-learning/
zuletzt geprüft: 09.10.2017.

[32] Microsoft Azure Machine Learning Dienste.
https://azure.microsoft.com/de-de/services/machine-learning-studio/
zuletzt geprüft: 09.10.2017.

[33] Abbildungsbeispiel einer Modellierung mit Azure Machine Learning.
https://azurecomcdn.azureedge.net/cvt-f50521786d7494d1129069f701d8751293420aaad404b4aa8260a68606055807/images/page/services/machine-learning-studio/simple-scalable-cutting-edge.jpg
zuletzt geprüft: 09.10.2017.

[34] Homepage der SAP AG. Bereich Machine Learning.
https://www.sap.com/germany/trends/machine-learning.html
zuletzt geprüft: 09.10.2017.

[35] Webseite der Association for the Advancement of Artificial Intelligence.
https://aaai.org

zuletzt geprüft: 09.10.2017.

[36] Webseite des Deutschen Forschungszentrums für Künstliche Intelligenz (2015). „Google ist neuer Gesellschafter des DFKI“.
https://www.dfki.de/web/presse/pressemitteilungen_intern/2015/google-ist-neuer-gesellschafter-des-dfki
zuletzt geprüft: 09.10.2017.

[37] Social News Community reddit. Thread im Bereich für maschinelles Lernen. „Why Google's new patent applications are alarming“
https://www.reddit.com/r/MachineLearning/comments/3cbui2/why_googles_new_patent_applications_are_alarming/
zuletzt geprüft: 09.10.2017.

[38] ZILIS, Shivon; CHAM, James (2016). Webseite des O’Reilly Verlags. „The current state of machine intelligence 3.0”.
https://www.oreilly.com/ideas/the-current-state-of-machine-intelligence-3-0
zuletzt geprüft: 09.10.2017.

[39] MANYIKA, James (2017). Webseite des Unternehmens McKinsey Global Institute. Bereich Global Themes. „What’s now and next in analytics, AI, and Automation“.
https://www.mckinsey.com/global-themes/digital-disruption/whats-now-and-next-in-analytics-ai-and-automation
zuletzt geprüft: 08.10.2017.