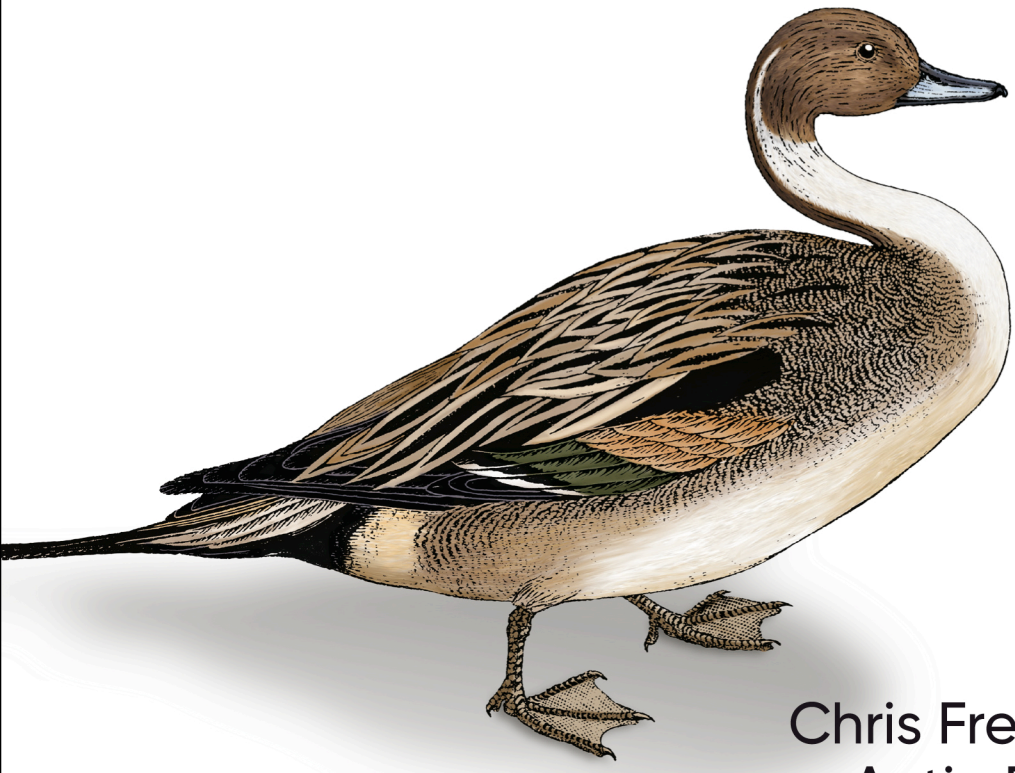
O'REILLY®

Deutsche Ausgabe

Data Science mit AWS

End-to-End-Pipelines für Continuous Machine Learning implementieren



Chris Fregly &
Antje Barth

Übersetzung von Marcus Fraaß

Data Science mit AWS

End-to-End-Pipelines für Continuous Machine Learning implementieren

Chris Fregly & Antje Barth

Deutsche Übersetzung von Marcus Fraaß

O'REILLY®

Chris Fregly und Antje Barth

Lektorat: Alexandra Follenius
Übersetzung: Marcus Fraaß
Korrektorat: Sibylle Feldmann, *www.richtiger-text.de*
Satz: III-Satz, *www.drei-satz.de*
Herstellung: Stefanie Weidner
Umschlaggestaltung: Karen Montgomery, Michael Oréal, *www.oreal.de*
Druck und Bindung: mediaprint solutions GmbH, 33100 Paderborn

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

ISBN:
Print 978-3-96009-184-4
PDF 978-3-96010-655-5
ePub 978-3-96010-656-2
mobi 978-3-96010-657-9

1. Auflage 2022

Wieblinger Weg 17
69123 Heidelberg

Authorized German translation of the English edition of *Data Science on AWS*, ISBN 9781492079392

Dieses Buch erscheint in Kooperation mit O'Reilly Media, Inc. unter dem Imprint »O'REILLY«.
O'REILLY ist ein Markenzeichen und eine eingetragene Marke von O'Reilly Media, Inc. und wird mit Einwilligung des Eigentümers verwendet.

Hinweis:
Dieses Buch wurde auf PEFC-zertifiziertem Papier aus nachhaltiger Waldwirtschaft gedruckt. Der Umwelt zuliebe verzichten wir zusätzlich auf die Einschweißfolie.

Schreiben Sie uns:
Falls Sie Anregungen, Wünsche und Kommentare haben, lassen Sie es uns wissen: *kommentar@oreilly.de*.

5 4 3 2 1 0

Inhalt

Vorwort

Mit diesem Buch lernen KI- und Machine-Learning-Praktikerinnen und -Praktiker, wie sie erfolgreich Data-Science-Projekte auf Amazon Web Services (AWS) entwickeln und deployen können. Der KI- und Machine-Learning-Stack von Amazon gibt Ihnen die Möglichkeit, durch die Vereinigung von Data Science, Data Engineering und Anwendungsentwicklung Ihre Fähigkeiten zu steigern. Dieser Leitfaden zeigt Ihnen, wie Sie Pipelines in der Cloud erstellen und ausführen und anschließend die Ergebnisse innerhalb von Minuten statt Tagen in Anwendungen integrieren können. Chris Fregly und Antje Barth machen zudem im gesamten Buch deutlich, wie Sie Ihre Kosten senken und die Leistung verbessern können.

- Wenden Sie Amazons KI- und ML-Stack auf reale Anwendungsfälle aus den Bereichen der natürlichen Sprachverarbeitung (*Natural Language Processing*, NLP), der Bildverarbeitung (*Computer Vision*), der Erkennung von Betrug (*Fraud Detection*) sowie im Rahmen des Einsatzes intelligenter Kommunikationsgeräte (*Conversational Devices*) und mehr an.
- Greifen Sie auf automatisierte ML-Algorithmen (AutoML) zurück, um bestimmte Anwendungsfälle mit Amazon SageMaker Autopilot umzusetzen.
- Erhalten Sie einen tiefen Einblick in den gesamten Lebenszyklus der Modellentwicklung für einen BERT-basierten Anwendungsfall der natürlichen Sprachverarbeitung (NLP), einschließlich, neben vielem anderen, Datenaufnahme (engl. *Data Ingestion*) und -analyse.
- Bündeln Sie alles in eine wiederverwendbare MLOps-Pipeline (*ML Operations*).
- Erkunden Sie die Möglichkeiten des Einsatzes von ML in Echtzeit, Anomalieerkennung und Streaming-Analysen auf Basis von Echtzeitdatenströmen mit Amazon Kinesis und *Amazon Managed Streaming for Apache Kafka* (Amazon MSK).
- Lernen Sie bewährte Sicherheitspraktiken für Data-Science-Projekte und -Workflows kennen, einschließlich *AWS Identity and Access Management* (IAM), Authentifizierung, Autorisierung, darunter Datenaufnahme und -analyse, Modelltraining und Deployment.

Die Kapitel im Überblick

Kapitel 1 bietet einen allgemeinen Überblick über den sehr umfang- und facettenreichen KI- und ML-Stack von Amazon, der ein enorm leistungsfähiges und vielfältiges Angebot an Diensten, Open-Source-Bibliotheken und Infrastrukturen bietet, die für Data-Science-Projekte jeder Komplexität und Größe genutzt werden können.

Kapitel 2 beschreibt, wie Amazons KI- und ML-Stack in realen Anwendungen aus den Bereichen Empfehlungssysteme, Computer Vision, Betrugserkennung, Verstehen natürlicher Sprache (*Natural Language Understanding*, NLU), Conversational Devices, Cognitive Search, Kundenbetreuung, vorausschauende Wartung (*Predictive Maintenance*) in der Industrie, Hausautomatisierung, Internet der Dinge (*Internet of Things*, IoT), aus dem Gesundheitswesen und auch dem Bereich Quantencomputing eingesetzt werden kann.

Kapitel 3 zeigt, wie Sie mit SageMaker Autopilot AutoML nutzen und einige dieser Anwendungsfälle implementieren können.

In den Kapiteln 4 bis 9 wird der komplette Lebenszyklus der Modellentwicklung (*Model Development Life Cycle*, MDLC) für einen BERT-basierten NLP-Anwendungsfall ausführlich vorgestellt. Die Vorstellung schließt die Datenaufnahme und -analyse, die Auswahl von Features (engl. *Feature Selection*) und das Feature Engineering, das Modelltraining sowie die Modellabstimmung und -bereitstellung mit Amazon SageMaker, Amazon Athena, Amazon Redshift, Amazon EMR, TensorFlow, PyTorch und serverloses Apache Spark mit ein.

In Kapitel 10 wird gezeigt, wie sich alle zuvor gezeigten Teilschritte mithilfe von MLOps auf Basis von SageMaker Pipelines, Kubeflow Pipelines, Apache Airflow, MLflow oder TFX in wiederverwendbaren Pipelines zusammenführen lassen.

Kapitel 11 gibt einen Einblick in den Themenkomplex rund um Echtzeit-ML, Anomalieerkennung und Streaming-Analysen für Echtzeitdatenströme unter Einsatz von Amazon Kinesis und Apache Kafka.

Kapitel 12 stellt eine Vielzahl von bewährten Sicherheitspraktiken für Data-Science-Projekte und -Workflows vor, darunter IAM, Authentifizierung, Autorisierung, Netzwerkisolierung, Verschlüsselung von Daten im Ruhezustand (*Data Encryption at Rest*), Post-Quanten-Netzwerkverschlüsselung bei der Übertragung von Daten, Governance und Auditierbarkeit.

Im gesamten Buch finden Sie zahlreiche Tipps dazu, wie Sie die Kosten senken und die Ergebnisse bzw. die Leistungsfähigkeit von Data-Science-Projekten auf AWS verbessern können.

An wen sich dieses Buch richtet

Dieses Buch richtet sich an alle, die auf der Grundlage von Datenanalysen wichtige Geschäftsentscheidungen treffen. Der Leitfaden hilft Data Analysts, Data Scientists, Data Engineers, Machine Learning Engineers, Research Scientists, Anwendungsentwicklerinnen und -entwicklern sowie DevOps Engineers dabei, ihre Kenntnisse des modernen Data-Science-Stacks auszuweiten und ihre Fähigkeiten im Hinblick auf die Entwicklung in der Cloud zu verbessern.

Amazons KI- und Machine-Learning-Stack vereint die Disziplinen Data Science, Data Engineering und Anwendungsentwicklung und hilft Benutzerinnen und Benutzern, ihre Fähigkeiten über ihre gegenwärtigen Tätigkeiten hinaus zu erweitern. Wir zeigen, wie man Pipelines in der Cloud erstellt und ausführt und dann die Ergebnisse innerhalb von Minuten – nicht in Tagen – in Anwendungen integriert.

Um den größtmöglichen Nutzen aus diesem Buch zu ziehen, sollten Sie idealerweise über die folgenden Kenntnisse verfügen:

- ein grundlegendes Verständnis von Cloud Computing,
- grundlegende Programmierkenntnisse in Python, R, Java/Scala oder SQL sowie
- Grundkenntnisse im Umgang mit Data-Science-Tools wie Jupyter Notebook, Pandas, NumPy oder scikit-learn.

Weitere Ressourcen

Es gibt eine Reihe großartiger Autorinnen und Autoren sowie Quellen, die uns im Hinblick auf dieses Buch inspiriert haben:

- Aurélien Gérons Buch *Praxiseinstieg Machine Learning mit Scikit-Learn, Keras und TensorFlow* (*https://oreilly.de/produkt/praxiseinstieg-machine-learning-mit-scikit-learn-keras-und-tensorflow/*) (O'Reilly, aktuell in 2. Auflage) ist ein hervorragender praktischer Leitfaden für den Aufbau intelligenter ML-Systeme mit gängigen Tools wie Python, scikit-learn und TensorFlow.
- *Deep Learning for Coders with fastai and PyTorch* (*https://www.oreilly.com/library/view/deep-learning-for/9781492045519/*) (O'Reilly) von Jeremy Howard und Sylvain Gugger bietet eine exzellente Einführung in die Erstellung von Deep-Learning-Anwendungen mit PyTorch – und zwar ohne dass ein Doktortitel vonnöten wäre, um dem Buch folgen zu können.
- Das Buch *Building Machine Learning Pipelines* (*https://www.oreilly.com/library/view/building-machine-learning/9781492053187/*) (O'Reilly) von Hannes Hapke und Catherine Nelson ist ein ausgezeichnetes und einfach zu lesendes Nachschlagewerk zum Aufbau von AutoML-Pipelines mit TensorFlow und TFX.
- Das Buch *Programming Quantum Computers* (*https://www.oreilly.com/library/view/programming-quantum-computers/9781492039679/*) (O'Reilly) von Eric R. Johnston, Nic Harrigan und Mercedes Gimeno-Segovia ist eine hervorra-

gende Einführung in Quantencomputer mit leicht verständlichen Beispielen, die den Nutzen von Quanten aufzeigen.

- Micha Gorelick und Ian Ozsvald haben ein Buch für Fortgeschrittene namens *High Performance Python* (*https://www.oreilly.com/library/view/high-performance-python/9781492055013/*) (O'Reilly) verfasst, das viele wertvolle Tipps und Tricks zum Profilen und Optimieren von Python-Code enthält, insbesondere im Hinblick auf eine hochleistungsfähige Datenverarbeitung, das Feature Engineering und das Modelltraining.

Zusätzlich zum Buch haben wir Ihnen eine Webseite (*Data Science on AWS, https://datascienceonaws.com*) bereitgestellt, die Workshops für Fortgeschrittene, monatliche Webinare, Meet-ups, Videos und Folien zu den Inhalten dieses Buchs bietet.

Außerdem teilen wir regelmäßig relevante Blogbeiträge, Konferenzvorträge, Folien, Termine für Meet-ups und Workshops auf Twitter oder LinkedIn:

- Folgen Sie uns auf Twitter: *https://twitter.com/cfregly* und *https://twitter.com/anbarth*
- Auf LinkedIn finden Sie uns ebenfalls: *https://www.linkedin.com/in/cfregly* und *https://www.linkedin.com/in/antje-barth*

In diesem Buch verwendete Konventionen

Die folgenden typografischen Konventionen werden in diesem Buch eingesetzt:

Kursiv
: Kennzeichnet neue Begriffe, URLs, E-Mail-Adressen, Dateinamen und Dateiendungen.

`Konstante Zeichenbreite`
: Wird für Programmlistings und Programmelemente in Textabschnitten wie Variablen- oder Funktionsnamen, Datenbanken, Datentypen, Umgebungsvariablen, Anweisungen und Schlüsselwörter verwendet.

`Konstante Zeichenbreite, fett`
: Kennzeichnet Befehle oder anderen Text, der vom Benutzer wörtlich eingegeben werden sollte.

Dieses Symbol steht für einen Tipp oder eine Empfehlung.

Dieses Symbol steht für einen allgemeinen Hinweis.

Verwenden von Codebeispielen

Ergänzendes Material (Codebeispiele, Übungen usw.) steht in dem Repository *oreilly_book* unter *https://github.com/data-science-on-aws* zum Download bereit. Einige der in diesem Buch gezeigten Codebeispiele sind gekürzt, um eine bestimmte Implementierung hervorzuheben. Das Repository enthält zusätzliche Notebooks, die nicht in diesem Buch behandelt werden, aber für den Leser nützlich sind. Die Notebooks sind jeweils nach den einzelnen Kapiteln des Buchs gegliedert und sollten leicht nachvollziehbar sein. Sie können die Zellen in den einzelnen Notebooks auf bequeme Weise ausführen, indem Sie die Umschalt- und die Eingabetaste gleichzeitig drücken.

Im Rahmen der Ersteinrichtung folgen Sie bitte den Anweisungen bzw. Screenshots in der *README.md*-Datei (unter *Instructions*) sowie den durchnummerierten Jupyter-Notebooks im Ordner *01_introduction*, um

- zunächst ein AWS-Konto zu erstellen (falls noch nicht vorhanden),
- sich in Ihrer AWS Management Console anzumelden,
- SageMaker Studio einzurichten,
- zum Dashboard des IAM-Services zu navigieren,
- als *Rolle SageMakerExecutionRole* auszuwählen,
- die IAM-Richtlinie (*IAM Policy*) *AdminstratorAccess* zuzuordnen, um die Zugriffsberechtigungen festzulegen,
- das GitHub-Repository zu klonen und
- die in diesem Buch verwendeten Bibliotheken zu installieren.

Beachten Sie, dass die hier verwendeten Berechtigungen nur Demonstrationszwecken dienen. Folgen Sie ansonsten bei der Vergabe von Zugriffsrechten in Ihrer Umgebung bitte stets dem Least-Privilege-Prinzip.

Dieses Buch dient dazu, Ihnen beim Erledigen Ihrer Arbeit zu helfen. Im Allgemeinen dürfen Sie die Codebeispiele aus diesem Buch in Ihren eigenen Programmen und der dazugehörigen Dokumentation verwenden. Sie müssen uns dazu nicht um Erlaubnis bitten, solange Sie nicht einen beträchtlichen Teil des Codes reproduzieren. Beispielsweise benötigen Sie keine Erlaubnis, um ein Programm zu schreiben, in dem mehrere Codefragmente aus diesem Buch vorkommen. Wollen Sie dagegen ein Produkt mit Beispielen aus Büchern von O'Reilly verkaufen oder verteilen, benötigen Sie eine Erlaubnis. Eine Frage zu beantworten, indem Sie aus diesem Buch zitieren und ein Codebeispiel wiedergeben, benötigt keine Erlaubnis. Eine beträchtliche Menge Beispielcode aus diesem Buch in die Dokumentation Ihres Produkts aufzunehmen, bedarf hingegen unserer ausdrücklichen Zustimmung.

Wir freuen uns über Zitate, verlangen diese aber nicht. Ein Zitat enthält Titel, Autor, Verlag und ISBN, zum Beispiel: »*Data Science mit AWS* von Chris Fregly und Antje Barth (O'Reilly). Copyright 2022 dpunkt.verlag, ISBN 978-3-96009-184-4.«

Wenn Sie glauben, dass Ihr Einsatz von Codebeispielen über die übliche Nutzung hinausgeht oder außerhalb der oben vorgestellten Nutzungsbedingungen liegt, kontaktieren Sie uns bitte unter *kommentar@oreilly.de*.

Danksagungen

Wir möchten uns bei Gary O'Brien, Development Editor bei O'Reilly, bedanken, der uns bei der Erstellung des Buchs geholfen hat und, was noch wichtiger ist, uns jedes Mal ein Lächeln ins Gesicht gezaubert hat, wenn wir uns unterhalten haben. Danke, Gary, dass wir den Quellcode und die Low-Level-Hardwarespezifikationen in Kapitel 1 aufnehmen durften! Wir möchten uns auch bei Jessica Haberman, Senior Acquisition Editor, bedanken, die uns wichtige Ratschläge zu allen Aspekten vom ersten Buchvorschlag bis zur endgültigen Seitenzahl gegeben hat. Mit ihrer Hilfe konnten wir nach sieben Jahren der Einreichung von Buchvorschlägen die Messlatte so hoch legen, dass der Vorschlag angenommen wurde! Ein besonderer Dank geht an Mike Loukides und Nicole Taché von O'Reilly für ihre wohlüberlegten Empfehlungen zu Beginn des Schreibprozesses, zur Gliederung der Kapitel, zu den Einleitungen und zu den Zusammenfassungen.

Wir möchten uns ganz herzlich bei den Fachgutachtern bedanken, die unermüdlich jede Seite dieses Buchs begutachtet – und erneut begutachtet – haben. Diese Reviewer sind hier in alphabetischer Reihenfolge nach Vornamen aufgeführt: Ali Arsanjani, Andy Petrella, Brent Rabowsky, Dean Wampler, Francesco Mosconi, Hannah Marlowe, Hannes Hapke, Josh Patterson, Josh Wills, Liam Morrison, Noah Gift, Ramine Tinati, Robert Monarch, Roy Ben-Alta, Rustem Feyzkhanov, Sean Owen, Shelbee Eigenbrode, Sireesha Muppala, Stefan Natu, Ted Dunning und Tim O'Brien. Ihr umfassendes technisches Fachwissen und ihr fundiertes Feedback waren nicht nur für dieses Buch von unschätzbarem Wert, sondern auch für die Art und Weise, wie wir in Zukunft technisches Material präsentieren werden. Sie haben dazu beigetragen, dass dieses Buch nicht nur gut, sondern großartig geworden ist, und es hat uns wirklich Spaß gemacht, mit allen diesen Menschen an diesem Projekt gearbeitet zu haben.

Chris

Ich möchte dieses Buch meinem verstorbenen Vater, Thomas Fregly, widmen. Dad, du hast meinen ersten Apple-Computer mit nach Hause gebracht, als ich acht Jahre alt war, und damit mein Leben für immer verändert. Du hast mir im Alter von zehn Jahren geholfen, dein Buch über Infinitesimalrechnung zu verinnerlichen, und mein starkes Interesse an der Mathematik weiter gefestigt. Du hast mir beigebracht, fleißig zu lesen, kurz und bündig zu schreiben, effektiv zu sprechen, schnell zu tippen und frühzeitig Fragen zu stellen. Dadurch, dass ich dir bei der Reparatur eines Bootsmotors zusah, als du auf dem Michigansee gestrandet warst, wurde ich immer wieder dazu inspiriert, tiefer einzutauchen und die Hardware zu verstehen, die meine Software zum Laufen bringt. Bei einem Rundgang durch dein

Büro bei der *Chicago Sun-Times* lernte ich, dass jeder eine interessante Geschichte zu erzählen hat, auch der Rezeptionist, der Geschäftsführer und das Wartungspersonal. Du hast alle gleichermaßen gegrüßt, dich nach ihren Kindern erkundigt, ihnen zugehört und sie mit einer eigenen lustigen Geschichte zum Lachen gebracht. Als ich als Kind an deiner Hand über deinen Universitätscampus lief, lernte ich, dass es in Ordnung ist, den Bürgersteig zu verlassen und mir meinen eigenen Weg durch das Gras zu bahnen. Du sagtest: »Keine Sorge, Chris, sie werden diesen Weg irgendwann pflastern, denn es ist eindeutig der kürzeste Weg vom Ingenieursgebäude zur Cafeteria.« Du hattest recht, Dad. Viele Jahre später gingen wir diesen neu gepflasterten Weg, um uns in der Cafeteria dein Lieblingsgetränk, Pepsi Light, zu holen. Von dir habe ich gelernt, meinen eigenen Weg im Leben zu gehen und nicht immer der Masse zu folgen. Auch wenn du Windows 95 nicht mehr erlebt hast, hast du, ehrlich gesagt, nicht viel verpasst. Und ja, Mac OS ist schließlich zu Linux gewechselt. Auch da hattest du recht.

Ich möchte auch meiner Co-Autorin, Antje Barth, dafür danken, dass sie viele Nächte und Wochenenden investiert hat, um das Schreiben dieses Buchs zu einem fantastischen Erlebnis zu machen. Trotz des Zeitunterschieds von acht bis neun Stunden zwischen San Francisco und Düsseldorf hast du dir für virtuelle Whiteboard-Sitzungen, kurzfristige Quellcodeverbesserungen und Diskussionen im Hinblick auf die Verwendung des Oxford-Kommas immer Zeit genommen. Durch diese Erfahrung sind wir noch bessere Freunde geworden, und ohne dich hätte ich ein so gehaltvolles und hochwertiges Buch nicht erstellen können. Ich freue mich auf die Zusammenarbeit mit dir im Rahmen von zahlreichen zukünftigen Projekten!

Antje

Ich möchte Ted Dunning und Ellen Friedman dafür danken, dass sie mir stets als großartige Mentoren zur Seite stehen und mich immer wieder dazu ermutigen, neue Herausforderungen in Angriff zu nehmen. Ted, wenn wir uns unterhalten, hast du immer weise Worte parat, die mir helfen, die Dinge aus einer anderen Perspektive zu sehen – sei es bei der Vorbereitung auf einen Demowettbewerb oder bei Gesprächen darüber, wie wir unseren Lesern helfen können, das Beste aus diesem Buch herauszuholen. Ellen, ich erinnere mich noch gut daran, wie du mir geholfen hast, überzeugende Vorschläge für Konferenzvorträge zu erstellen, als ich damit anfing, Vorträge für O'Reilly-Strata- und AI-Konferenzen einzureichen. Und bis zum heutigen Tag mache ich mir, wenn es darum geht, einprägsame Titel zu finden, besonders viele Gedanken. Leider kam O'Reilly meinen Vorschlag, als Titel dieses Buchs *Alexa, bitte trainiere mein Modell* zu nehmen, nicht nach.

Ihr beide geht mit gutem Beispiel voran, wenn ihr sagt: »Unterstützt den Traum eines Mädchens, das zu erreichen, was sie erreichen kann.« Aus demselben Grund möchte ich dieses Buch allen Frauen und Mädchen widmen, die von einer Karriere in der IT-Branche träumen oder diese anstreben. Solange ihr an euch selbst glaubt, steht euch nichts im Wege, eure Träume in diesem Berufsfeld zu verwirklichen.

Es gab noch so viele weitere Personen, die mich auf meinem beruflichen Weg unterstützt und ermutigt haben. Ich danke euch allen.

Ich möchte auch Chris dafür danken, dass er ein unterhaltsamer und kompetenter Mitverfasser war. Von Anfang an hast du immer auf die höchsten Standards gepocht, mich dazu gebracht, die Themen zu vertiefen, und mich ermutigt, neugierig zu sein und viele Fragen zu stellen. Du hast mir geholfen, meinen Code zu vereinfachen, meine Gedanken klar darzustellen und endlich das heiß diskutierte Oxford-Komma zu akzeptieren!

KAPITEL 1

Data Science mit AWS – eine Einführung

In diesem Kapitel erörtern wir die Vorteile der Erstellung von Data-Science-Projekten in der Cloud. Zunächst werden die Vorzüge von Cloud Computing dargelegt. Anschließend beschreiben wir einen typischen Arbeitsablauf beim Machine Learning und die allgemeinen Herausforderungen bei der Überführung unserer Modelle und Anwendungen vom Prototyp in die Produktion. Wir gehen auf die allgemeinen Vorteile der Entwicklung von Data-Science-Projekten mit *Amazon Web Services* (AWS) ein und stellen die relevanten AWS-Services für jeden Schritt des Modellentwicklungsworkflows vor. Darüber hinaus zeigen wir bewährte Ansätze zur Gestaltung der Architektur auf, insbesondere bezogen auf Operational Excellence, Sicherheit, Zuverlässigkeit und Leistungs- sowie Kostenoptimierung.

Vorzüge des Cloud Computing

Cloud Computing ermöglicht eine bedarfsgerechte Bereitstellung von IT-Ressourcen über das Internet, wobei sich die Kosten nach dem jeweiligen Bedarf bemessen. Anstatt eigene Rechenzentren und Server zu kaufen, sie zu betreiben und zu warten, können wir Technologien wie Rechenleistung, Speicherplatz, Datenbanken und andere Dienste je nach Bedarf erwerben. Ähnlich wie ein Stromversorger sofort Strom liefert, wenn wir einen Lichtschalter in unserem Haus betätigen, stellt die Cloud IT-Ressourcen bei Bedarf per Mausklick oder durch den Aufruf einer API bereit.

»Es gibt keinen Algorithmus, der Erfahrung kompensiert«, lautet ein berühmtes Zitat von Andy Jassy, ehemals CEO von Amazon Web Services und inzwischen CEO von Amazon. Das Zitat drückt die langjährige Erfahrung des Unternehmens beim Aufbau zuverlässiger, sicherer und leistungsstarker Dienste seit dem Jahr 2006 aus.

AWS hat sein Leistungsportfolio fortwährend erweitert, um praktisch jede Art von Arbeit in der Cloud zu unterstützen, einschließlich zahlreicher Dienste und Funktionen auf dem Gebiet der künstlichen Intelligenz und des maschinellen Lernens. Viele dieser KI- und Machine-Learning-Dienste gehen auf Amazons Pionierarbeit in den Bereichen Empfehlungssysteme, Computer Vision, Sprach- bzw. Textverarbei-

tung und neuronale Netze in den letzten 20 Jahren zurück. Ein Forschungsbeitrag aus dem Jahr 2003 mit dem Titel »Amazon.com Recommendations: Item-to-Item Collaborative Filtering« (*https://oreil.ly/UlCDV*) wurde kürzlich vom Institute of Electrical and Electronics Engineers als ein Beitrag ausgezeichnet, der den »Test der Zeit« überstanden hat. Lassen Sie uns die Vorzüge von Cloud Computing im Zusammenhang mit Data-Science-Projekten in der AWS-Cloud betrachten.

Agilität

Cloud Computing ermöglicht uns, Ressourcen bedarfsgerecht bereitzustellen, was uns wiederum die Möglichkeit bietet, zügig und häufig Experimente durchzuführen. Vielleicht möchten wir eine neue Bibliothek testen, um die Qualität unseres Datensatzes zu überprüfen, oder ein Modell mithilfe der neuesten Generation von Grafikprozessoren (GPUs) schneller trainieren. Innerhalb von Minuten können wir Dutzende, Hunderte oder sogar Tausende von Servern in Betrieb nehmen, die derartige Aufgaben für uns erledigen. Führt ein Experiment nicht zum gewünschten Erfolg, können wir die entsprechenden Ressourcen jederzeit und ohne jegliches Risiko wieder abstellen.

Kosten einsparen

Cloud Computing ermöglicht uns, auf Investitionen mit hohem Kapitaleinsatz zu verzichten und dafür Kosten zu tragen, die variabler Natur sind: Wir zahlen nur für das, was wir tatsächlich in Anspruch nehmen, und müssen keine Vorabinvestitionen in Hardware tätigen, die in ein paar Monaten veraltet sein könnte. Wenn wir Rechenressourcen im Rahmen unserer Datenqualitätsprüfungen und -transformationen oder unserem Modelltraining einsetzen, zahlen wir ausschließlich für die Zeit, in der diese Rechenressourcen genutzt werden. Wir können weitere Kosteneinsparungen erzielen, indem wir Amazon-EC2-Spot-Instanzen für unser Modelltraining nutzen. Mit Spot-Instanzen können wir ungenutzte EC2-Kapazitäten in der AWS-Cloud mit einem Preisnachlass von bis zu 90 % im Vergleich zu On-Demand-Instanzen verwenden. Mit reservierten Instanzen (*Reserved Instances*) und Sparplänen (*Savings Plans*) können wir Geld sparen, indem wir für einen bestimmten Zeitraum im Voraus bezahlen.

Elastizität

Cloud Computing ermöglicht uns, unsere Ressourcen automatisch herauf- oder herunterzuskalieren (engl. *Scaling-out* bzw. *Scaling-in*), um sie an die Bedürfnisse unserer Anwendung anzupassen. Nehmen wir an, wir hätten unsere Data-Science-Anwendung in die Produktion überführt und unser Modell diente der Echtzeitvorhersage. Sollten wir feststellen, dass es zu einem Anstieg der Modellanfragen kommt, können wir die Ressourcen, die das Modell hosten, automatisch erhöhen. Ebenso können wir die Ressourcen automatisch reduzieren, wenn die

Anzahl der Modellanfragen sinkt. Es ist nicht mehr notwendig, unnötig viele Ressourcen bereitzustellen, um Lastspitzen zu bewältigen.

Schneller innovieren

Cloud Computing ermöglicht uns, Innovationen schneller einzuführen, da wir uns auf die Entwicklung von Anwendungen konzentrieren können, die dazu verhelfen, unser Unternehmen von anderen abzuheben, anstatt uns mit der aufwendigen Verwaltung der Infrastruktur zu beschäftigen. Die Cloud hilft uns, mit neuen Algorithmen, Frameworks und Hardware in Sekunden statt Monaten Experimente anzustellen.

Globales Deployment in Minutenschnelle

Cloud Computing ermöglicht uns, unsere Data-Science-Anwendungen innerhalb von Minuten weltweit zur Verfügung zu stellen. In der heute global vernetzten Wirtschaft ist es wichtig, nahe bei unseren Kunden zu sein. AWS hat das Konzept von Regionen etabliert, die physischen Standorten auf der ganzen Welt entsprechen und AWS-Rechenzentren zu Clustern zusammenfasst. Jede Gruppe von logischen Rechenzentren wird als *Availability Zone* (AZ) bezeichnet. Jede AWS-Region besteht aus mehreren isolierten und physisch getrennten AZs innerhalb eines geografischen Gebiets. Die Anzahl der verfügbaren AWS-Regionen und AZs wächst stetig (*https://oreil.ly/qegDk*).

Wir können die weltweite Verfügbarkeit der AWS-Regionen und AZs dazu nutzen, unsere Data-Science-Anwendungen in der Nähe unserer Kunden bereitzustellen, die Anwendungsleistung mit extrem schnellen Reaktionszeiten zu verbessern und die Datenschutzbestimmungen der einzelnen Regionen einzuhalten.

Reibungsloser Übergang vom Prototyp zur Produktion

Einer der Vorteile der Entwicklung von Data-Science-Projekten in der Cloud ist der reibungslose Übergang vom Prototyp zur Produktion. Wir können innerhalb von Minuten von der Programmierung eines Modellprototyps in unserem Notebook zur Überprüfung der Datenqualität oder zum verteilten Modelltraining mit Petabytes an Daten übergehen. Im Anschluss daran können wir unsere trainierten Modelle für Echtzeit- oder Batch-Vorhersagen für Millionen von Nutzerinnen und Nutzern auf der ganzen Welt einsetzen.

Die Erstellung von Prototypen erfolgt häufig in Entwicklungsumgebungen mit nur einem Rechner unter Verwendung von Jupyter Notebook, NumPy und Pandas. Dieser Ansatz funktioniert gut für kleine Datensätze. Bei der Arbeit mit großen Datensätzen werden wir schnell die CPU- und RAM-Ressourcen eines einzelnen Rechners überschreiten. Außerdem möchten wir vielleicht GPUs – oder mehrere Rechner – verwenden, um unser Modelltraining zu beschleunigen. Dies ist mit einer einzelnen Maschine bzw. einem einzelnen Rechner in der Regel nicht möglich.

Die nächste Herausforderung wartet auf uns, wenn wir unser Modell (bzw. unsere Anwendung) in der Produktionsumgebung zum Einsatz bringen bzw. deployen möchten. Zudem müssen wir sicherstellen, dass unsere Anwendung Tausende oder Millionen gleichzeitiger Benutzer global bedienen kann.

Das Deployment in die Produktionsumgebung erfordert oft eine enge Zusammenarbeit zwischen verschiedenen Teams, wie etwa Data Science, Data Engineering, Anwendungsentwicklung und DevOps. Und sobald unsere Anwendung erfolgreich bereitgestellt wurde, müssen wir die Modellleistung und die Datenqualität kontinuierlich überwachen und auf Probleme reagieren, die nach der Überführung des Modells in die Produktion auftreten können.

Die Entwicklung von Data-Science-Projekten in der Cloud ermöglicht uns, unsere Modelle reibungslos – vom Prototyp ausgehend – in die Produktion zu überführen, ohne dass wir eine eigene physische Infrastruktur aufbauen müssen. Verwaltete Cloud-Dienste geben uns die Werkzeuge an die Hand, um unsere Arbeitsabläufe zu automatisieren und Modelle in einer skalierbaren und äußerst leistungsfähigen Produktionsumgebung bereitzustellen.

Data-Science-Pipelines und -Workflows

Data-Science-Pipelines und -Workflows umfassen viele komplexe, multidisziplinäre und iterative Schritte. Nehmen wir als Beispiel einen typischen Workflow im Rahmen der Entwicklung eines Machine-Learning-Modells. Wir beginnen mit der Datenaufbereitung und gehen dann zum Training und zum Feintuning eines Modells über. Schließlich stellen wir unser Modell (bzw. unsere Anwendung) in einer Produktionsumgebung bereit. Jeder dieser Schritte besteht wiederum aus mehreren Teilschritten, wie in Abbildung 1-1 dargestellt.

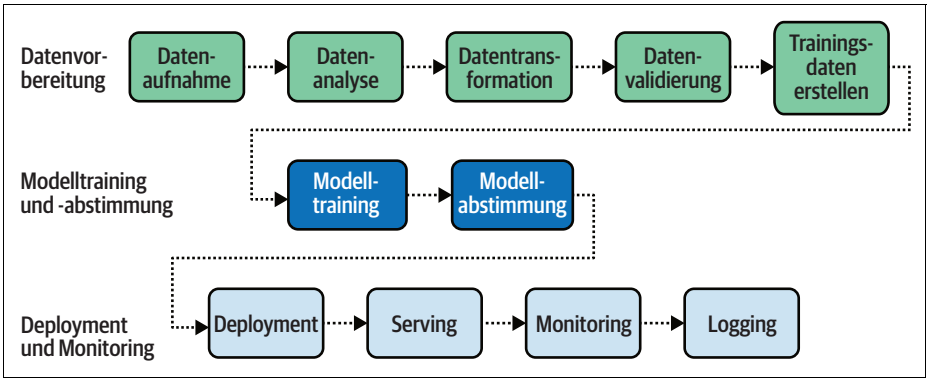


Abbildung 1-1: Ein typischer Machine-Learning-Workflow umfasst viele komplexe, multidisziplinäre und iterative Schritte.

Wenn wir AWS verwenden, befinden sich unsere Rohdaten wahrscheinlich bereits im *Amazon Simple Storage Service* (Amazon S3) und sind als CSV-Datei, als Apache Parquet oder in einem ähnlichen Format gespeichert. Mit den Amazon-KI- oder

-AutoML-Diensten können wir in kürzester Zeit damit beginnen, Modelle zu trainieren, um eine Baseline für die Modellleistung zu erhalten, indem wir direkt auf unseren Datensatz verweisen und auf eine einzige Schaltfläche *train* klicken. Die KI-Dienste und AutoML werden in den Kapiteln 2 und 3 ausführlich behandelt.

Für kundenspezifisch gestaltete Machine-Learning-Modelle – den Hauptschwerpunkt dieses Buchs – können wir mit der manuellen Datenaufnahme und der explorativen Analyse beginnen, einschließlich Datenanalyse, Überprüfung der Datenqualität, zusammenfassender Statistiken, fehlender Werte, Quantilsberechnungen, Analyse der Schiefe der Daten, Korrelationsanalyse usw. Die Kapitel 4 und 5 behandeln ausführlich die Datenaufnahme und die explorative Datenanalyse.

Dann sollten wir die Art der maschinellen Lernaufgabe definieren – Regression, Klassifikation, Clustering usw. Sobald wir die Art des zu lösenden Problems bestimmt haben, können wir einen Machine-Learning-Algorithmus auswählen, der sich am besten für die Lösung des jeweiligen Problems eignet. Je nachdem, welchen Algorithmus wir wählen, müssen wir eine Teilmenge unserer Daten auswählen, um unser Modell zu trainieren, zu validieren und zu testen. Unsere Rohdaten müssen in der Regel in mathematische Vektoren umgewandelt werden, um die numerische Optimierung und das Modelltraining zu ermöglichen. Wir könnten uns zum Beispiel dafür entscheiden, kategoriale Datenspalten in One-Hot-codierte Vektoren umzuwandeln oder textbasierte Datenspalten in Worteinbettungsvektoren, sogenannte Word Embeddings, umzuwandeln. Nachdem wir einen Teil der Rohdaten in Features umgewandelt haben, sollten wir die Features bzw. Daten in Trainings-, Validierungs- und Testdatensätze aufteilen, um sie für das Modelltraining, das Feintuning und das Testen vorzubereiten. Die Auswahl und Transformation von Features – die auch als Merkmale bezeichnet werden – wird in den Kapiteln 5 und 6 detaillierter behandelt.

In der Phase des Modelltrainings wählen wir einen Algorithmus aus und trainieren unser Modell mit unserem Trainingsdatensatz, um zu prüfen, ob unser Programmcode und unser Algorithmus geeignet sind, das gegebene Problem zu lösen. In Kapitel 7 werden wir uns ausgiebig mit dem Modelltraining beschäftigen.

In der Phase der Modellabstimmung bzw. -optimierung stimmen wir die Hyperparameter des Algorithmus ab und bewerten die Leistung des Modells anhand des Validierungsdatensatzes. Wir wiederholen diese Schritte, fügen weitere Daten hinzu oder ändern je nach Bedarf die Hyperparameter, bis das Modell die erwarteten Ergebnisse auf dem Testdatensatz erzielt. Bevor wir das Modell in die Produktion überführen, sollten wir sicherstellen, dass sich diese Ergebnisse mit unserem Geschäftsziel decken. In Kapitel 8 werden wir uns ausführlich mit der Abstimmung von Hyperparametern beschäftigen.

Die letzte Phase – die Überführung von Prototypen in die Produktion – stellt für Data Scientists und Machine-Learning-Experten oft die größte Herausforderung dar. In Kapitel 9 nehmen wir genauer unter die Lupe, wie wir Modelle deployen können.

In Kapitel 10 führen wir alles zu einer automatisierten Pipeline zusammen. In Kapitel 11 widmen wir uns Streaming-Daten und zeigen, wie sich diese analysieren lassen und wie sie in Machine-Learning-Modellen verarbeitet werden können. Kapitel 12 beschreibt die besten Praktiken zur Sicherung von Data-Science-Projekten in der Cloud.

Sobald wir jeden einzelnen Schritt unseres Machine-Learning-Workflows aufgebaut haben, können wir damit beginnen, die Schritte in einer einzelnen, wiederverwendbaren Machine-Learning-Pipeline zu automatisieren. Wenn neue Daten in S3 abgelegt werden, wird unsere Pipeline mit den neuesten Daten neu gestartet, und das neueste Modell wird in die Produktion geschickt, um unsere Anwendungen zu bedienen. Es gibt zahlreiche Tools zur Workflow-Orchestrierung und AWS-Dienste, die uns beim Aufbau automatisierter Pipelines für maschinelle Lernmodelle helfen.

Amazon SageMaker Pipelines

Amazon SageMaker Pipelines bieten die standardmäßige, voll funktionsfähige und vollständigste Methode zur Implementierung von KI- und Machine-Learning-Pipelines in Amazon SageMaker. SageMaker Pipelines sind in SageMaker Feature Store, SageMaker Data Wrangler, SageMaker Processing Jobs, SageMaker Training Jobs, SageMaker Hyperparameter Tuning Jobs, SageMaker Model Registry, SageMaker Batch Transform und SageMaker Model Endpoints integrierbar, die wir im Laufe des Buchs besprechen werden. In Kapitel 10 befassen wir uns eingehend mit verwalteten SageMaker Pipelines und besprechen, wie Sie Pipelines mit AWS Step Functions, Kubeflow Pipelines, Apache Airflow, MLflow, TFX sowie Human-in-the-Loop-Workflows erstellen können.

AWS Step Functions Data Science SDK

Step Functions, ein verwalteter AWS-Service, ist eine großartige Möglichkeit, komplexe Workflows zu erstellen, ohne dass wir eine eigene Infrastruktur aufbauen und warten müssen. Wir können das Step Functions Data Science SDK verwenden, um Machine-Learning-Pipelines aus Python-Umgebungen wie Jupyter Notebook zu erstellen. Step Functions in Bezug auf maschinelles Lernen werden wir uns in Kapitel 10 genauer ansehen.

Kubeflow Pipelines

Kubeflow ist ein relativ neues Ökosystem, das auf Kubernetes aufbaut und ein Orchestrierungssubsystem namens *Kubeflow Pipeline*s enthält. Mit Kubeflow können wir fehlgeschlagene Pipelines neu starten, die Ausführung von Pipelines planen, Trainingsmetriken analysieren und den Verlauf der Entwicklung von Pipelines nachverfolgen bzw. tracken. In Kapitel 10 werden wir uns näher mit der Verwal-

tung eines Kubeflow-Clusters auf *Amazon Elastic Kubernetes Service* (Amazon EKS) befassen.

Managed Workflows for Apache Airflow in AWS

Apache Airflow ist eine sehr ausgereifte und beliebte Lösung, die in erster Linie für die Orchestrierung von Data-Engineering- und ETL-Pipelines (Extract – Transform – Load) entwickelt wurde. Wir können Airflow verwenden, um Workflows als gerichtete azyklische Graphen von Aufgaben zu erstellen. Der Airflow-Scheduler führt unsere Aufgaben auf einem Verbund von Workern aus und folgt dabei den angegebenen Abhängigkeiten. Über die Benutzeroberfläche von Airflow können wir die in der Produktion befindlichen Pipelines visualisieren, den Fortschritt überwachen und bei Bedarf Probleme beheben. In Kapitel 10 werden wir *Amazon Managed Workflows for Apache Airflow* (Amazon MWAA) näher beleuchten.

MLflow

MLflow ist ein Open-Source-Projekt, das sich ursprünglich auf die Rückverfolgbarkeit (Tracking) von Experimenten konzentrierte, jetzt aber auch Pipelines namens *MLflow Workflows* unterstützt. Wir können MLflow auch dazu verwenden, Experimente mit Kubeflow- und Apache-Airflow-Workflows zu tracken. MLflow erfordert jedoch, dass wir unsere eigenen Amazon-EC2- oder Amazon-EKS-Cluster aufbauen und warten. Wir werden MLflow noch ausführlicher in Kapitel 10 vorstellen.

TensorFlow Extended

TensorFlow Extended (TFX) ist eine Open-Source-Sammlung von Python-Bibliotheken, die in einem Pipeline-Orchestrator wie AWS Step Functions, Kubeflow Pipelines, Apache Airflow oder MLflow verwendet werden. TFX ist spezifisch für TensorFlow und hängt von einem anderen Open-Source-Projekt, Apache Beam, ab, um über einen einzelnen Verarbeitungsknoten hinaus zu skalieren. In Kapitel 10 werden wir TFX ausführlicher besprechen.

Human-in-the-Loop – Workflows, die den Menschen einbeziehen

Während die auf KI und Machine Learning basierenden Dienste unser Leben einfacher machen, ist der Mensch noch lange nicht überflüssig. Tatsächlich hat sich das Konzept des *Human-in-the-Loop* zu einem wichtigen Eckpfeiler in vielen KI- bzw. ML-Workflows entwickelt. Die Einbindung des Menschen trägt wesentlich zur Qualitätssicherung von sensiblen und regulierten Modellen in der Produktion bei.

Amazon Augmented AI (Amazon A2I) ist ein vollständig verwalteter Service zur Entwicklung von Human-in-the-Loop-Workflows, die eine übersichtliche Benutzeroberfläche, eine rollenbasierte Zugriffskontrolle mit *AWS Identity and Access*

Management (IAM) und eine skalierbare Datenspeicherung unter Verwendung von S3 umfassen. Amazon A2I ist in zahlreiche Amazon-Services integriert, darunter Amazon Rekognition für die Handhabung von Medieninhalten (Content-Moderation) und Amazon Textract für die Extraktion von Formulardaten. Des Weiteren können wir Amazon A2I mit Amazon SageMaker und jedem unserer benutzerdefinierten ML-Modelle verwenden. Wir werden uns in Kapitel 10 eingehender mit Human-in-the-Loop-Workflows befassen.

Best Practices für MLOps

Der Bereich *Machine Learning Operations* (MLOps) hat sich in den letzten zehn Jahren herausgebildet, um den einzigartigen Herausforderungen beim Betrieb von KI- und ML-basierten Systemen zu begegnen, die aus der Kombination von Software und Daten resultieren. Mithilfe von MLOps entwickeln wir die End-to-End-Architektur für ein automatisiertes Modelltraining, das Hosten von Modellen und die Überwachung der Pipeline. Indem wir von Beginn an eine vollständige MLOps-Strategie verfolgen, bauen wir Fachwissen auf, reduzieren die Wahrscheinlichkeit menschlicher Fehler, verringern das Risiko unseres Projekts und gewinnen Zeit, um uns auf die eigentlichen Herausforderungen der Data Science zu konzentrieren.

MLOps hat drei verschiedene Entwicklungsphasen durchlaufen:

MLOps 1.0
: Modelle manuell entwickeln, trainieren, optimieren und deployen.

MLOps 2.0
: Modell-Pipelines manuell erstellen und orchestrieren.

MLOps 3.0
: Pipelines werden automatisch ausgeführt, wenn neue Daten eintreffen oder der Code durch deterministische Auslöser (Trigger) wie GitOps verändert wird oder wenn die Leistung der Modelle aufgrund statistischer Auslöser wie Drift, Bias (Verzerrung) und Abweichungen in der Erklärbarkeit nachlässt.

AWS und Amazon SageMaker Pipelines unterstützen die komplette MLOps-Strategie, einschließlich des automatischen Pipeline-Retrainings mit sowohl vorab bestimmten bzw. deterministischen GitOps-Triggern als auch statistisch basierten Triggern infolge einer Drift der Daten, eines Bias des Modells oder Veränderungen in der Erklärbarkeit. In den Kapiteln 5, 6, 7 und 9 werden wir uns eingehend mit statistischer Drift, statistischem Bias (bzw. statistischer Verzerrung) und Erklärbarkeit (*Explainability*) beschäftigen. Außerdem implementieren wir kontinuierliche (*Continuous*) und automatisierte Pipelines in Kapitel 10 mit verschiedenen Pipeline-Orchestrierungs- und Automatisierungsoptionen, darunter SageMaker Pipelines, AWS Step Functions, Apache Airflow, Kubeflow und andere Optionen, einschließlich Human-in-the-Loop-Workflows. Lassen Sie uns nun einige Best Practices für Operational Excellence, Sicherheit, Zuverlässigkeit, Leistungseffizienz und Kostenoptimierung von MLOps besprechen.

Operational Excellence

Im Folgenden finden Sie einige Best Practices in Bezug auf Machine Learning, die uns helfen, Data-Science-Projekte erfolgreich in der Cloud aufzubauen:

Datenqualitätsprüfungen.
: Da alle unsere ML-Projekte mit Daten beginnen, sollten Sie sicherstellen, dass Sie Zugang zu hochwertigen Datensätzen haben und wiederholt Prüfungen der Datenqualität durchführen können. Eine unzureichende Datenqualität führt häufig dazu, dass Projekte scheitern. Behalten Sie diese Probleme schon früh in Ihrer Pipeline im Auge.

Fangen Sie einfach an und verwenden Sie bestehende Lösungen wieder.
: Beginnen Sie mit der einfachsten Lösung, denn es gibt keinen Grund, das Rad neu zu erfinden, wenn es nicht zwingend erforderlich ist. Wahrscheinlich gibt es bereits einen KI-Dienst, der unsere Aufgabe bewältigen kann. Greifen Sie auf verwaltete Dienste wie Amazon SageMaker zurück, die über eine Vielzahl integrierter Algorithmen und vortrainierter Modelle verfügen.

Legen Sie Gütemaße bzw. Leistungsmetriken für das Modell fest.
: Ordnen Sie die Leistungsmetriken des Modells den Geschäftszielen zu und überwachen Sie diese Maße kontinuierlich. Wir sollten eine Strategie entwickeln, um bei nachlassender Leistung Modelle für unzureichend zu erklären und neu zu trainieren.

Tracken und versionieren Sie alles.
: Tracken Sie die Modellentwicklung mithilfe von Experimenten und dokumentieren Sie den vollständigen Verlauf (*Lineage*) in nachvollziehbarer Weise. Sie sollten ebenfalls die Datensätze, den Code für die Feature Transformation, die Hyperparameter und die trainierten Modelle versionieren.

Wählen Sie eine geeignete Hardware für das Modelltraining und den Betrieb des Modells aus.
: In vielen Fällen stellt das Training des Modells andere Anforderungen an die Infrastruktur als der Betrieb des Modells zur Erstellung von Vorhersagen. Wählen Sie für die einzelnen Phasen die entsprechenden Ressourcen aus.

Überwachen Sie die im Einsatz befindlichen Modelle fortlaufend.
: Erkennen Sie eine Drift in den Daten oder im Modell und ergreifen Sie geeignete Maßnahmen, wie z. B. das Modell neu zu trainieren (Retraining).

Automatisieren Sie ML-Workflows.
: Bauen Sie konsistente, automatisierte Pipelines auf, um menschliche Fehler zu reduzieren und Zeit für die eigentlichen Kernaufgaben zu gewinnen. Die Pipelines können Schritte umfassen, in denen die Modelle von Menschen genehmigt werden müssen, bevor sie in die Produktion überführt werden.

Sicherheit

Die Verantwortung für Sicherheit und Compliance liegt sowohl bei AWS als auch beim Kunden. AWS sorgt für die Sicherheit »der« Cloud, während der Kunde für die Sicherheit »in der« Cloud verantwortlich ist.

Die häufigsten Sicherheitsüberlegungen für den Aufbau sicherer Data-Science-Projekte in der Cloud betreffen die Bereiche Zugriffsverwaltung, Isolierung von Rechnern und Netzwerken, Verschlüsselung, Governance und Auditierbarkeit.

Wir benötigen umfassende Sicherheits- und Zugriffskontrollfunktionen für unsere Daten. Dementsprechend gilt es, den Zugriff auf Aufträge bzw. Jobs wie das Labeln von Daten, auf Skripte, die der Datenverarbeitung dienen, auf Modelle, Endpoints für die Inferenz oder auch auf Jobs für Batch-Vorhersagen zu beschränken.

Außerdem sollten wir eine Daten-Governance-Strategie verfolgen, die die Integrität, Sicherheit und Verfügbarkeit unserer Datensätze gewährleistet. Implementieren und erzwingen Sie eine Datenabfolge, die die auf unsere Trainingsdaten angewandten Datentransformationen überwacht und trackt. Stellen Sie sicher, dass die Daten im Ruhezustand und bei der Übertragung verschlüsselt sind. Des Weiteren sollten Sie bei Bedarf die Einhaltung gesetzlicher Vorschriften gewährleisten.

In Kapitel 12 werden wir einige Best Practices für den Aufbau sicherer Data-Science- und Machine-Learning-Anwendungen auf AWS noch genauer erörtern.

Reliabilität

Der Begriff *Reliabilität* bzw. Zuverlässigkeit eines Systems beschreibt seine Fähigkeit, Störungen bzw. Unterbrechungen der Infrastruktur oder des Diensts auszugleichen, dynamisch Rechenressourcen zu beziehen, um der Nachfrage gerecht zu werden, und Störungen wie Fehlkonfigurationen oder vorübergehende Netzwerkprobleme abzufedern.

Wir sollten das Tracking von Änderungen und die Versionskontrolle für unsere Trainingsdaten automatisieren. Auf diese Weise können wir im Fall eines Fehlers exakt die gleiche Version eines Modells neu erstellen. Wir werden das Modell einmal erstellen und anschließend die Modellartefakte verwenden, um das Modell in mehreren AWS-Konten und -Umgebungen bereitzustellen.

Leistungseffizienz

Die *Leistungseffizienz* (engl. *Performance Efficiency*) bezieht sich auf die effiziente Nutzung von Computerressourcen zur Erfüllung des Bedarfs und auf die Frage, wie diese Effizienz aufrechterhalten werden kann, wenn sich der Bedarf ändert oder Technologien weiterentwickelt werden.

Wir sollten die für unseren Machine-Learning-Workload geeignete Rechenleistung verwenden. So können wir zum Beispiel GPU-basierte Instanzen nutzen, um Deep-

Learning-Modelle effizienter zu trainieren, indem wir eine längere Warteschlangentiefe, höhere arithmetische Logikeinheiten oder mehr Register verwenden.

Machen Sie sich mit den Leistungsanforderungen der Modelle in Bezug auf Latenz und Netzwerkbandbreite vertraut und deployen Sie jedes Modell bei Bedarf möglichst nah am Kunden. Es gibt Situationen, in denen wir unsere Modelle direkt vor Ort einsetzen möchten, um die Leistung zu verbessern oder die Datenschutzbestimmungen einzuhalten. Mit »direkt vor Ort« ist gemeint, dass das Modell auf dem Gerät selbst ausgeführt wird und die Vorhersagen somit lokal ermittelt werden. Außerdem möchten wir die wichtigsten Leistungsmetriken unseres Modells kontinuierlich überwachen, um frühzeitig Veränderungen der Leistung zu erkennen.

Optimierung der Kosten

Wir können die anfallenden Kosten optimieren bzw. minimieren, indem wir verschiedene Preisoptionen für unsere Amazon-EC2-Instanzen nutzen. Zum Beispiel bieten Sparpläne, sogenannte *Savings Plans*, erhebliche Einsparungen im Vergleich zu den Preisen für On-Demand-Instanzen. Im Gegenzug verpflichten Sie sich, eine bestimmte Menge an Rechenleistung für eine bestimmte Zeit zu nutzen. Savings Plans sind eine gute Wahl, wenn Ihnen die Arbeitslasten bekannt sind oder sich nicht ändern, wie z. B. bei festen/stabilen Inferenzarbeitslasten.

Bei On-Demand-Instanzen zahlen wir für die Rechenkapazität stundenweise oder sekundenweise, je nachdem, welche Instanzen wir verwenden. On-Demand-Instanzen eignen sich am besten für neue oder zustandsbehaftete Workloads mit hohem Bedarf, wie z. B. kurzfristige Aufträge im Rahmen des Modelltrainings.

Schließlich können wir mit Amazon-EC2-Spot-Instanzen freie Amazon-EC2-Rechenkapazitäten zu einem Preis abrufen, der bis zu 90 % unter dem On-Demand-Preis liegt. Spot-Instanzen können flexible, fehlertolerante Arbeitslasten abdecken, wie z. B. Modelltrainingsaufträge, die nicht zeitkritisch sind. Abbildung 1-2 veranschaulicht die durch die Savings Plans, On-Demand- und Spot-Instanzen resultierende Kombination.

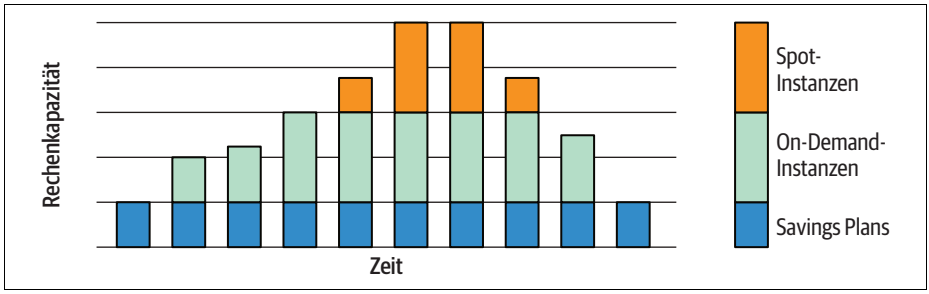


Abbildung 1-2: Optimieren Sie die Kosten, indem Sie eine Kombination aus Savings Plans, On-Demand- und Spot-Instanzen wählen.

Bei vielen der verwalteten Services können wir von dem Modell »Zahlen Sie nur für das, was Sie nutzen« profitieren. Bei Amazon SageMaker zahlen wir zum Beispiel lediglich für die Zeit, in der unser Modell trainiert wird, oder auch nur für die Zeit, in der wir unsere automatische Modelloptimierung durchführen. Beginnen Sie bei der Entwicklung von Modellen mit kleineren Datensätzen, um schneller und sparsamer iterieren zu können. Sobald wir ein gut funktionierendes Modell haben, können wir das Training auf den gesamten Datensatz ausweiten. Ein weiterer wichtiger Aspekt ist die Wahl der geeigneten Größe der Instanzen für das Training und Hosting von Modellen.

In vielen Fällen profitiert das Modelltraining von der GPU-Beschleunigung. Jedoch benötigt die Modellinferenz möglicherweise nicht die gleiche höhere Rechenleistung. Tatsächlich handelt es sich bei den meisten Machine-Learning-Workloads um Vorhersagen. Während das Trainieren des Modells mehrere Stunden oder Tage dauern kann, läuft das eingesetzte Modell wahrscheinlich 24 Stunden am Tag und sieben Tage die Woche auf Tausenden von Servern für Vorhersagen, die Millionen von Kunden unterstützen. Wir sollten entscheiden, ob unser Anwendungsfall einen 24 × 7-Echtzeit-Endpoint oder eine Batch-Vorhersage (*Batch Transformation*) auf Spot-Instanzen am späten Abend erfordert.

Amazons KI-Services und AutoML mit Amazon SageMaker

Wie wir wissen, umfassen datenwissenschaftliche Projekte viele komplexe, multidisziplinäre und iterative Schritte. Wir benötigen Zugang zu einer ML-Entwicklungsumgebung, die die Modellprototyping-Phase unterstützt und gleichzeitig einen reibungslosen Übergang zur Vorbereitung unseres Modells auf die Produktion ermöglicht. Wir werden wahrscheinlich mit verschiedenen ML-Frameworks und -Algorithmen experimentieren und einen benutzerdefinierten Code für das Modelltraining und die Inferenz entwickeln wollen.

In anderen Fällen möchten wir vielleicht einfach nur ein sofort verfügbares, vorab trainiertes Modell verwenden, um eine einfache Problemstellung zu lösen. Oder wir möchten AutoML-Techniken nutzen, um eine erste Ausgangsbasis für unser Projekt zu schaffen. AWS bietet eine breite Palette an Diensten und Funktionen für jedes Szenario. Abbildung 1-3 zeigt den gesamten KI- und ML-Stack von Amazon, einschließlich der AI-Services und Amazon SageMaker Autopilot für AutoML.

Amazons KI-Services

Für viele gängige Anwendungsfälle wie personalisierte Produktempfehlungen, Inhaltsmoderation oder Bedarfsprognosen können wir auch die verwalteten KI-Services von Amazon mit der Option des Feintunings auf unsere eigenen Datensätze anwenden. Wir können diese »1-Click«-KI-Services über einfache API-Auf-