Fraunhofer
IAIS

FRAUNHOFER-INSTITUT FÜR INTELLIGENTE ANALYSE- UND INFORMATIONSSYSTEME IAIS

ZUKUNFTSSICHERE LÖSUNGEN FÜR MASCHINELLES LERNEN

MACHINE LEARNING OPERATIONS (MLOPS) – PROZESSE FÜR ENTWICKLUNG, INTEGRATION UND BETRIEB

Fraunhofer
IAIS

FRAUNHOFER-INSTITUT FÜR INTELLIGENTE
ANALYSE- UND INFORMATIONSSYSTEME IAIS

ZUKUNFTSSICHERE LÖSUNGEN FÜR MASCHINELLES LERNEN

MACHINE LEARNING OPERATIONS (MLOPS) – PROZESSE FÜR ENTWICKLUNG, INTEGRATION UND BETRIEB

Autoren

Niklas Beck
Claudio Martens
Karl-Heinz Sylla
Dr. Dennis Wegener
Alexander Zimmermann

DIE HERAUSGEBER

Das Fraunhofer IAIS

Als Teil der größten Organisation für anwendungsorientierte Forschung in Europa ist das Fraunhofer-Institut für Intelligente Analyse- und Informationssysteme IAIS mit Sitz in Sankt Augustin bei Bonn eines der führenden Wissenschaftsinstitute auf den Gebieten Künstliche Intelligenz, Maschinelles Lernen und Big Data in Deutschland und Europa. Mit seinen rund 300 Mitarbeitenden unterstützt das Institut Unternehmen bei der Optimierung von Produkten, Dienstleistungen, Prozessen und Strukturen sowie bei der Entwicklung neuer digitaler Geschäftsmodelle. Damit gestaltet das Fraunhofer IAIS die digitale Transformation unserer Arbeits- und Lebenswelt.

www.iais.fraunhofer.de

Das KI-Lernlabor

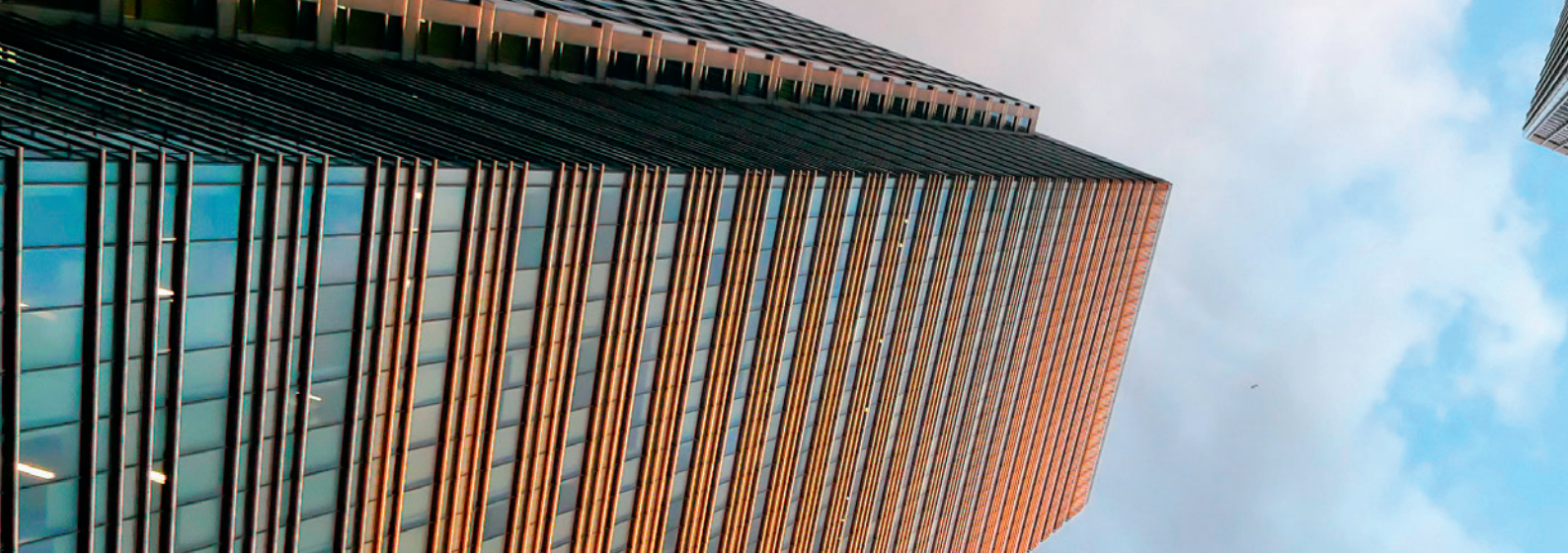
Das KI-Lernlabor unterstützt Unternehmen dabei, in die Künstliche Intelligenz und das Maschinelle Lernen einzusteigen und setzt dazu auf den Dreiklang aus Forschung, Schulung und Praxis. In Research Labs wird innovative Forschung betrieben, im Rahmen der angebotenen Schulungen werden Mitarbeitende qualifiziert und im Innovation Campus können konkrete Anwendungen kennengelernt und entwickelt werden.

Gefördert wird der Aufbau des KI-Lernlabors durch Mittel aus dem Wettbewerb für KI-Labore des Bundesministeriums für Bildung und Forschung. Beteiligt sind die Fraunhofer-Institute für Intelligente Analyse- und Informationssysteme IAIS, für Algorithmen und Wissenschaftliches Rechnen SCAI, die Fraunhofer-Personenzertifizierungsstelle am Fraunhofer-Institut für Angewandte Informationstechnik FIT sowie die Fraunhofer Academy.

www.ki-lernlabor.de

Diese Publikation wurde durch das Bundesministerium für Bildung und Forschung (BMBF) gefördert.

INHALT



1 MANAGEMENT SUMMARY

Maschinelles Lernen (ML) ist ein integraler Bestandteil von Künstlicher Intelligenz (KI). Beide Technologien spielen Schlüsselrollen für die Digitalisierung in Wirtschaft, Wissenschaft und Gesellschaft. In einigen Unternehmen ist Maschinelles Lernen heute schon im Einsatz und trägt zur Wertschöpfung bei. Diese ML-Pioniere sind nicht nur Technologiekonzerne und Start-ups, sondern auch etablierte Unternehmen aus unterschiedlichen Branchen. So existieren ML-Lösungen, die zu einer ressourceneffizienteren Produktion beitragen, die Logistikplanung erleichtern oder in Vertrieb und Marketing dabei helfen, Kundenwünsche besser zu verstehen.

In anderen Unternehmen sind ML-Lösungen noch vor allem Gegenstand von Forschung und Entwicklung. Der Übergang aus dem forschungsorientierten Nischendasein in das strukturierte Tagesgeschäft eines Unternehmens bedarf auf Maschinelles Lernen zugeschnittene, technische und organisatorische Konzepte. In diesem Whitepaper möchten wir dazu Orientierungshilfen geben: Diese richten sich sowohl an das Management und Unternehmensstrategen als auch an Fachleute, die für IT-Lösungen und digitale Infrastruktur zuständig sind.

Denn uns ist eine ganzheitliche Sichtweise wichtig: Um Maschinelles Lernen in die Anwendung zu überführen, ist es neben den rein technischen Lösungen essentiell, auch auf der organisatorischen Ebene frühzeitig Expert*innen aus den Fachabteilungen in die Konzept- und Use-Case-Entwicklung einzubinden. Die Vor- und Nachteile von ML-Lösungen müssen auf allen fachlichen und technischen Ebenen des Unternehmens bekannt sein und verstanden werden – beginnend bei der Entwicklung bis hin zum produktiven Einsatz.

Mit diesem Whitepaper möchten wir zur Zukunftssicherheit von ML-Lösungen in Unternehmen beitragen. Die hier beschriebenen Prozesse, Verfahren und Werkzeuge sollen dazu dienen, Fehler, die in der Vergangenheit bei der Einführung neuer Softwaretechnologien gemacht wurden, zu vermeiden. Aus unserer Sicht muss verhindert werden, dass Lösungen entstehen, die in den Kern von umfangreichen Anwendungen integriert sind und nur durch hoch qualifizierte Expert*innen verstanden, gewartet und weiterentwickelt werden können. Bis heute entsteht beispielsweise Banken ein enormer Aufwand dadurch, dass nicht nachhaltig integrierte Lösungen gewartet oder abgelöst werden müssen, ohne den operativen Betrieb zu behindern.

Nachhaltige und zukunftssichere Entwicklung, Integration und Betrieb von ML-Lösungen sind nicht allein durch technische Lösungen erreichbar, sondern vor allem durch ein frühzeitiges Schaffen von Bewusstsein bei allen Mitarbeitenden für den Mehrwert von ML-Technologien, aber auch für die dahinterstehenden speziellen Anforderungen und Herausforderungen. Unser Ansatz ist es daher, ML-Lösungen modular und nachvollziehbar zu integrieren, damit sie ohne großen Aufwand gewartet und später bei Bedarf aktualisiert oder ausgetauscht werden können.

Bisher werden auch für ML-Lösungen etablierte Prozesse der Softwareentwicklung verwendet. Aufgrund ihres Aufbaus und ihres Entwicklungszyklus stellen ML-Lösungen jedoch in vielen Bereichen der Entwicklung, der Qualitätssicherung und des Betriebs Anforderungen, die durch bisherige Methoden nicht oder nur teilweise abgedeckt werden können. So ist das für Data-Science-Projekte etablierte Vorgehen CRISP-DM (Cross Industry Standard Process for Data Mining) primär für

die Modellentwicklung gedacht, für den Übergang in die Nutzung sind jedoch bisher nur wenige Konzepte vorhanden.

In diesem Whitepaper beschreiben wir das Prozessschema »MLOps« (Machine Learning Operations). Dabei handelt es sich um eine Erweiterung und Anpassung des für die Softwareentwicklung bewährten Prozesses »DevOps« (Development Operations). MLOps integriert Elemente von DevOps und CRISP-DM und erweitert diese um ML-spezifische Charakteristika. Damit bieten wir Leitlinien für die zielgenaue Entwicklung, die schnelle Integration und den gesicherten Betrieb von ML-Technologien.

Der Fokus von MLOps liegt auf einer technisch hoch automatisierten Vorgehensweise, die ML-Lösungen nachhaltig und verlässlich zur Verfügung stellt. Außerdem lassen sich aktuelle Virtualisierungstechnologien, wie beispielsweise Docker-Container, nutzen, um eine durchlässige Architektur zu entwickeln, die die Überführung von ML-Lösungen aus der Entwicklung hin zum operativen Betrieb unterstützt.

Die Integration, Skalierung und Überwachung von ML-Lösungen für unterschiedliche Infrastrukturen, von einzelnen Arbeitsplätzen bis hin zu Cloud-Implementierungen, wird von einer Vielzahl von Plattformen unterstützt. Wir geben einen Einblick in heute verfügbare Werkzeuge, die die Umsetzung der MLOps-Prozesse erleichtern. Neben der Technologie und den Werkzeugen spielt die Kommunikation eine zentrale Rolle für die erfolgreiche Implementierung von ML-Lösungen. MLOps ist ein agiler Prozess mit einer starken und häufigen Rückkopplung zwischen Anwender*innen und Entwickler*innen innerhalb von Data-Science-Projekten.

Dieses Whitepaper ist am Fraunhofer IAIS in Kooperation mit dem KI-Lernlabor entstanden. Eingeflossen sind unsere Erfahrungen aus vielen Projekten zum Maschinellen Lernen für ganz unterschiedliche Unternehmen, Branchen und Anwendungsszenarien.

Wir sind gespannt, wie Sie die Technologie einsetzen und laden Sie ein, mit uns ins Gespräch zu kommen!

2 SPEZIELLE CHARAKTERISTIKA VON ML-LÖSUNGEN

Anwendungen des Maschinellen Lernens gewinnen in vielen Bereichen mehr und mehr an Bedeutung. Konstruktion und Betrieb von ML-Lösungen sind jedoch sehr komplex und bergen einige Herausforderungen.

Entwicklung und Umsetzung von ML-Lösungen werden durch das in Abbildung 1 dargestellte Vorgehensmodell CRISP-DM (Cross Industry Standard Process for Data Mining) beschrieben. Dieses ist unterteilt in die Phasen Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation und Deployment. Typischerweise werden dabei die Phasen Business Understanding und Deployment in einer Produktiv-Systemumgebung durchgeführt, während die Phasen von Data Understanding bis Evaluation in einer Analyseumgebung durchgeführt werden.[1] Während CRISP-DM das Vorgehen hinsichtlich des Data-Science-Anteils gut beschreibt, deckt es jedoch nicht dezidiert die Konstruktion und den Betrieb von ML-Softwareanwendungen ab. Es existieren spezielle Herausforderungen beim Erstellen und Betreiben von ML-Lösungen. Diese resultieren aus der Komplexität und Unwägbarkeit einiger Schritte zur Konstruktion von ML-Workflows.[2] Abstrakt wird die Konstruktion von ML-Workflows oft mit folgenden Schritten beschrieben:[2] [3]

1) Identifizieren eines sinnvollen ML-Ansatzes (z. B. basierend auf Literatur)
2) Erstellen eines initialen Prototypen / initiales Sammeln und Vorverarbeiten von Daten
3) Iteration bei der Modellerstellung (Training, Experiment, Evaluation)
4) Deployment und Monitoring

Abbildung 1: Das CRISP-DM Vorgehensmodell

Zusätzliche Herausforderungen birgt die Tatsache, dass das Trainieren von Modellen oft in einer Analyseumgebung erfolgt, die sich von der Produktivumgebung, in der das Modell später angewendet wird, unterscheidet. Dies beruht zum einen darauf, dass für das Trainieren von Modellen andere Anforderungen an die zur Verfügung stehenden Computing-Ressourcen gestellt werden als für die Modellanwendung. Das Trainieren ist typischerweise sehr viel ressourcenintensiver und benötigt zum Teil spezielle

Hardware wie beispielsweise Grafikprozessoren (GPUs). Zum anderen ist es oft der Fall, dass sich die Prozesse, Tools und Frameworks, die in der Analyseumgebung nötig sind, sehr von denen der Produktivumgebung unterscheiden.

Bei der Entwicklung von Software gibt es viele Prinzipien zur Automation und zur Qualitätssicherung: beispielsweise das Testen sowie das Prinzip des CI/CD (Continuous Integration/ Continuous Delivery bzw. Continuous Deployment). Diese lassen sich jedoch nicht 1:1 auf die Entwicklung von ML-Software übertragen. Insbesondere kann es sein, dass ein ML-Workflow getestet und ohne Fehler ausgeführt werden kann, die Ergebnisse aber trotzdem fehlerhaft oder unbrauchbar sind.[3]

Aus der Softwareentwicklung ist das Prozessschema DevOps bekannt, dessen Name aus der Verknüpfung der Begriffe Development und Operations abgeleitet ist. Analog dazu wird aus den Begriffen Machine Learning und Operations das für ML-Lösungen angepasste und erweiterte Prozessschema MLOps.

Für viele Anwendungsfälle werden mehrere ML-Komponenten, wie beispielsweise Dateneingabe, Modelle zur Analyse und Datenausgabemodule zu mehrstufigen Prozessketten, sogenannten Pipelines, verknüpft. Der Grad der Automatisierung von ML-Prozessen und Pipelines kann in verschiedenen MLOps Stufen beschrieben werden:[4]

- Stufe 0: Der Prozess zum Erstellen und Bereitstellen von ML-Modellen erfolgt vollständig manuell.
- Stufe 1: Continuous Training Pipeline – durch Automatisierung wird ein kontinuierliches Training der Modelle ermöglicht.
- Stufe 2: CI/CD Pipeline Automatisierung – Komponenten der ML-Pipeline können automatisiert erstellt, getestet und bereitgestellt werden.

Derzeit findet ein Wandel statt, weg von der Bereitstellung eines ML-Modells für die Anwendung in einer Produktionsumgebung hin zur Bereitstellung einer ML-Pipeline, mit der das erneute Trainieren und Bereitstellen neuer Modelle automatisiert werden kann.[4]

3 ORGANISATION, WORKFLOWS & PROZESSE

Die im vorigen Kapitel beschriebenen Charakteristika von ML-Lösungen können und müssen in mehreren Dimensionen behandelt werden. So reicht es auf der einen Seite nicht aus, sich auf technische Werkzeuge und Neuerungen zu verlassen, die versprechen, den administrativen Aufwand zu reduzieren. Auf der anderen Seite verliert man durch falsche Projektorganisation schnell die Flexibilität und Kreativität, die bei der Erstellung von ML-Lösungen notwendig ist.

Wir haben bereits in mehreren unserer Kundenprojekte beobachtet, dass sich Data-Science-Teams oft hauptsächlich aus IT- und technikaffinen Mitarbeiter*innen rekrutieren, die organisatorischen und unternehmensstrategischen Ebenen jedoch häufig nicht genügend einbezogen werden. Dies kann dazu führen, dass ML-Entwicklungsteams in einzelnen Demonstratoren im kleinen Maßstab den Mehrwert von ML-Ansätzen zeigen, anschließend jedoch durch Ihren Erfolg und damit einhergehende erhöhte Anfragen in die Grenzbereiche der technischen und organisatorischen Skalierbarkeit geraten.

Das Prozessschema MLOps begegnet diesen Herausforderungen durch eine aufeinander abgestimmte Kombination aus Methoden und Werkzeugen, vom Projektmanagement bis zur technischen Architektur von ML-Lösungen.

Die Arbeiten im Rahmen eines Projekts zur Erstellung einer ML-Lösung lassen sich in die folgenden Kategorien einteilen: "Entwicklung und Weiterentwicklung einer Anwendung", "Betrieb der Anwendung" und "Projektorganisation". Die Arbeiten werden durch Werkzeuge und Frameworks unterstützt, die selbst in Arbeitsumgebungen integriert sind. Der Bearbeitungsstand eines Projekts ist in Verzeichnissen (Repositories) gespeichert und beschrieben. Ein Vorteil einer integrierten Arbeitsumgebung ist, dass man Arbeitsschritte und Abfolgen von Arbeitsschritten (Workflows) automatisieren kann. In welchem Umfang eine Automatisierung durch Prozeduren oder Skripte möglich und sinnvoll ist, hängt von der Aufgabe und von der Systematisierung wiederholbarer Verfahren im Projektablauf ab. Bei der Festlegung von Automatisierungsschritten sollten nach Möglichkeit Erfahrungen aus ähnlichen Projekten eingehen.

Abbildung 2: In einem ML-Projekt sollten Entwicklung und Betrieb einer Anwendung ineinandergreifen.

Abbildung 3: Der ML-Engineering-Prozess verbindet technische und organisatorische Elemente.

Der MLOps-Ablauf wird auch als ML-Engineering-Prozess bezeichnet. Das Ablaufschema erweitert den auf Softwareentwicklung zugeschnittenen DevOps-Prozessansatz um ML-spezifische Anteile, die auf dem CRISP-DM-Vorgehensmodell basieren. Der MLOps-Prozess ist unterteilt in organisatorisch orientierte (blaue Sechsecke) und eher technisch orientierte (orangefarbene Sechsecke) Elemente. Die sechs Hauptschritte, dargestellt in Abbildung 3, ergeben einen kontinuierlichen Prozess, dessen technisch geprägte Schritte soweit möglich automatisiert ablaufen.

Wie von DevOps bekannt, sind die Prozessschritte eng, oft sogar automatisiert, miteinander verknüpft. Gerade bei der Automatisierung muss die Integration in die Unternehmensprozesse exakt definiert und abgestimmt sein. Dies führt dazu, dass bei einer kontinuierlichen und eng aufeinander abgestimmten Abfolge von Schritten noch eingehender auf Rollen, Verantwortlichkeiten und bereichsübergreifende Kommunikation geachtet werden muss. Es kann also von einem wichtigen siebten Hauptbereich von MLOps gesprochen werden, der die Kommunikation, eine einheitliche Datenhaltung für Projekt- und Prozessinformation sowie die Informationsübergänge zwischen den anderen sechs Hauptbereichen umfasst. In Abbildung 3 ist dies als zentraler Projektdatenbereich ausgewiesen. Wie bereits erläutert, sind für die organisatorisch orientierten Bereiche aus dem Standard-Software-Engineering bekannte Werkzeuge und Methoden vorhanden, die sich gut für ML-spezifische Aufgaben anpassen lassen.

Grundsätzlich basieren die Arbeiten in den Bereichen der Anforderungsanalyse, der Planung und des Designs sowie die gemeinsame Kommunikation auf abgestimmten Diskussionsgrundlagen, wie beispielsweise Architekturbilder, und schließen auch regelmäßige und formale Abstimmungen ein. Dies unterscheidet sich bis auf den ML-spezifischen fachlichen Inhalt nicht von klassischen oder agilen Projektorganisationen.

Die weiteren Bereiche von MLOps sind eher technisch orientiert und umfassen den Prozess der Entwicklung von ML-Lösungen, über das Testen und die Abnahme ebendieser bis hin zur Bereitstellung und dem Betrieb der ML-Systeme. Die Entwicklungsarbeiten sind durch jederzeit mögliche Rückmeldung, Überarbeitung und Wiederholung eng miteinander verzahnt und nach agilen Mustern wie

Scrum und Kanban organsiert. Das typisch experimentelle Vorgehen bei der Modellentwicklung von ML-Lösungen ist ein iterativer Optimierungsprozess, der an der technischen Validierung und vor allem an der fachlichen Plausibilität und Güte der Modellbildung orientiert ist. Wichtig ist, dass Anforderungen und Rückmeldungen von fachlicher Seite, also beispielsweise aus Management, Unternehmensstrategie, Organisationsentwicklung, Vertrieb und Marketing oder der Produktion berücksichtigt werden können. Dazu muss in der Projektorganisation dafür gesorgt werden, dass die Fachseite gut in die agile Arbeitsplanung sowie in die Zielsetzung und Bewertung der Modelle einbezogen wird.

Generell folgen die Projekte zur Erstellung von ML-Lösungen einer agilen Arbeitsorganisation. Jedoch sind die Arbeiten zur Auslieferung und zum Betrieb von Anwendungsversionen auf technische Ziele wie Robustheit, Skalierbarkeit, geringe Latenz und Sicherheit ausgerichtet. Diese Arbeitsschritte folgen einem strikten Plan, sind sequentiell verbunden und automatisierbar. Mit den Fachabteilungen wird die Inbetriebnahme neuer Versionen und das Monitoring im Betrieb organisatorisch abgestimmt.

Ein zentraler Ansatz der Anwendungsentwicklung ist die Trennung von Entwicklungs-, Test- und Produktionsumgebung. Organisatorisch betrachtet ist hier die klare Definition von Rollen und Verantwortlichkeiten während der Entwicklung essentiell, die sich von der klassischen Softwareentwicklung kaum unterscheiden. Der Hauptunterschied von MLOps zum klassischen Vorgehen liegt also in den technischen Prozessschritten.

In der Entwicklungsumgebung gestalten sich die Freiheiten zur Entwicklung von Modellen und Algorithmen relativ groß und können in einer agilen Art und Weise mit den Anforderungen der Fachseite abgestimmt werden. Die Arbeitsumgebungen können sich hier noch stark unterscheiden, beispielsweise durch den Einsatz ML-spezifischer Werkzeuge, Bibliotheken oder Spezialhardware. Dies alles dient der Unterstützung der Kreativität und erlaubt ein schnelles und adaptives Entwicklungsvorgehen. Im besten Fall wird dieser Bereich durch Leitplanken umfasst, die je nach Unternehmen Vorgaben zu Programmiersprachen, Entwicklungswerkzeugen oder Programmierrichtlinien machen können. Spätestens jedoch beim Verlassen der explorativen Phase und Übergang der ML-Lösung in die Projektphase der Verstetigung und gegebenenfalls Implementierung in Form eines Produktes ist eine professionelle Entwicklungsumgebung notwendig. Eine solche Umgebung braucht integrierte Möglichkeiten zum Qualitätsmanagement, eine Anbindung an ein Versionierungssystem und eine Strukturierung des Quellcodes (Codebasis).

Die Testumgebung stellt den ersten zentralen und gemeinsamen Konsolidierungspunkt dar: Die bis dahin entwickelten und implementierten ML-Komponenten werden hier automatisiert zusammengebaut, getestet und als Ergebnisse (Artefakte) mit einer Versionsnummer versehen zur Verfügung gestellt. Dazu können auch Continuous Integration (CI) Werkzeuge, wie beispielsweise Build-Server und Image-Repositories, zum Einsatz kommen.

Außerdem können Virtualisierungstechnologien verwendet werden, zum Beispiel Container auf Basis von Docker Images. Vorteil dieser Virtualisierung ist es, dass unabhängig von der volatilen Entwicklungsumgebung in einer gemeinsamen Zielumgebung gebaut und getestet werden kann. Zudem wird so gewährleistet, dass einheitliche, deterministische und wiederverwendbare Bau- und Testumgebungen inklusive kuratierter Testdaten zur Verfügung stehen. Des Weiteren können die Bau- und Testumgebungen auch als Artefakte für Demonstrationszwecke oder den späteren produktiven Betrieb wiederverwendet werden, sollte die Zielumgebung die Virtualisierungstechnologie unterstützen.

Mittels Continuous Deployment (CD) als Ergänzung zur Continuous Integration können automatisiert

verfügbare Artefakte konfiguriert und in entsprechenden Zielumgebungen bereitgestellt werden. Die Zielumgebungen können hierbei verschieden ausgestattet und eingebunden sein: von Test- und Abnahmeumgebungen mit entwicklungsorientiertem Hintergrund bis hin zu Produktivumgebungen, die operativ im Unternehmen eingebunden sind.

Die Produktionsumgebung schlussendlich stellt eine robuste und skalierbare Plattform zur Verfügung, auf der die in der Testumgebung erstellten Artefakte bereitgestellt werden.

An die Produktionsumgebungen werden eine Reihe von Anforderungen gestellt: neben der standardmäßig geforderten Ausfallsicherheit und Erreichbarkeit müssen die ML-Artefakte selbst speziellen Kriterien bezüglich Überprüfbarkeit und Bewertungsfähigkeit genügen. So muss die Güte der Modelle, insbesondere für lernende Systeme, fortwährend messbar und überprüfbar sein. Auf Basis der Ergebnisse und Auswertungen aus dem laufenden Betrieb können Fachleute aus den Unternehmen, die die ML-Lösungen einsetzen, wertvolle Informationen gewinnen, ihre Erfahrung einbringen und bei Bedarf Fehleranalysen tätigen oder Nachbesserungen anregen.

Abbildung 4: ML-Engineering-Prozess mit Kommunikationsflüssen und zentralen Funktionalitäten

4 UMSETZUNG AM FRAUNHOFER IAIS

In diesem Kapitel geben wir einen Einblick in die praktische Umsetzung von Projekten, in denen am Fraunhofer IAIS ML-Lösungen entwickelt und eingesetzt wurden. Wir fassen die Erfahrungen von Data Scientists und ML-Ingenieuren zusammen und geben Best-Practice-Beispiele für eine erfolgreiche Anwendung der MLOps-Prozessschritte. Die beschriebenen Prozesse, Werkzeuge und Strukturen haben sich in Projekten für unterschiedliche Kunden, Branchen und Anwendungsszenarien bewährt.

Anforderungsanalyse, Planung & Design

Das Fraunhofer IAIS setzt für den Großteil seiner ML-Lösungen zur projektübergreifenden Kommunikation und Dokumentation auf die Atlassian Toolsuite. Diese unterstützt eine integrierte Zuordnung (Mapping) zwischen den einzelnen Entwicklungsschritten, Prozessen und Ergebnissen, die in Tabelle 1 dargestellt sind.

Hierdurch kann der komplette MLOps Projektablauf in einer Toolsuite abgedeckt werden. Weiter ist durch das integrierte Mapping gewährleistet, dass für jede ausgelieferte Version eine Rückverfolgbarkeit bis hin zu den Anforderungen automatisiert vorhanden ist. Die Planung und Durchführung der Projekte erfolgt in einem an Scrum oder Kanban orientierten agilen Vorgehen. Dadurch wird eine Projektabwicklung unterstützt, die iterativ fortschreitet und Fachleute auf Kundenseite eng einbinden kann.

Implementierung

In der Phase der Implementierung ist zu unterscheiden zwischen Programmiersprachen, Entwicklungsumgebungen und angewandten ML-Ansätzen. Während die angewandten ML-Ansätze, wie beispielsweise neuronale Netz-Architekturen oder Modellvarianten, der Forschung unterliegen und sehr individuell für jedes Projekt angepasst werden, ist bei der

Schritt	Prozesse und Ergebnisse	Name des Tools (Atlassian Toolsuite)
1	Dokumentierte Anforderungen	Confluence
2	Abgeleitete Entwicklungsaufgaben	Jira
3	Zugehöriger Sourcecode mit Nachverfolgung der eingespielten Versionen (Commit-Historie)	Bitbucket
4	Angestoßene Continuous Integration Workflows	Bamboo
5	Ausgelieferte Ergebnisse (Artefakt-Deployments)	Bamboo

Tabelle 1: Prozesse, Ergebnisse und Werkzeuge für ML-Lösungen

Programmiersprache am Fraunhofer IAIS ein klarer Fokus auf Python gesetzt. Einzelne Projekte mit Java und R sind aufgrund deren charakteristischer Fähigkeiten und einer weiten Verbreitung allerdings auch vorhanden. Die meist genutzten ML-Frameworks sind TensorFlow, PyTorch und Keras, wobei aktuell noch kein Trend zu erkennen ist, welches sich über die Zeit durchsetzt.

Bei den Entwicklungsumgebungen ergibt sich ein gemischtes Bild; dies ist der Arbeitsweise des CRISP-DM Vorgehens geschuldet und wird auch im MLOps-Vorgehen beschrieben. Aufgrund ihrer intuitiven Nutzung und graphischen Oberfläche werden in der explorativen Phase zu Beginn von ML-Projekten zumeist Jupyter Notebooks verwendet. Diese bieten auch eine sehr gute Basis für die Kommunikation mit dem Kunden und schnelle Interaktionen. Vor allem auch im Bereich der Schulungen zum Maschinellen Lernen werden für alle praktischen Übungen solche Notebooks verwendet, da sie einen einfachen und schnellen Zugang zur Thematik erlauben.

Beim Übergang von der explorativen in die entwicklungsgeprägte Phase wird am Fraunhofer IAIS ein strikter Wechsel von Jupyter Notebooks in Entwicklungsumgebungen wie beispielsweise JetBrains PyCharm verlangt. Nur so ist die Entwicklung eines strukturierten und qualitätsgesicherten Codes für die spätere ML-Lösung möglich. In vielen Projekten hat sich gezeigt, dass dieser Wechsel, auch wenn er durch vorhandene Export-Werkzeuge unterstützt wird, teils hohen Aufwand bedeutet. In diesem Schritt muss der Code entlang allgemeiner Entwicklungsprinzipien strukturiert werden. Zudem ist es nötig, Anbindungen an die Datenschicht und Visualisierungskomponenten neu zu gestalten und

gegebenenfalls auch Umgebungssysteme einzubinden. In dieser Phase wird auch ein erfahrener Softwarearchitekt hinzugezogen, um das Gesamtsystem strukturiert entlang der technischen Architektur, die später vorgestellt wird, zu planen. Falls noch nicht geschehen, werden nun auch Versionierungssysteme wie Bitbucket oder GitLab für das Code-Management eingeführt und mit einer Build- und Test-Infrastruktur verbunden.

Continuous Integration & Qualitätsmanagement

Für den Prozessschritt Continuous Integration wird der Build-Server Atlassian Bamboo genutzt, der aus einem Cluster von Build-Agents zur Lastverteilung besteht. Für jedes Projekt kommen definierte Docker-Build- und Test-Umgebungen zum Einsatz. Auf diese Weise kann unabhängig von der Anzahl der eingesetzten Entwickler und deren individuellen Entwicklungsumgebungen und –vorgehen eine zentrale, deterministische Artefakt-Erzeugung gewährleistet werden.

Im Rahmen des Qualitätsmanagements werden während des Build-Prozesses Unit-, Integrations- und End2End-Tests durchgeführt. Zusätzlich übernimmt eine SonarQube-Instanz die Validierung der Code-Qualität unter Einbeziehung von Quality Gates. Als Ergänzung wird derzeit aktiv am frühzeitigen, automatisierten Testen von Jupyter Notebooks geforscht. Da die Jupyter Notebooks bereits in der explorativen Phase von ML-Projekten zum Einsatz kommen und dort einer Versionierung unterliegen, birgt dies den Vorteil, die Qualitätssicherungsmaßnahmen noch früher einsetzen zu lassen und damit einen beschleunigten und effizienteren Übergang zwischen Explorations- und Entwicklungsphase zu ermöglichen.

Als Möglichkeit zum Überwachen (Tracking) und zur Aufzeichnung (Logging) von Experimenten und Trainingsmetriken werden framework-spezifische Tools wie TensorBoard oder generische wie MLflow genutzt. Die trainierten Modelle werden dann typischerweise in Framework-spezifischen, haltbaren (persistenten) Formaten wie beispielsweise Spark, TensorFlow oder PyTorch abgespeichert und versioniert. Auch hierbei bietet MLflow – als Model Repository mit Versionierung – eine Unterstützung.

Die abschließende Bereitstellung der Artefakte, welche sich zumeist als Docker-Images darstellen, erfolgt mittels GitLab, eine an allen Fraunhofer-Instituten eingesetzte Plattform. Falls ein späteres Deployment mittels Docker nicht möglich oder gewünscht ist, hat es sich je nach Anwendung und Zielumgebung als zielführend herausgestellt, die erzeugten Modelle in standardisierte Formate wie ONNX zu konvertieren. Der Verlust an spezifischen Modellinformationen wird hier durch eine sehr hohe Interoperabilität in verschiedenen Laufzeitumgebungen aufgewogen.

Continuous Deployment, Betrieb & Überwachung

Abbildung 5: Technische Schichten-Architektur, die beim Fraunhofer IAIS eingesetzt wird.

Das Deployment, ob manuell oder automatisiert, findet unter Nutzung von Atlassian Bamboo statt. Die Zielumgebungen, soweit von Fraunhofer IAIS betrieben, sind normalerweise Linux-Distributionen mit Docker-Umgebung. Für hochperformante und skalierbare Anwendungen wird ein Kubernetes-Cluster genutzt. Die bereitgestellten Modelle laufen intern teilweise auch in Framework-spezifischen Serving-Umgebungen wie TorchServe und TensorFlowServing.

Das Logging erfolgt entweder individuell per Anwendung direkt in den Zielumgebungen oder zentral per Konnektor in den ELK-Stack. Das Monitoring von ML-Lösungen und die Überwachung der inhärenten Modelle ist essentiell, da sich die Modelle in einem Lebenszyklus befinden und deren Prognosegüte im Allgemeinen über die Zeit abnimmt. Auch hier bieten Frameworks wie MLflow eine gute Unterstützung zur Überwachung der ML-spezifischen Metriken im Betrieb.

Zur Unterstützung von Produktivsystemen und um die genauen Rahmenbedingungen für die ML-Lösung zu definieren, werden Service Level Agreements (SLAs) mit dem Kunden vereinbart. Änderungswünsche und Verbesserungen, die nicht als Bugfixes über die SLAs abgedeckt sind, erfolgen über strukturierte Änderungsanforderungen (Change Requests) und münden in einem neuen MLOps-Zyklus.

Technische Architektur

Um die MLOps-Prozessschritte nahtlos, skalierbar und möglichst unabhängig von spezifischen Hardware- und Infrastrukturanforderungen durchführen zu können, wird beim Fraunhofer IAIS eine virtualisierte, geschichtete Architektur eingesetzt.

Zentraler Bestandteil ist eine Containervirtualisierung mittels Docker. Dies ermöglicht, die zugrundeliegende ML-Lösung in Bezug auf Hardware und Infrastruktur unabhängig von der Zielumgebung zu halten. Ob die ML-Lösung auf einem lokalen PC, einem Unternehmenscluster oder in der Cloud gehostet wird, ist durch diesen Ansatz unerheblich. Die Architektur ermöglicht es den Softwareentwicklern, in abgeschlossenen Umgebungen zu entwickeln, die sich über den gesamten Lebenszyklus der Anwendung erhalten lassen, einfach und automatisiert testbar sind und durch gängige Werkzeuge des CI/CD unterstützt werden. Aufgrund der modularen Bauweise auf Ebene der Frameworks, Tools und Daten lassen sich auch wiederverwendbare und leicht ersetzbare Entwicklungsumgebungen erzeugen, die auf individuelle fachliche Anforderungen zugeschnitten werden können. Dieser Ansatz wird bereits erfolgreich in mehreren Kundenprojekten und dem kompletten Hands-On-Anteil des Data-Scientist-Schulungsprogramms vom Fraunhofer IAIS angewandt.

Eine aktuelle Herausforderung und Forschungsgegenstand sind die effiziente, bereichsübergreifende Bereitstellung und Nutzung von Daten im allgemeinen und großen Modell-Artefakten im Speziellen. Beispielsweise im Bereich Textmining eingesetzte Modelle und neuronale Netze benötigen Trainingsdaten, die schnell einen Umfang von mehreren hundert Gigabyte oder noch mehr erreichen. Eine performante und sichere Übertragung solcher Datenmengen kommt mit einem einfachen Kopieren schnell an seine Grenzen. Hier müssen neue Konzepte zu verteiltem Lernen, Reduktion der Modellgrößen und mandantenorientierter Wiederverwendung von verteilten Datenbeständen entwickelt werden.

5 FAZIT

Maschinelles Lernen als Schlüsseltechnologie für Künstliche Intelligenz gewinnt durch die technologischen und methodischen Fortschritte der letzten Jahre auch im produktiven Umfeld von Firmen an Bedeutung. Die Digitalisierung führt zur Erzeugung von immer mehr Daten und damit auch zu neuen Möglichkeiten für den effektiven Einsatz von Maschinellem Lernen. Das Maschinelle Lernen wird zukünftig mehr und mehr zum festen Bestandteil von Wertschöpfungsketten in Unternehmen.

Eine zukünftige Herausforderung besteht darin, die Potenziale von Maschinellem Lernen vollständig zu heben und im Unternehmenskontext gewinnbringend und zukunftssicher einzusetzen. Des Weiteren muss in vielen Firmen das Maschinelle Lernen zunächst aus seinem Nischendasein in Forschungs- und Entwicklungsabteilungen herausgelöst und strukturiert in den Betrieb des Unternehmens eingegliedert werden. Derzeit existieren vor allem für den operativen Einsatz von ML-Lösungen nur wenige etablierte Methoden und Werkzeuge.

Um ML-Technologien erfolgreich in die Anwendung zu bringen, sind neue Ansätze zu Prozessen und Kommunikationsstrukturen sowie zur Nutzung von Werkzeugen und Technologieplattformen nötig. Denn die gängigen Methoden zur Softwareentwicklung und zur Implementierung sind für ML-Lösungen nur bedingt geeignet. Herausforderungen in Bezug auf das Training von ML-Modellen, die Validierung von Verfahren und den robusten Betrieb muss geeignet begegnet werden.

Dazu haben wir den Prozessansatz MLOps vorgestellt, der bereits etablierte Prozesse aus der Softwareentwicklung aufgreift und für die Konzeption, Entwicklung, Integration und den Betrieb von ML-Lösungen erweitert. Insbesondere der DevOps-Prozessansatz aus der Softwareentwicklung kann mit einigen Anpassungen für den Anwendungsbereich des Maschinellen Lernens erweitert werden. Prozesse, die das Projektmanagement und organisatorische Aspekte von DevOps betreffen, können dabei nahezu 1:1 für MLOps übernommen werden.

Für das Maschinelle Lernen angepasst werden müssen dagegen technische Prozessschritte und der Einsatz von Werkzeugen. Unsere Erfahrung zeigt, dass sich für den professionellen Einsatz von ML-Lösungen ganzheitliche Ansätze wie beispielsweise MLflow bewährt haben. Dies sind skalierbare Plattformen, die den gesamten Lebenszyklus von ML-Lösungen abbilden: von der Entwicklung über die Integration und die Inbetriebnahme einschließlich Qualitätssicherung und kontinuierliche Verbesserung. Zudem profitieren solche Entwicklungsprojekte von Virtualisierungstechnologien und Microservice-Architekturen.

Bei all diesen Schritten unterstützen und begleiten wir Sie. Wir evaluieren bestehende Prozesse, identifizieren Verbesserungspotenzial und entwickeln gemeinsam mit den Expert*innen aus Ihren Fachabteilungen neue Anwendungsszenarien für Maschinelles Lernen. Um Ihre Prozesse zukunftssicher zu gestalten, erstellen wir eine Roadmap, beraten und schulen Ihre Mitarbeiter*innen und implementieren Methoden zur Qualitätssicherung. Schließlich setzen wir unsere Erfahrung ein, um die technischen Entwicklungsschritte durchzuführen und bei Bedarf Infrastruktur anzupassen sowie organisatorische Veränderungen zu begleiten.

6 LITERATURVERZEICHNIS

[1] M. Mock und K.-H. Sylla, »Entwicklung und Deployment von KI-Anwendungen mit Big Data«, OBJEKTspektrum, 2019.

[2] L. Ampil, »Basics of Data Science Product Management: The ML Workflow«, 31. August 2019. [Online]. Available: https://towardsdatascience.com/basics-of-ai-product-management-orchestrating-the-ml-workflow-3601e0ff6baa [Zugriff am 2. Juli 2020].

[3] E. Ameisen, »Building Machine Learning Powered Applications«, O'Reilly Media, Inc., 2020.

[4] »MLOps: Continuous Delivery und Pipelines zur Automatisierung im maschinellen Lernen«, [Online]. Available: https://cloud.google.com/solutions/machine-learning/mlops-continuous-delivery-and-automation-pipelines-in-machine-learning [Zugriff am 2. Juli 2020].

7 IMPRESSUM

Herausgeber
Fraunhofer-Institut für Intelligente Analyse- und Informationssysteme IAIS
Schloss Birlinghoven
53757 Sankt Augustin
Telefon 02241 14-2900
Fax 02241 14-2436
pr@iais.fraunhofer.de
www.iais.fraunhofer.de

Redaktion
Inga Daase
Silke Loh

Grafik
Inga Daase
Maya Schwarzer

Layout
Achim Kapusta
Maya Schwarzer

Bildnachweise
Titelbild: Alisa/stock.adobe.com
Seite 4-5: alice_photo/stock.adobe.com
Seite 6-7: evannovostro/stock.adobe.com
Seite 8-11: unikyluckk/stock.adobe.com
Seite 12-15: artegorov3@gmail/stock.adobe.com
Seite 16-17: AVTG/stock.adobe.com
Seite 18: muratart/stock.adobe.com