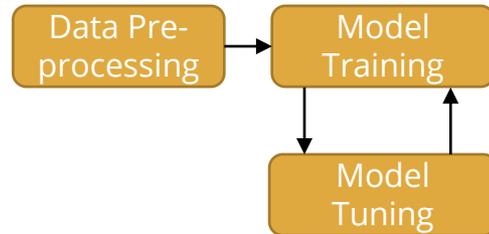


SAS VIYA VS. KUBEFLOW

Ein Vergleich aus der Perspektive eines MLOps Engineers

Nils Quiring & Verena Barth

ML LEBENSZYKLUS: TRADITIONELLES ML



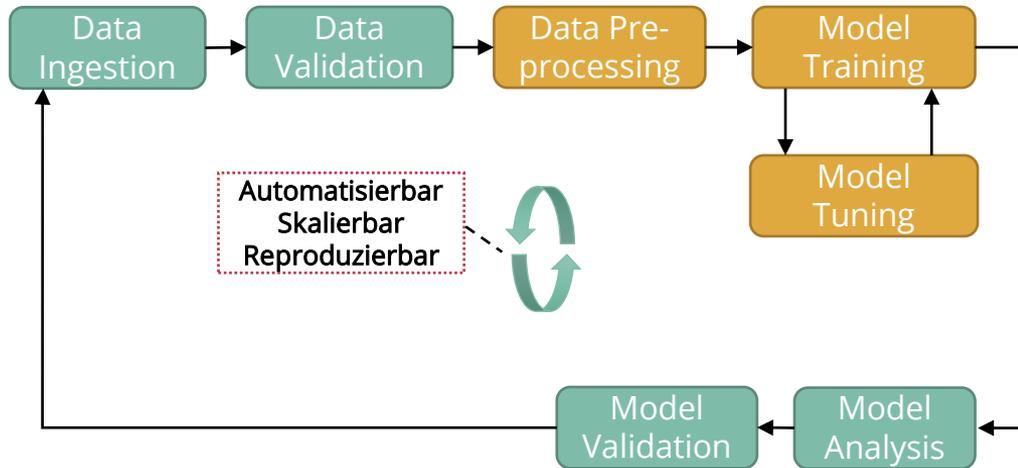
Experimentieren

- Datenvorverarbeitung
- Algorithmen
- Feature Selection
- Modell Tuning

Ziele

- Kurze Entwicklungszyklen
- Verbesserung der Modellmetriken

ML LEBENSZYKLUS: DEV / PIPELINES

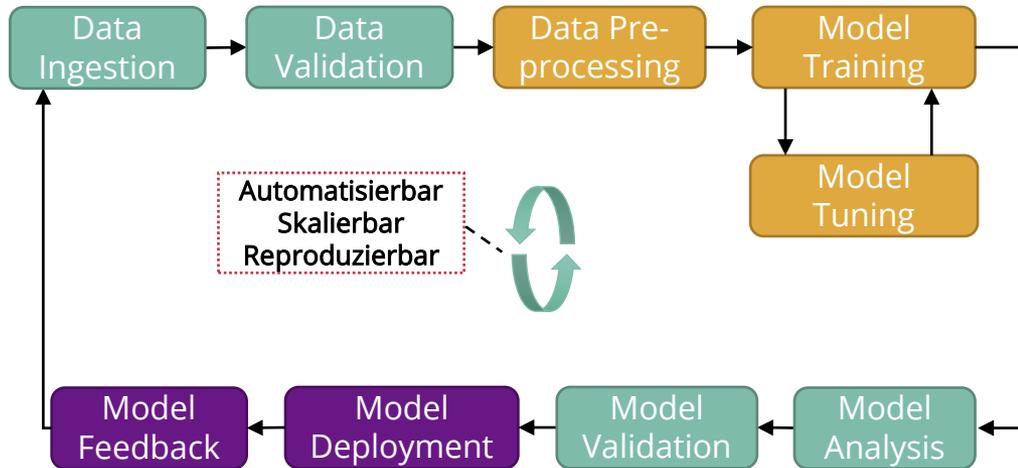


- Professionalisierung des Lebenszyklus und Tests
- Erstellung orchestrierter Pipelines

Ziele

- Continuous integration
- Reduzierung des Wartungsaufwands

ML LEBENSZYKLUS: PRODUKTION

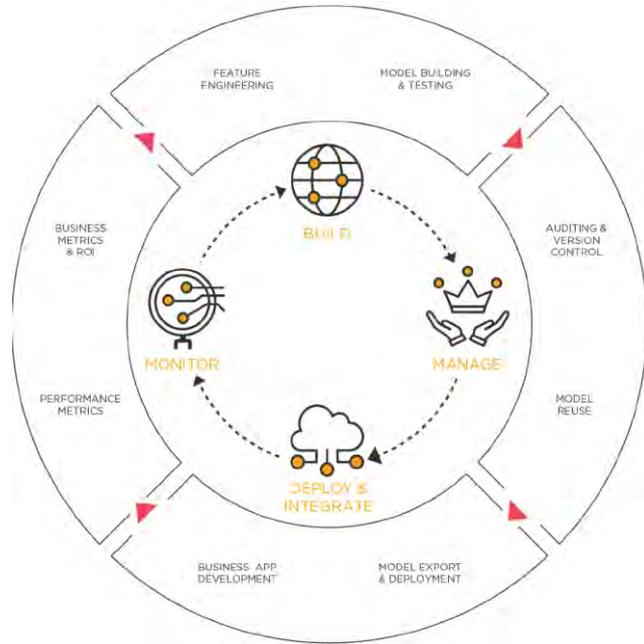


- Model Inferenz
- Monitoring & Management

Ziele:

- Continuous delivery
- End-to-end ML
- Hoher Reifegrad und Automatisierung

MLOPS



Ziele

- Schnellere Erprobung und Modellentwicklung
- Schnellere Einführung aktualisierter Modelle in Produktion
- Qualitätssicherung

Quelle: <https://census.ai/blogs/mlops-guide>

WARUM MLOPS?

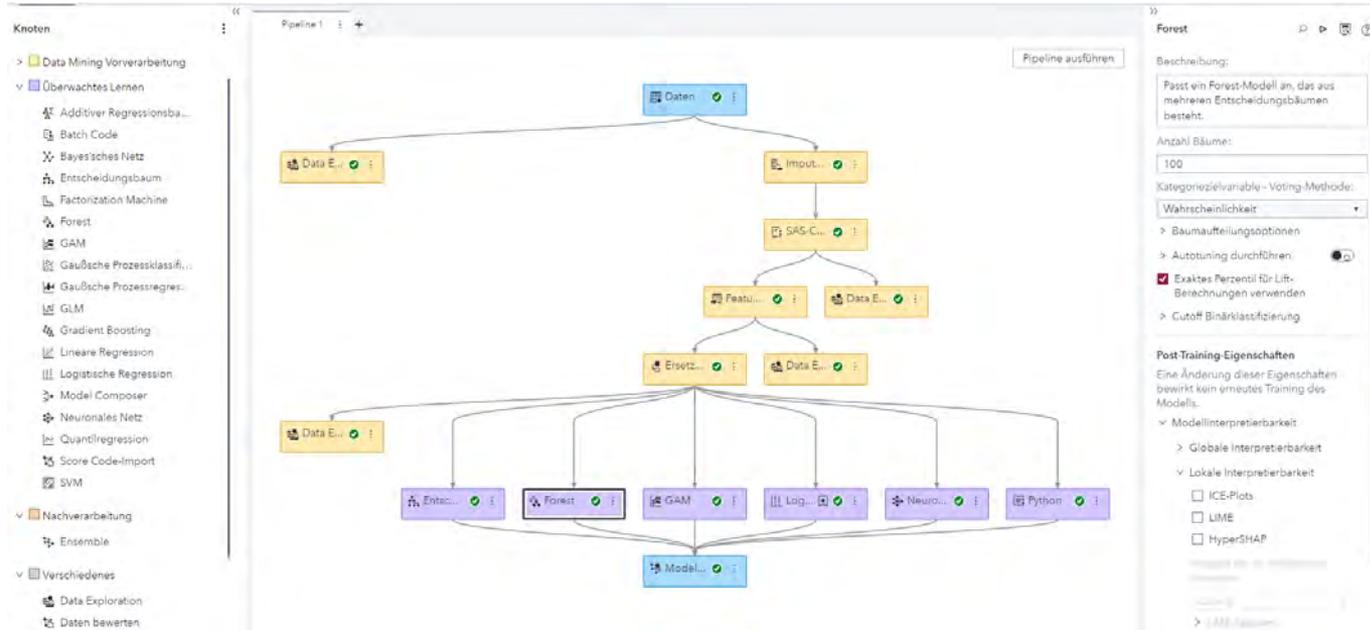
Eine Gartner-Studie zeigt, dass nur 53 % der Projekte den Weg vom Prototyp zur Produktion von künstlicher Intelligenz (KI) schaffen

Gartner

→ Über die Hälfte aller großen Projekte scheitern nach dem PoC

- Mangelnde Skalierbarkeit
- Mangel an Werkzeugen zur Erstellung und Pflege produktionsgerechter KI-Pipelines
- → Governance and Lebenszyklus Management == MLOps

- Kubernetes basierte Plattform für ML, Analytics und Daten Management
 - Deckt den kompletten MLOps Lebenszyklus ab
 - Ganzheitliche Plattform für Data Scientists, Business Analysten, ML Engineers, Führungskräften, ...
- Webbasierte graphische Oberfläche
- Open-Source-Integration mit etwa Python oder R
- Nach Bedarf dynamisch skalierbare Architektur





ÜBERBLICK

- Open Source Projekt mit großer Community
- Multi-Architecture für End-2-End Machine Learning
 - Besteht aus den Best-Of-Breed Open-source ML Systemen
- Ziel: Vereinfachung der Skalierung und Bereitstellung von ML Modellen
 - Deployment auf verschiedenen Infrastrukturen
 - Bereitstellung und Verwaltung lose gekoppelter Microservices
 - Skalierung nach Bedarf
- Unterstützung vieler Programmiersprachen und Frameworks



kubernetes

ML PIPELINES



GEMEINSAMKEITEN

- Ziele
 - Orchestrierung ML Aufgaben in ML Pipelines
 - Schnelle und vereinfachte Erzeugung, Bereitstellung, Versionierung und (Performance) Überwachung von ML Modellen
- Fördern die Zusammenarbeit und gemeinsame Nutzung von Teams
- Nutzung von Kubernetes Ressourcen
 - Skalierung nach Bedarf
 - Bereitstellung in verschiedene Infrastrukturen

UNTERSCHIEDE



Geschlossene Plattform



Modular und erweiterbar

Schneller Entwicklungsstart
(No/Low-Code)



Einarbeitung notwendig

Grundlegendes Vorwissen
vorausgesetzt



Fachkenntnisse notwendig
(ML, Kubernetes)

(Zielgruppe nicht nur Entwickler)

UNTERSCHIEDE



Kompletter Lebenszyklus



Kein Datenmanagement

Vordefinierte, limitierte Auswahl
(Komponenten, Programmiersprachen,
Algorithmen, Prozesse)



Maximale Flexibilität

Archivierung



Volle Nachvollziehbarkeit

FAZIT

Wann ist welche Plattform zu empfehlen?



schnelle, visuelle (datenanalytische) Ergebnisse mit wenig programmatischem Aufwand
→ Pipelines per Click



maximale Flexibilität bei mehr programmatischem Aufwand
→ Everything-as-code

VIELEN DANK FÜR IHRE AUFMERKSAMKEIT

Verena Barth

Verena.Barth@viadee.de



Nils Quiring

Nils.Quiring@viadee.de

