DOZIERENDER: MAX MUSTERMANN

DATA ENGINEERING

THEMENLANDKARTE

Datensystem-Grundlagen	1
Datenverarbeitung „at Scale“	2
Microservices	3
Governance und Sicherheit	4
Verbreitete Cloud-Plattformen und -Dienste	5
Data Ops	6

LEKTION 6

DATAOPS

LERNZIELE

- erklären, was **DevOps, DataOps** und die **Grundsätze von DataOps** sind
- erklären, was **MLOps** ist und welche **Phasen ein Data Science-Projektplan** umfasst.
- die Methode der **Containerisierung** von Anwendungen erklären
- erläutern, was **Docker** und **Kubernetes** sind
- erklären, was eine **Pipeline für maschinelles Lernen** ist und wie eine entsprechende Architektur aufgebaut werden kann
- beschreiben, was das Uber-Produkt, **Michelangelo ML Workflow** ist

EINFACH ERKLÄREN

1. Was meint der Begriff **CI/CD**?
2. Was unterscheidet **Container** von **virtuellen Maschinen**?
3. Ein **Data Science-Projekt** läuft für gewöhnlich **nicht linear**, sondern in mehreren Durchläufen ab, wobei Schritte revidiert und wiederholt werden müssen. Wie passt das mit dem linearen Character von DevOps-bzw. MLOps-Pipelines zusammen?

WIEDERHOLUNG DER KERNPUNKTE DER LEKTION

Grundlegende Prinzipien

- DevOps
- DataOps
- MLOps

Containerisierung

- Einführung und Abgrenzung zu virtuellen Maschinen
- Docker
- Kubernetes

Aufbau von Daten- und ML-Pipelines

- Einführung in ML-Pipelines
- Kubeflow Pipelines
- ML-Pipelines für Echtzeit-Vorhersagen
- Ein Beispiel für eine ML-Pipeline

WIEDERHOLUNG DER KERNPUNKTE DER LEKTION

Grundlegende Prinzipien

- DevOps
- DataOps
- MLOps

Containerisierung

- Einführung und Abgrenzung zu virtuellen Maschinen
- Docker
- Kubernetes

Aufbau von Daten- und ML-Pipelines

- Einführung in ML-Pipelines
- Kubeflow Pipelines
- ML-Pipelines für Echtzeit-Vorhersagen
- Ein Beispiel für eine ML-Pipeline

DevOps

- Verbesserung der **Zusammenarbeit** zwischen Entwicklungs- und Betriebsteams (Allspaw & Hammond, 2009)
- **Nahtlose**, **transparente** und **vollständig integrierte Anwendungsentwicklung** und **Inbetriebnahme** (Allspaw & Hammond, 2009)
- **Continuous Integration und Continuous Delivery (CI/CD)**
- **Verkürzung von Entwicklungszyklen**

Quelle Text: Mezak, 2018.

DataOps

- Übertragung von **DevOps-Prinzipien und Werkzeugen** auf **Datenmanagement** und **Datenanalyse**

- **Verkürzung von Entwicklungszyklen**

Quelle Text: Bergh et al., 2019.

MLOps Übertragung von **DevOps- und DataOps-Prinzipien und Werkzeugen** auf **Data Science** Projekte

— **Reproduierbare Entwicklung, Bereitstellung, Überwachung und Wartung** von **Machine Learning (ML) Modellen** in operativen Systemen

Quelle Text: Sweenor et al., 2020.

WIEDERHOLUNG DER KERNPUNKTE DER LEKTION

Grundlegende Prinzipien

- DevOps
- DataOps
- MLOps

Containerisierung

- Einführung und Abgrenzung zu virtuellen Maschinen
- Docker
- Kubernetes

Aufbau von Daten- und ML-Pipelines

- Einführung in ML-Pipelines
- Kubeflow Pipelines
- ML-Pipelines für Echtzeit-Vorhersagen
- Ein Beispiel für eine ML-Pipeline

- **Standardisierte Einheit**
- Umfasst den **Anwendungscode**…
- …und seine **Abhängigkeiten**
- **Unabhängig von der Umgebung** (bspw. OS)
- **Entkopplung** der Anwendung von der Runtime des Hosts
- **Portable** Anwendungen

Quelle Text: Docker, 2020

Tab. 1: Vergleich zwischen virtuellen Maschinen und Containern

Virtuelle Maschine (VM)	**Container**
Isolation von Anwendungen	
Entkopplung von **Anwendungen** und **Hosts**	
Virtualisierung von **Hardware**	Virtualisierung von **Betriebssystemen (OS)**
pro VM **eigener Kernel**	teilen sich den **Kernel** mit dem **Host-OS**
großer Overhead beim start (müssen „booten“)	starten mit **wenig overhead**
verbrauchen **viel Speicherplatz**	verbrauchen **wenig Speicherplatz**

Quelle Text: Tab. 1: Christian Müller-Kett, 2022.

Docker

- **weit verbreitete Open-Source-Software** zur Containerisierung
- für **alle gängigen Betriebssysteme** verfügbar
- Container werden durch **Dockerfiles** und **Images** definiert, die in einer zentralen **Image Registry** verwaltet und geteilt werden können
- **Automatisierte Bereitstellung** durch Ausführung von **Containern als Instanzen** dieser Images

Docker-Komponenten

- **Dockerfile**
- **Docker Image**
- **Docker run**
- **Docker Engine**

Docker-Workflow

- **Revisionskontrolle**
- **Build**
- **Testen**
- **Bereitstellung**

Quelle Text: Matthias & Kane, 2018.

Kubernetes

- **Container-Orchestrierer**
- **2014** von **Google** als **Open-Source-Projekt** gegründet
- **Cluster** aus mehreren **Nodes** und **Pods**

Kubernetes-Komponenten

- Control Plane
  - kube-apiserver
  - etcd
  - kube-scheduler
  - kube-controller-manager
  - cloud-controller-manager
- Nodes
  - kublet
  - kube-proxy

Quelle Text: Kubernetes, 2020.

WIEDERHOLUNG DER KERNPUNKTE DER LEKTION

Grundlegende Prinzipien

- DevOps
- DataOps
- MLOps

Containerisierung

- Einführung und Abgrenzung zu virtuellen Maschinen
- Docker
- Kubernetes

Aufbau von Daten- und ML-Pipelines

- Einführung in ML-Pipelines
- Kubeflow Pipelines
- ML-Pipelines für Echtzeit-Vorhersagen
- Ein Beispiel für eine ML-Pipeline

ML-Pipelines

1. Problemdefinition
2. Datenerfassung
3. Datenaufbereitung
4. Datenpartitionierung
5. Modell-Training
6. Modell-Bewertung
7. Modell-Bereitstellung
8. Überwachung der Modellgüte

Quelle Text: Koen, 2019, Hapke, 2020.

Kubeflow Pipelines

- Plattform zur standardisierten Erstellung von ML-Pipelines zur Bereitstellung in einem Kubernetes-Cluster
- Benutzeroberfläche (UI)
- Motor
- Software Development Kit (SDK)
- Notebooks
- Nutzt Argo

Quelle Text: Kubeflow, 2020

Beispiel für eine ML-Pipeline

- Uber‘s **Michelangelo**
- Pipeline für **automatisiertes Modell-Training und Bereitstellung**
- **Vorhersage von Lieferzeiten** durch UberEATS (Lieferdienst für Essensbestellungen)
- **Online-** und **Offline-Elemente**

Quelle Text: Hermann & Del Balso, 2017.

REVIEW LERNZIELE

- erklären, was **DevOps, DataOps** und die **Grundsätze von DataOps** sind
- erklären, was **MLOps** ist und welche **Phasen ein Data Science-Projektplan** umfasst.
- die Methode der **Containerisierung** von Anwendungen erklären
- erläutern, was **Docker** und **Kubernetes** sind
- erklären, was eine **Pipeline für maschinelles Lernen** ist und wie eine entsprechende Architektur aufgebaut werden kann
- beschreiben, was das Uber-Produkt, **Michelangelo ML Workflow** ist

EINHEIT 1

TRANSFERAUFGABE

TRANSFERAUFGABE

Ein Start-Up das **nachhaltige Produkte in kleineren Geschäften** vertreibt war in den letzten Jahren sehr erfolgreich. In Folge sollen **weltweit weitere Filialen** eröffnet werden. Als Data Engineer:in sind Sie damit beauftragt, das **Datensystem zu entwerfen**, welches Daten über die **angebotenen Produkte** und **deren Zulieferer** speichert und verarbeitet.

Mithilfe von Methoden des **maschinellen Lernens** werden Modelle erstellt, die Vorhersagen über die zukünftig benötigte Artikelmengen erlauben. Vorab ist uns nicht klar, **welche Algorithmen** und **welche Parameter** am besten geeignet sind, um möglichst performante Modelle zu trainieren. Außerdem wollen wir uns bei der Bereitstellung für die nächsten Jahre **nicht auf eine bestimmte Umgebung festlegen**.

TRANSFERAUFGABE

Die Modelle sollten also problemlos in allen Cloud-Umgebungen und auch unseren eigenen Servern laufen, unabhängig von den dort installierten Betriebssystemen und Bibliotheken. Eine weitere Anforderung sind **schnelle Entwicklungszyklen**, die es uns erlauben, Modelle schnell und einfach neu zu trainieren, wenn bspw. neue Frameworks entwickelt werden, oder wir über andere Daten verfügen.

Erläutern Sie, wie diese Anforderungen mit Hilfe von **DevOps/DataOps/MLOps-Methoden und Tools** erfüllt werden können.

TRANSFERAUFGABE
PRÄSENTATION DER ERGEBNISSE

Bitte stelle deine Ergebnisse vor.
Im Plenum werden die Ergebnisse diskutiert.

1. Welcher Schritt gehört nicht zu einem operationalisierenden Data-Science-Projektplan?.
   a) DSVGO-Überwachung
   b) Model Aufbau
   c) Verwaltung von Lebenszyklus des Models
   d) Überwachung des Models

2. Wie heißt die portable, erweiterbare Open-Source-Plattform für die Verwaltung containerisierter Workloads und Services?
   a) Kubernetes
   b) Docker
   c) Google Functions
   d) Azure Functions

3. In welchem Schritt der Pipeline für maschinelles Lernen werden die eingelesenen Daten auf Formatunterschiede, Ausreißer, Trends, fehlende Werte, Anomalien usw. untersucht?
   a) Daten-Ingestion (Erfassung)
   b) Datenpartitionierung
   c) Model-Training
   d) Datenvorbereitung

Wie hat Ihnen der Kurs gefallen?

QUELLENVERZEICHNIS

Allspaw, J., & Hammond, P. (2009, June 22 - 24). *10+ deploys per day: Dev and ops cooperation at Flickr* [Video]. O'Reilly Velocity: Web Performance and Operations Conference. https://www.oreilly.com/library/view/devops-in-practice/9781491902998/video169253.html

Bergh, C., Benghiat, G., & Strod, E. (2019). *DataOps cookbook*. DataKitchen.io

Docker. (o. D.). *What is a Container?* https://www.docker.com/resources/what-container/

Hapke, H. (2020). *Building machine learning pipelines: Automating model life cycles with TensorFlow*. O'Reilly.

Hermann, J. & del Baso, M. (2017, 5. September). *Meet Michelangelo: Uber's Machine Learning Platform*. Uber Engineering. https://eng.uber.com/michelangelo-machine-learning-platform/

Koen, S. (2019, 5. April). *Architecting a Machine Learning Pipeline*. Towards Data Science. https://towardsdatascience.com/architecting-a-machine-learning-pipeline-a847f094d1c7

Kubernetes. (2022, 20. Februar). *Kubernetes Documentation*. https://kubernetes.io/docs/home/

Kubeflow. (2022, 5. Mai). *Kubeflow Pipelines Introduction*. https://www.kubeflow.org/docs/components/pipelines/introduction/

Matthias, K., & Kane, S. P. (2018). *Docker: Up & running: Shipping reliable containers in production (2nd ed.)*. O'Reilly.

Mezak, S. (2018, 25. Januar). *The Origins of DevOps: What's in a Name?* DevOps.Com. https://devops.com/the-origins-of-devops-whats-in-a-name/

Sweenor, D., Hillion, S., Rope, D., Kannabiran, D., Hill, T., & O'Connell, M. (2020). *ML Ops: Operationalizing Data Science*. O'Reilly.