

# Einführung in MLOps

23 FREIE ÜBUNGEN

Tobias Mérinat

teaching2025@fsck.ch

Lucerne University of  
Applied Sciences and Arts

**HOCHSCHULE  
LUZERN**

DEPARTMENT OF INFORMATION TECHNOLOGY  
Lucerne University of Applied Sciences and Arts  
6343 Rotkreuz, Switzerland

14. und 15. Februar 2025

Nachfolgend findest du eine Liste von möglichen Übungen, welche in Bezug zum Gelernten stehen. Suche dir daraus diejenigen aus, welche dich am meisten interessieren.

Du musst am Schluss nichts abgeben, aber wenn du möchtest, darfst du deine Arbeit gerne kurz präsentieren. Melde dich dazu beim Dozenten.

Willst du das gelernte direkt im Kontext deiner Transferarbeit anwenden? Versuche, dein Modell zu operationalisieren!

# Erweitere das Mushroom Beispiel

Wir haben beim Mushroom Modell einige Interessante Aspekte noch nicht implementiert.

- Baue ein Realtime (RT) Feature ein. Füge dazu im Trainingscode ein einfaches, weiteres Feature hinzu, welches aus den bestehenden Features berechnet wird. Zum Beispiel ein Average. Passe auch den Inference-Code und den Datengenerator entsprechend an.
- Verwende ein weiteres zusätzliches Feature, welches in einem Batch-Prozess zum Voraus berechnet wurde. Beispielsweise könnte eine Pilz-Identifikations-App die aktuellen Koordinaten verwenden, um in einer Tabelle nachzuschlagen, welche Pilze in der Region typisch sind. Dies ist vom Vorgehen her nicht viel anders als der erste Vorschlag.
- Schwieriger wird es, wenn die in der Region typischen Pilze häufig ändern. Vielleicht arbeiten wir in einer nuklear verseuchten Umgebung und katalogisieren zusammen mit vielen anderen Personen Pilze, und benötigen die durchschnittliche Pilz-Grösse der letzten 30 Minuten in der Region. Diese müssten wir als Near Realtime (NRT) Feature in einem Streaming Prozess berechnen.

Wir haben im Kurs nur ein sehr einfaches Modell operationalisiert. Es wäre interessant, auch ein Modell zu operationalisieren, welches Feature-Berechnungen durchführt sowie ein Natural Label hat und dadurch mehr Überwachung und Retraining erlaubt. Dazu drei Ideen mit Datenquellen:

- Baue eine Vorhersage für **Luftqualität**
- Sage den **Bitcoin Preis** vorher
- Mache eine **Wetterprognose**

Möchtest du mehr lernen über Feature Stores und einen Feature Store verwenden? Dann ist Hopsworks eine gute Quelle. Der Feature Store von Hopsworks kann mit einem freien Account gut für Versuche verwendet werden. Hopsworks bietet eine grosse Anzahl an **Tutorials**. Einige Vorschläge

- Die wichtigsten Konzepte (**Feature Store**, **Feature Groups**, **Feature Views**)
- **Kurze Tutorials zum starten** (filtere nach *Basic Tutorial*)
  - Variante des Fraud Batch Tutorials mit Data Validation mit Great Expectations
- **Fortgeschrittene Tutorials** (filtere nach *Advanced Tutorial*)

Interessierst du dich für Data Drift, findest du auf der Webseite von Evidently viel gutes Material.

- Tutorials mit Notebooks
- Blog Posts, zum Beispiel
  - How to handle ML model drift in production
  - How to interpret data and prediction drift together?
  - 5 methods to detect drift in ML embeddings

Weights and Biases (kurz wandb) ist ein Cloud Dienst, der unter anderem **Experiment Tracking** und eine **Model Registry** anbietet.

- Absolviere ein paar **Tutorials** und lies die **Dokumentation**, um das Produkt kennenzulernen
- Ersetze MLFlow in unserer Infrastruktur durch wandb
- Schreibe den bestehenden Code um, damit dieser wandb verwendet anstelle von MLFlow



Wenn du dich mit Orchestrierung befassen möchtest, ist **Mage** ein Produkt, welches sich gut für den Einstieg eignet.

- Setze analog der anderen Services einen Orchestrierungs-Service auf. **Docker Files** stehen zur Verfügung.
- Schau dir die **Quickstarts** an
- Bringe unsere Batch-Pipeline unter Mage zum laufen

Möchtest du lieber Airflow verwenden, findest du hier ein Tutorial mit Code:

- **Why use Apache Airflow (or any orchestrator)**

Möchtest du Great mehr über Great Expectations wissen, kannst du die folgenden Vorschläge anschauen.

- Ein **kurzes Tutorial** mit Code zur Verwendung von Great Expectations in Datenpipelines
- Ein **kurzer Artikel** zu Data Quality Checks

Möchtest du Grafana besser kennenlernen und das Monitoring ausbauen? Hier einige Ideen dazu.

- Schau dir den **Grafana Quickstart** an
- Erweitere das Drift Detection Dashboard um Visualisierungen, welche
  - den Verlauf der berechneten Distanzen anzeigen
  - Den Zustand des Datensatzes in einer einzigen Visualisierung gut darstellen
  - Setze einen Threshold und lasse Grafana einen Alarm anzeigen, sobald dieser überschritten ist
- Lasse dich von den **Visualisierungen von Evidently** inspirieren
- Baue ein Dashboard, um den Betrieb von Statsd, Prometheus und Grafana zu überwachen

Möchtest du mehr über Stream Processing lernen, schau dir das untenstehende, kurze Tutorial an.

- Data Engineering Project: Stream Edition

Möchtest du mehr über Docker erfahren, eignet sich das folgende längere Tutorial gut, um sich alle notwendigen Grundlagen anzueignen.

- Docker Curriculum
- Danach kannst du versuchen, den Data Generator zu containerisieren

Möchtest du wissen, wie grössere Firmen MLOps implementieren?

- Lies ein paar **Case Studies**

Wenn du in einfach verschiedenen Artikeln stöbern möchtest, beginne am besten mit der untenstehenden Liste.

- **MLOps Guide von Chip Huyen**