

MLOps-Programmi erkurs

MLflow Ambassador - 2024-06-05

Médéric HURIER und Matthieu JIMENEZ

Über uns

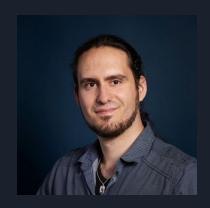












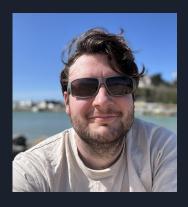
Dr. Médéric HURIER

Freiberuflicher MLOps-Ingenieur

Arbeiten für Decathlon Digital

Doktor in Computersicherheit und KI

https://www.fmind.dev/



Dr. Matthieu JIMENEZ

Wissenschaftlicher Mitarbeiter

Arbeiten für Universität Luxemburg

Doktor in Computersicherheit und KI

Stärken Sie Ihre
KI/ML-Programmierkenntnisse mit dem

Wo anfangen?

Vieles zu wissen im Maschinellen Lernen (Frameworks, Lösungen, Theorien, Daten



Aber wie sieht es mit den codebezogenen aus?

Viele Software-Engineering-Konzepte und -Technologien finden Anwendung im ML, insbesondere für die Produktion.

Eine Lösung besteht darin, einen ML-Ingenieur zu fragen, doch dies führt zu einem Engpass und wird durch Kommunikationsprobleme behindert.



© adobe

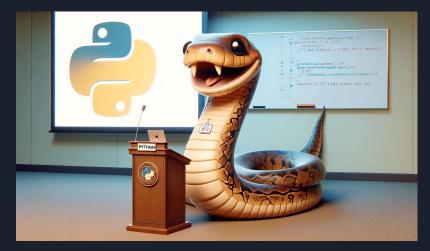
MLOps-Python-Paket

Wo alles wirklich begann

Basierend auf einer echten MLOps-Vorlage erstellt.

Beinhaltet viele nützliche Open-Source-Bibliotheken für MLOps.

Design Patterns von Haus aus integriert.



© datascientest

Problem, wie man alle Konzepte dahinter erklärt?

https://github.com/fmind/mlops-python-package

Die Idee

Einen Kurs entwickeln, um die Wissenslücke zu schließen.

Sollte sein **plattformunabhängig** (bereits viele Anbieterlösungskurse existieren)

Sofort anwendbar konzipiert (z.B. für Mederics Kollegen bei Decathlon)



© Wikimedia Commons ₆

Einführung in den MLOps Coding-Kurs

Veröffentlichung am 22. Mai 2024

7 Kapitel, die verschiedene MLOps-Themen aus der Coding-Perspektive abdecken

Kapitel folgen dem natürlichen Verlauf eines Projekts.

Aufgebaut um ein Q&A-System, mit zusätzlichen Ressourcen und Beispielen

Gekoppelt an das MLOps Python Package

MLOps Coding Course

```
KIND: T.Literal["TrainingJob"]
                                                                        PunConfig(name="Training")
run_config: services.Mlflow
inputs: datasets.ReaderK
targets: datasets.Reade
model: models.ModelKin
                                                                             nator="KIND")
metrics: list[metrics]
                                                                             iscriminator="KIND")
splitter: splitters.Sp
                                                                              discriminator="KIND")
saver: registries.Saver.
                                                                            nator="KIND")
signer: signers.SignerKind
                                                                          cor="KIND")
# - avoid shadowing pydantic
registry: registries.RegisterKi
                                                                   __ister(), discriminator="KIND")
```

https://mlops-coding-course.fmind.dev/

Nun zum Kurs!



Overview

Course, Project, Plateforme





Initializing

Python, pip, pyenv, Poetry, git, Vscode

Prototypin

Notebook, Import, Config, Dataset, Analysis Modeling

Productionizing

Package, Module,
Paradigm, Entrypoint,
Configuration,
Documentation

Validating

Typing, Linting, Testing, Logging, Security, Debugging

Refinina

Design Pattern, Task Automation, CI/CD, Container, Model Registry

Sharing

Repository, License, Readme, Release, Template

<u>O. Übersicht</u>: Eine Einführung in den Kurs

Nullbasierte Indizierung 🕖

- → <u>0.0. Kurs</u>Worum geht es bei diesem Kurs?
- → <u>0.1. Projekte</u>Standard oder eigenes mitbringen!
- → <u>0.2. Datensätze</u>Zur (Un-)Struktur der Daten
- → <u>0.3. Plattformen</u>Wir sind hier plattformunabhäng
- → <u>0.4. Mentoring</u>Einzel-Lernsitzungen
- → 0.5. Assistenten Ein Premium RAG für Sie
- → <u>0.6. Ressourcen</u>Über den Kurs hinaus



Things You Learn https://xkcd.com/1775/

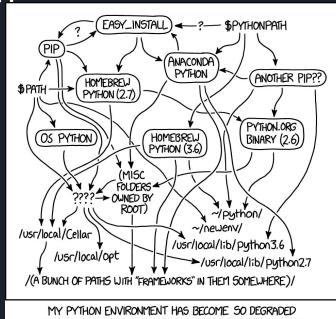
<u>Initialisierung</u>Bereite deine

Entwicklungsumgebung vor.

Lasst uns unsere Pythons mitbringen 🐍



- 1.0. System Sie haben mein Betriebssystem
- 1.1. Python Und mein Python-Interpreter
- 1.2. Pyeny Und meine Python-Umgebung
- 1.3. PoesieUnd mein Python-Paketmanager
- 1.4. GitUnd mein Versionskontrollsystem
- 1.5. GitHubUnd mein Code-Repository
- 1.6. VS Code: Und mein Code-Editor



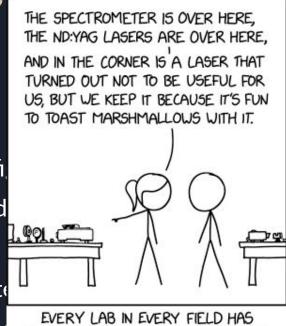
THAT MY LAPTOP HAS BEEN DECLARED A SUPERFUND SITE.

Python Environment https://xkcd.com/1987/

2. Prototyping Erkunden Sie Ihren Suchraum

Zu Jupyter Notebooks und darüber hinaus 🔊

- → 2.0. Notebooks Buchen oder nicht buchen?
- → <u>2.1. Importe Mit der Kraft des Python-Ökosystems</u>
- → <u>2.2. Konfigurationen</u>Trenne den Code von der Konfi
- → <u>2.3. Datensätze:</u> Wie man Datensätze lädt und hand
- → <u>2.4. Analyse</u>Mit Daten sprechen lernen
- → 2.5. Modellierung KI/ML-Modelle und Pipelines erste
- → <u>2.6. Evaluierungen</u>Die Wissenschaft in die Modellie einbringen.



EVERY LAB IN EVERY FIELD HAS SOME PIECE OF EQUIPMENT LIKE THIS.

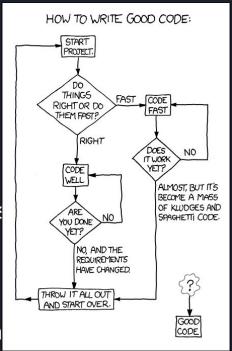
> Laborausrüstung Stärken Sie Ihre

3. Produktivsetzung Erreichen Sie Ihr

Publikum!

Eintritt in das Industriezeitalter 🔅

- → 3.0. Paket: Package = Metadaten + Code
- → 3.1. ModuleIn Module aufteilen und meistern
- → <u>3.2. Paradigmen Die beste Programmierphilosophie</u>
- → 3.3. Einstiegspunkte Pakete für andere Systeme bereits
- → <u>3.4. Konfigurationen</u>Bearbeite deine Konfigurationen, deinen Code
- → 3.5. <u>Dokumentationen</u>Gute Dokus machen gute Freun
- → <u>3.6. VS Code Arbeitsbereich</u>Organisieren Sie Ihren Arbeitsbereich



Guter Code
Stärken Sie Ihre

MI Onc-Programmiarkurs

3. Produktivsetzung – Wichtige Erkenntnisse

- Das Erstellen eines Python-Pakets ist entscheidend.
- Modularität hilft, Belange zu trennen.
- Konfigurationen vom Code trennen
- Ihre Lösung anderen Systemen bereitstellen

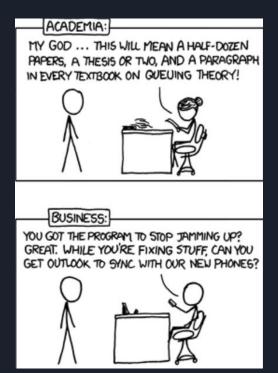
4.0 Validierung Versicherung und

Rückversicherung

Nichts zu sehen von formalen Methoden 🌝



- 4.0. Codieren Typen sind Schemata
- 4.1. LintingLinters sind Schutzmechanismen
- 4.2. Testen Tests sind ausführbare Spezifikationen
- 4.3. Logging Logging ist Ihr Nachrichtendienst
- 4.4. Sicherheit Sicherheit ist Lösung, Bodyguard
- 4.5. FormatierungCode stilgerecht formatieren mit Formatierern
- 4.6. Debugging Debugging ist der beste Bug-Killer



Wissenschaft vs. Wirtschaft Stärken Sie Ihre

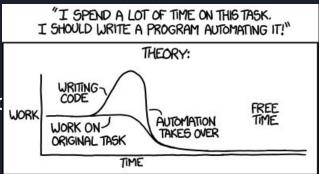
4.0 Validierung - Schlüsselerkenntnisse

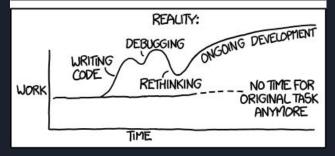
- Softwarevalidierung ist eine Investition ...
- Aber es hat langfristig positive Erträge.
- Man will nicht von einem dummen Bug aus dem
 - Schlaf gerissen werden.
- Die Validierungen sind die Regeln der

<u>5. Verfeinerung</u>fortgeschrittene Konzepte für Ingenieure

Softwareentwickler 2.0 &

- → <u>5.0. Entwurfsmuster</u>SOLID Codebasen aufbauen
- → <u>5.1. Aufgabenautomatisierung</u>Wiederhole dich nic
- → <u>5.2. Pre-Commit-Hooks</u>Ihr lokales CI/CD
- → <u>5.3. CI/CD Workflows</u> Ihr gemeinsames CI/CD
- → <u>5.4. Software-Container</u>Halte es reproduzierbar
- → <u>5.5. AI/ML Experimente MLflow-Tracking</u>
- → <u>5.6. Modellregister</u>MLflow Model Registry





Automatisierung <u>Stärken Sie Ihre</u>

5. Verfeinerung – Erkenntnisse

Automatisierung, Automatisierung,

Automatisierung

- Entwurfsmuster sind Lösungen von der Stange.
- Container sind Ihr Mittel der Wahl für

Reproduzierbarkeit.

6. Teilen Bauen mit anderen für andere

Mach jetzt mit und teile die Software 🎜

- → <u>6.0. Repository</u>Referenzieren Sie Ihr Projekt
- → <u>6.1. Lizenz</u>Erzählen Sie, wie man es nutzen kann.
- → <u>6.2. Readme</u>Sagen Sie, was man damit machen kan
- → <u>6.3. Veröffentlichungen</u>Sagen Sie den Leuten, was : hat.
- → <u>6.4. Vorlagen</u>: Erklären, wie man sein eigenes erste
- → 6.5. Arbeitsstationen Hilf Menschen, deinen Code a
- → <u>6.6. Beiträge</u>Erzählen Sie den Leuten, wie sie beitragen köhntertützte Funktionen Stärken Sie Ihre

IT TOOK A LOT OF WORK, BUT THIS LATEST LINUX PATCH ENABLES SUPPORT FOR MACHINES WITH 4,096 CPUs. UP FROM THE OLD LIMIT OF 1,024. DO YOU HAVE SUPPORT FOR SMOOTH FULL-SOREEN FLASH VIDEO YET? NO, BUT WHO USES THAT?

Brauchen Sie Hilfe?

Ob virtuell oder real, wir haben an alles gedacht!

• MLOPS-Coding-Chatbot-Assistent

Gemini mit dem gesamten Kurs als Kontext, in der Lage, zu jedem beliebigen Teil des Kurses Fragen zu beantworten.

Begründen Sie die Wahl der Fragerunde.



Mentoring

Demozeit

Beiträge und Vorschläge

Der Kurs ist Open Source auf GitHub

Zögert nicht, darauf aufzubauen (erwähnt uns einfach;))

Falls Sie Anmerkungen, Korrekturen oder Ideen für neue Inhalte haben, melden Sie sich gerne!

Ich hoffe, dieser Kurs wird Lernenden helfen, Experiment-/Modellverwaltung durch **MLflow**



© DALL.E

Vielen Dank für Ihre

https://mlops-coding-course.fmind.dev

Begleiten Sie uns am MLOps Community