



Stärkt Är AI/ML
Kodéierfäegkeeten
mat dem

MLOps Coding Cours

MLflow Ambassador - 2024-06-05

Médéric HURIER & Matthieu JIMENEZ

Iwwert eis



Dr. Médéric HURIER

Freelance MLOps Ingenieur

Schaffen fir [Decathlon Digital](#)

Doktorat an der Computersécherheet an der KI

<https://www.fmind.dev/>



Dr. Matthieu JIMENEZ

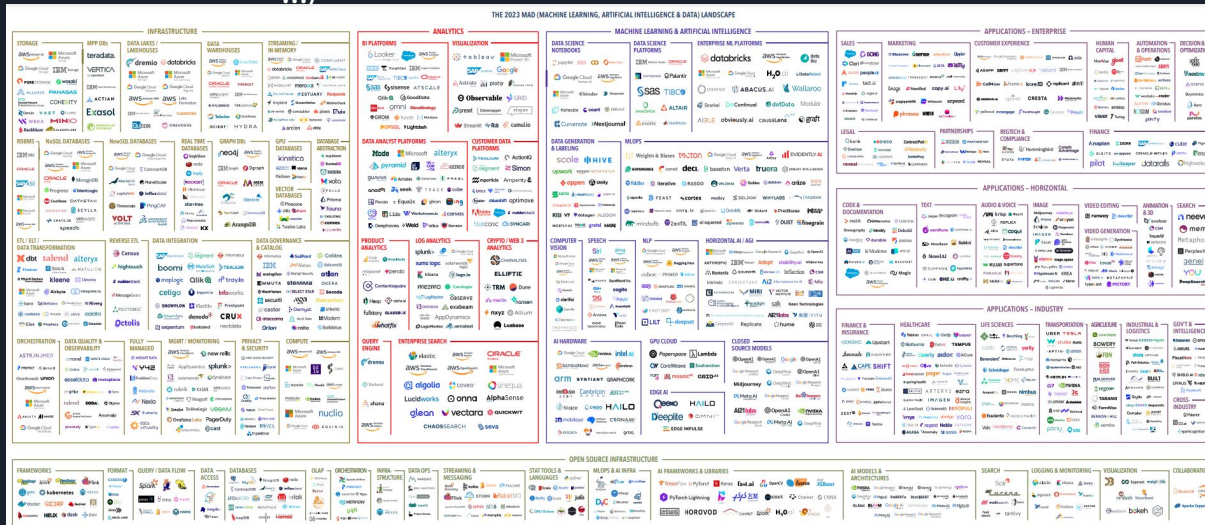
Fuerschungsmataarbechter

Schaffen fir [Universitéit Lëtzebuerg](#)

<https://website.jimenez.lu/>

Wou ufänken?

Et gött vill ze wëssen am Maschinneléieren
(Frameworks, Léisungen, Theorien, Daten
...)



Verréckt Landschaft 2023

Mee wat ass mat deenen, déi codebezunn sinn?

Vill Software Engineering Konzepter an Technologien gëllen fir ML, besonnesch fir d'Produktioun.

Eng Léisung ass en ML Ingenieur ze froen, awer dëst féiert zu engem Flaschenhals an gëtt duerch Kommunikatiounsproblemer behënnert.



MLOps Python Paket

Wou alles wierklech ugefaangen huet

Erstellt baséiert op enger aktueller MLOps
Schabloun

Vill nützlich OSS Bibliothéike fir MLOps abegraff.

Designmuster out-of-the-box integréiert.

Problem wéi een all d'Begrëffer dohannert
erkläert?



© datascientest

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

D'Iddi

E Cours entwéckelen, fir d'Wëssenslück ze iwwerbrécken

Soll sinn **Plattformonofhängeg** (Et
gi schonn vill Coursen iwwer
Ubiddlerléisungen.)

Direkt asetzbar entworfen (z.B. fir dem Mederic
seng Kollege bei Decathlon)



Afëierung vum MLOps Coding Course

Verëffentlechung den 22. Mee 2024

7 Kapitelen, déi verschidden MLOps Themen aus der Kodéierungs-Perspektiv ofdecken.

Kapitele follegen den natierleche Wee vun engem Projet.

Opgebaut ëm e Froen an Äntwerten System, mat zousätzleche Ressourcen a Beispiller

Gekoppelt un den MLOps Python Package

MLOps Coding Course



```
33 KIND: T.Literal["TrainingJob"] = "T"
34
35 # Run
36 run_config: services.Mlflow
37 # Data
38 inputs: datasets.ReaderK
39 targets: datasets.Reade
40 # Model
41 model: models.ModelKin
42 # Metrics
43 metrics: list[metrics.
44 # Splitter
45 splitter: splitters.Sp
46 # Saver
47 saver: registries.Saver.
48 # Signer
49 signer: signers.SignerKin
50 # Registry
51 # - avoid shadowing pydantic
52 registry: registries.RegisterKin
53
```

```
RunConfig(name="Training")
ator="KIND")
iscriminator="KIND")
, discriminator="KIND")
ator="KIND")
cor="KIND")
ister(), discriminator="KIND")
```

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

Elo de Cours!



Overview

Course, Project,
Plateforme



Initializing

Python, pip, pyenv,
Poetry, git, Vscode



Prototyping

Notebook, Import,
Config, Dataset, Analysis,
Modeling



Productionizing

Package, Module,
Paradigm, Entrypoint,
Configuration,
Documentation



Validating

Typing, Linting, Testing,
Logging, Security,
Debugging



Refining

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



Sharing

Repository, License,
Readme, Release,
Template

0. Iwwerbléck eng Aféierung zum Cours

Null-baséiert Indexéierung 

- 0.0. Cours Wat geet et an dësem Cours?
- 0.1. Projeten Standard oder selwer matbréngen!
- 0.2. Donnéeësets Iwwert d'(On)Struktur vu Daten
- 0.3. Plattformen Mir sinn plattformonofhängeg he
- 0.4. Mentoring Eenzel Léiersessiounen
- 0.5. Assistenten Eng Premium RAG fir Iech
- 0.6. Ressourcen Iwwer de Cours eraus



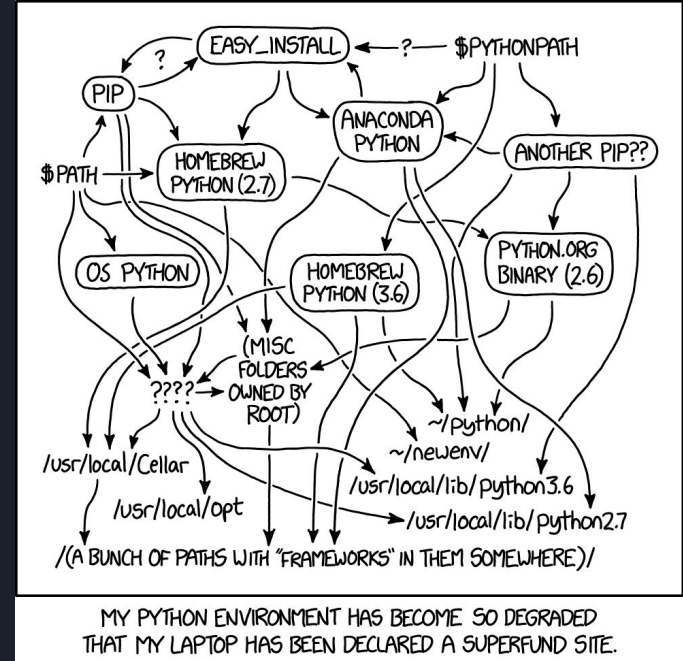
Things You Learn

<https://xkcd.com/1775/>

Initialisieren Bereit dein Entwicklungsumfeld für.

Loosst eis eis Schlaangen matbréngen 🐍

- 1.0. System Dein heutiges Betriebssystem.
- 1.1. Python Dein Python Interpreter
- 1.2. Pyenv Dein Python Umfeld
- 1.3. Poesie Dein Python Package Manager
- 1.4. Git Dein Versionskontrollsystem
- 1.5. GitHub Dein Code Repository
- 1.6. VS Code Dein Code Editor

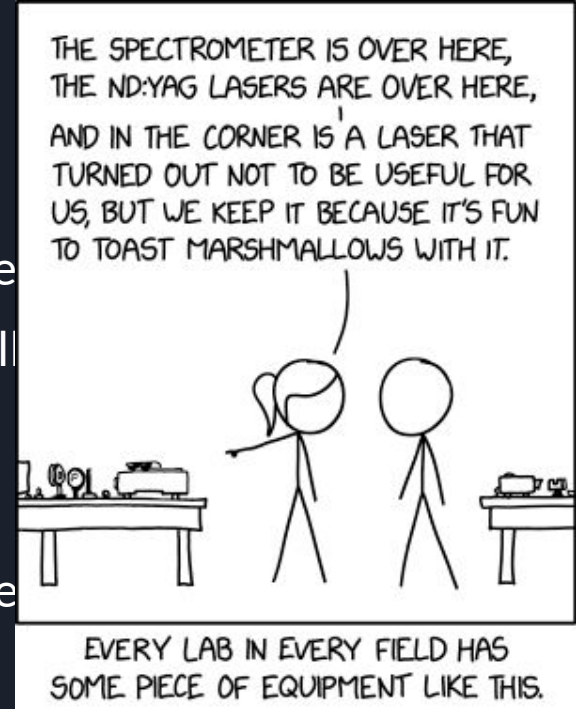


Python Environment
<https://xkcd.com/1987/>

2. Prototyping Erfuersch Äre Sichraum

Zu Jupyter Notebooks an doriwwer eraus 🪐

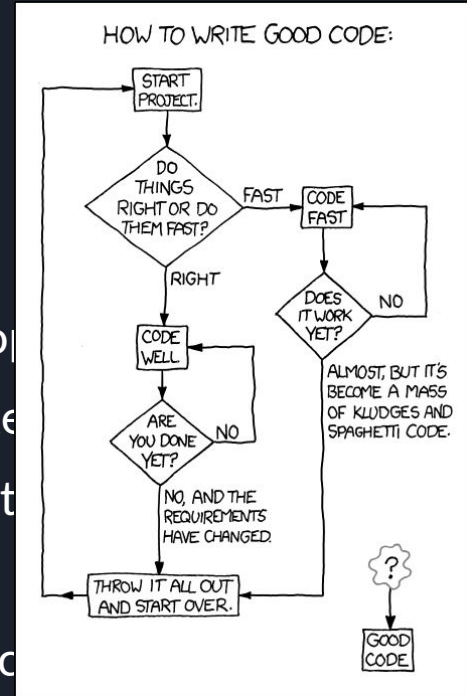
- 2.0. Notebooks Buchéieren oder net buchéieren?
- 2.1. Importen Duerch d'Kraaft vum Python Ekosyste
- 2.2. Konfiguriounen Trennt de Code vun den Astell
- 2.3. Datesätz: Wéi Datesätz lueden an ëmgoen
- 2.4. Analyse Léieren, mat Daten ze schwätzen
- 2.5. Modelléierung KI/ML Modeller a Pipelines erste
- 2.6. Evaluatiounen Intégréier d'Wëssenschaft an d'Modelléierung.



Betriebshilfen Gêi dâi Publikum erreechen

An d'Industriezäitalter 

- 3.0. Package Package = Metadaten + Code
- 3.1. Moduler An Moduler ënnerdeelen an erueweren.
- 3.2. Paradigmen Déi bescht Programméierungsphilosophie
- 3.3. Agankspunkten Packagen aner Systemer bereetst
- 3.4. Konfigurationen Ännert Är Konfigurationen, net
- 3.5. Dokumentationen Gutt Doku mécht gutt Frënn
- 3.6. VS Code Aarbechtsberäich Organiséiert Är Aarbecht



Gutt Code

Stärkt Är AI/ML

Programméierfäegkeete mam MLOps

Programmécours



3. Produktioun - Erkenntnisser

- E Python Package bauen ass de Schlëssel.
- Modularitéit hëlleft, Belangen ze trennen.
- Konfiguratiounen vum Code trennen
- Är Léisung aner Systemer zur Verfügung stellen

4.0 Validéieren

Versécherung an Réckversécherung

Keng Spuer vu formelle Methoden 😊

- 4.0. Typiséierung Typen sinn Schemaen
- 4.1. Linting Linters si Ofsécherungen.
- 4.2. Testen Tester sinn ausféierbar Spezifikatiounen.
- 4.3. Protokolléierung Logging ass Ären Intelligenzdéng
- 4.4. Sécherheet Sécherheet ass Léisungs-Bodyguard
- 4.5. Formatéierung Codéiert stilvoll mat Formatéierer
- 4.6. Feelerbehebung Debuggen ass de beschte Bug-Ki



Akademie géint Entreprise
Stärkt Är AI/ML Kodéierungsfäegkeete
am MLOps Kodéierungscours



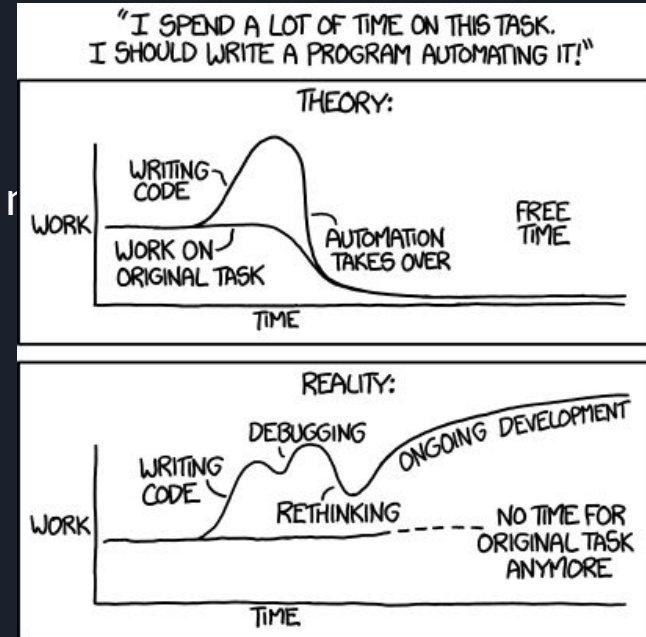
4.0 Validéierung - Haauptpunkte

- Softwarevalidéierung ass eng Investitioun ...
- Awer et huet positiv Rendementen op laang Siicht.
- Dir wëllt net duerch en domme Bug waakreg gemaach ginn.

5. Verfeinerung Fortgeschrittene Konzepte für Ingenieure

Software Engineer 2.0 🙌

- [5.0. Entwurfsmuster](#) SOLID Codebasen opbauen
- [5.1. Aufgabenautomatisierung](#) Wiederholung der
- [5.2. Vir-Commit Hooks](#) Är lokal CI/CD
- [5.3. CI/CD Workflows](#) Är gedeelt CI/CD
- [5.4. Software Container](#) Halt et reproduzierbar
- [5.5. AI/ML Experimentier](#) MLflow Tracking
- [5.6. Modell Register](#) MLflow Modellregister



Automatisierung
Stärkt Är AI/ML Kodierfähigkeiten
mit MLOps Kodierkurs



5. Verfeinerung - Hauptpunkten

- Automatisierung, Automatisierung, Automatisierung
- Designmuster sind für Lösungen.
- Containern sind für Adress für Reproduzierbarkeit.

6. Deelen Bauen mat aneren fir anerer

Maacht elo mat an deelt de Software 🎵

- 6.0. Repository Referéier op däi Projet
- 6.1. Lizenz Sot, wéi d'Leit et kënne benotzen
- 6.2. Readme Sot, wat d'Leit domat kënne maachen.
- 6.3. Verëffentlechungen Sot de Leit, wat geännert h
- 6.4. Schablounen Sot de Leit, wéi si eppes Eegent op
- 6.5. Aarbechtsstatiounen Hëlleft Leit äre Code aust
- 6.6. Bäitrag Sot de Leit, wéi ee bäidroen kann.



Ënnerstëtzt Fonctiounen
Stärkt Är AI/ML Kodéierungsfäegkeete
mam MLOps Kodéierungscours

Braucht Dir eng Hand?

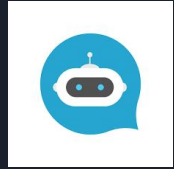
Virtuell oder real, mir këmmen eis ëm iech!

- MLOPS Codéierungs-Chatbot-Assistent

Gemini mat dem ganze Cours als Kontext, kann iwuer all spezifeschen Deel vum Cours äntweren.

D'Wiel vu Q&A erklären

- Mentoring



Gemini



Demo Zäit

Bäidroen a Virschléi

De Cours ass Open Source op Github.

Zéckt net, dorop opzebauen (ernimmt eis einfach ;))

Wann Dir Kommentaren, Korrekturen oder eng nei
Inhaltsiddi hutt, méllt lech!

Ech hoffen, dëse Cours wäert de Coursisten hëllefen,
d'Gestioun vun Experimenter a Modeller ze verstoen duerch
MLflow



© DALL.E

Merci fir Är
Opmierksamk
eet!

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

Maacht mat um [MLOps Communautéit](#)
[Slack](#)