Machine-Learning-Pipelines mit scikit-learn

Datenvorbereitung und Modellierung reproduzierbar zu einem Workflow verbinden

01 INTRO

Ein Machine-Learning-Modell besteht in der Praxis selten nur aus einem Algorithmus. Imputation, Skalierung, Encoding, Feature Engineering und Modellierung müssen in derselben Reihenfolge und mit denselben Parametern auf Trainings- und Produktionsdaten angewendet werden.

Der Kurs behandelt die Entwicklung reproduzierbarer Machine-Learning-Pipelines mit scikit-learn. Im Mittelpunkt stehen Pipeline, ColumnTransformer, eigene Transformer, Cross Validation und Hyperparameteroptimierung sowie die Vermeidung von Datenleckage.

DAUER

2 Tage

FORMAT

Präsenz · Live Online · Inhouse

VORKENNTNISSE

Praktische Kenntnisse in Python, Pandas, scikit-learn und Machine Learning

02 TRAININGSPROGRAMM

Inhalte

Warum ML-Pipelines?

- Preprocessing und Modell als Einheit
- Reproduzierbarkeit
- Training und Prediction
- manuelle Workflows und ihre Fehler
- Leakage
- Wartbarkeit

scikit-learn Estimator API

- Estimator
- Transformer
- fit
- transform
- predict
- Parameter
- einheitliche Schnittstellen

Pipeline

- Verarbeitungsschritte verbinden
- Preprocessing
- Modell
- Pipeline erstellen
- Parameter adressieren
- Pipeline ausführen

Numerische Features vorbereiten

- Imputation
- Skalierung
- Transformation
- mehrere Schritte
- Feature-Namen
- reproduzierbare Verarbeitung

Kategoriale Features vorbereiten

- Imputation
- One-Hot Encoding
- unbekannte Kategorien
- seltene Kategorien
- robuste Verarbeitung neuer Daten

ColumnTransformer

- unterschiedliche Feature-Gruppen
- paralleles Preprocessing
- numerische und kategoriale Pipelines
- Restspalten
- komplexere Datenstrukturen

Eigene Transformer

- eigene Transformationen
- FunctionTransformer
- scikit-learn-kompatible Komponenten
- Fit-Zustand
- Wiederverwendbarkeit
- Testbarkeit

Pipeline und Cross Validation

- vollständige Pipeline validieren
- Preprocessing innerhalb der Folds
- Leakage vermeiden
- Cross Validation
- mehrere Metriken

Hyperparameter in Pipelines

- Parameterpfade
- Grid Search
- Randomized Search
- Preprocessing und Modell gemeinsam optimieren
- Suchräume
- Rechenaufwand

Pipelines vergleichen und testen

- Baselines
- alternative Preprocessing-Schritte
- unterschiedliche Modelle
- reproduzierbare Experimente
- automatisierte Tests
- Fehleranalyse

Pipeline für den Betrieb vorbereiten

- trainiertes Gesamtmodell
- Persistenz
- Versionsinformationen
- Eingabeschema
- Prediction
- unbekannte Daten
- Grenzen zwischen Training und Serving

Praktisches Pipeline-Projekt

- heterogenen Datensatz vorbereiten
- ColumnTransformer entwickeln
- Modell integrieren
- Cross Validation durchführen
- Parameter optimieren
- vollständigen Workflow reproduzieren

03 DURCHFÜHRUNG

Rahmen und Organisation

Dauer	2 Tage
Durchführung	Präsenz oder Live Online, vorzugsweise als Inhouse-Schulung
Schwerpunkt	Inhouse-Schulungen für Unternehmen und Teams
Sprache	Deutsch, englische Fachbegriffe und Dokumentation
Unterlagen	Kursunterlagen, Beispiele und Übungsaufgaben
Technik	Eigener Rechner mit Python, Jupyter und scikit-learn

Inhouse-Schulungen
Vorhandene Datensätze und Modellierungsworkflows können als Grundlage für die Übungen dienen.

04 ZIELGRUPPE

Für wen ist der Kurs gedacht?

Data Scientists, Entwickler und fortgeschrittene Data Analysts, die Machine-Learning-Workflows mit scikit-learn reproduzierbar und wartbar entwickeln möchten.

Voraussetzungen
Praktische Python- und scikit-learn-Kenntnisse sowie Grundlagen in Machine Learning.

05 LERNZIELE

Was Sie nach dem Kurs können

Die Teilnehmer können vollständige ML-Pipelines entwickeln, unterschiedliche Feature-Typen korrekt verarbeiten, Leakage vermeiden und Preprocessing und Modelle gemeinsam validieren und optimieren.

06 FAQ

Häufige Fragen

Ist das ein MLOps-Kurs?

Nein. Der Schwerpunkt liegt auf reproduzierbaren Modellierungs-Pipelines innerhalb von scikit-learn. Deployment-Infrastruktur und umfassendes MLOps gehen darüber hinaus.

Werden eigene Transformer entwickelt?

Ja. Dadurch lassen sich fachliche Transformationen sauber in den Workflow integrieren.

Was unterscheidet den Kurs vom Feature Engineering?

Der Feature-Engineering-Kurs konzentriert sich auf Merkmalsentwicklung und Dimensionsreduktion als methodisches Thema. Hier geht es um die technische Orchestrierung der gesamten Verarbeitungskette.

UC IT Service · Ulrich Cuber · kontakt@uc-it.de
https://www.uc-it.de/coaching/ml-pipelines/