



Machine Learning und Künstliche Intelligenz

Grundlagen verstehen und verantwortet anwenden

Daten untersuchen · Modelle entwickeln · sicher prüfen

BKWI1 und BKWI2 · BPE 6 und BPE 7

Didaktisch erweiterte Kursfassung · A5 · Schuljahr 2026/2027 · v1.0 · Serienmaster R7

**Christine Janischek · Lizenz Custom License NC School-Only 1.0 ·
emotionalspirit.de**

*Dieses Lernskript wurde mit Unterstützung einer KI erstellt und anschließend redaktionell geprüft.
Trotz sorgfältiger Bearbeitung können Fehler enthalten sein. Bitte melde Fehler und
Verbesserungsvorschläge über die angegebene Kontaktmöglichkeit.*

Hinweis zur Erstellung

Dieses Unterrichtsmaterial wurde mit Unterstützung Künstlicher Intelligenz erstellt und fachlich strukturiert. Trotz sorgfältiger Prüfung können Fehler oder Unvollständigkeiten enthalten sein. Hinweise bitte an die Autorin Christine Janischek melden.

Inhaltsverzeichnis

Die Einträge führen direkt zu den Kapiteln und Hauptabschnitten. In den Überschriften bringt dich „↶ Inhalt“ hierher zurück.

- 01 Start-Prompts für deinen KI-Coach**
- 02 Unser Ziel: ein nachvollziehbarer Lernagent**
- 03 Von der View zur technischen Umsetzung**
- 04 Orientierung für die Lernenden**

- 05 Vom Pythonprogramm zur Datenanalyse**
- 06 CSV-Dateien verstehen und sicher einlesen**
- 07 Daten mit pandas untersuchen und visualisieren**
- 08 Orientierung: Vier Fragen - vier Verfahren**
- 09 Vom Datensatz zum Machine-Learning-Modell**
- 10 Regression: Eine Zahl vorhersagen**
- 11 Klassifikation: Entscheidungsbaum und k-NN**
- 12 Clustering mit k-Means**
- 13 Der rote Faden**
- 14 Modelle prüfen statt Ergebnissen glauben**
- 15 Neuronale Netze, Sprachmodelle und Verantwortung**
- 16 Teil B · BPE 7: KI-Systeme in betrieblichen Situationen**
- 17 Große Datenmengen sichtbar machen**
- 18 Ein reproduzierbarer ML-Workflow mit Bibliotheken**
- 19 KI-Werkzeuge für BWL-Probleme auswählen**
- 20 Abschlussprojekt: einen verantworteten Lernagenten entwickeln**
- 21 Abschluss: Begriffe sicher verwenden**

Start-Prompts für deinen KI-Coach [↩ Inhalt](#)

Diese Prompts helfen dir beim Denken und Lernen. Lass dir nicht einfach eine fertige Lösung erzeugen. Beschreibe zuerst deinen eigenen Stand und fordere eine Hilfe an, die dich zum nächsten Schritt führt.

Lernsituation	Start-Prompt
Begriff verstehen	„Erkläre mir [Begriff] mit einem einfachen Beispiel aus unserem Lernagenten. Stelle mir danach eine Kontrollfrage.“
Code verstehen	„Erkläre den folgenden Python-Code Zeile für Zeile. Nenne zuerst die Aufgabe des Codes und verrate noch keine fertige Änderung: [Code]“
Fehler untersuchen	„Ich erhalte folgende Fehlermeldung: [Meldung]. Hilf mir mit einer Hypothese und genau einem nächsten Prüfschritt.“
Daten prüfen	„Welche drei Qualitätsprobleme sollte ich in diesem Datensatz zuerst prüfen? Begründe die Reihenfolge: [Spalten/Beschreibung]“
Modell auswählen	„Meine Zielgröße ist [Ziel] und meine Daten sind [Beschreibung]. Stelle mir Fragen, mit denen ich selbst zwischen Regression, Klassifikation und Clustering entscheide.“
Ergebnis beurteilen	„Prüfe meine Deutung kritisch. Trenne: Was zeigen die Daten, was vermute ich und welche Information fehlt? [Deutung]“
Transfer zum Lernagenten	„Hilf mir, aus diesem Mockup die benötigten Eingaben, Ausgaben, Regeln und Testfälle abzuleiten. Beginne mit Fragen, nicht mit Programmcode.“
Prüfregel: Übernimm keine Antwort ungeprüft. Vergleiche sie mit Skript, Datensatz und eigenem Programm. Frage nach Begründungen und teste Vorschläge an kleinen Beispielen.	

Unser Ziel: ein nachvollziehbarer Lernagent

↔ Inhalt

Wir beginnen mit dem sichtbaren Ergebnis. Die Oberfläche zeigt eine Empfehlung für die nächste Aufgabe. Anschließend entwickeln wir Schritt für Schritt, was hinter dieser Anzeige geschehen muss.

Startauftrag

Betrachte die Oberfläche zunächst wie eine Nutzerin oder ein Nutzer. Beschreibe, was du siehst, welche Entscheidung angeboten wird und welche Fragen dabei entstehen.

Ansicht für die Lernenden



Mockup: sichtbare Lernagent-View als Ausgangspunkt der Entwicklung.

Die Oberfläche untersuchen

1. Welche Information ist die eigentliche Empfehlung?
2. Welche Daten werden zur Begründung angezeigt?
3. Welche Handlungsmöglichkeiten besitzen die Lernenden?
4. Welche zusätzliche Erklärung würdest du vor dem Start benötigen?
5. Welche Entscheidung sollte der Lernagent nicht allein treffen?

Meine ersten Beobachtungen

Merksatz

Eine gute View zeigt nicht nur ein Ergebnis. Sie macht deutlich, worauf es beruht, was als Nächstes möglich ist und wo eine menschliche Prüfung angefordert werden kann.

Erster Gestaltungsauftrag

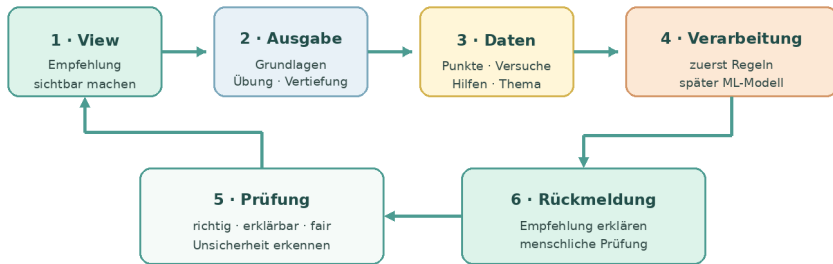
Skizziere eine eigene Variante der Lernagent-Ansicht. Sie muss eine Empfehlung, eine kurze Begründung, einen nächsten Schritt und eine Prüfmöglichkeit enthalten.

Von der View zur technischen Umsetzung

↪ Inhalt

Nun betrachten wir dieselbe Oberfläche aus Sicht der Entwicklung. Wir gehen rückwärts vom sichtbaren Ergebnis zu den benötigten Daten und Verfahren. Einige Felder kannst du schon ausfüllen; andere werden erst in späteren Kapiteln geklärt.

Was muss hinter der View entwickelt werden?



Die View ist Ausgangspunkt und Prüfstein der gesamten Entwicklung.

Mockup: Outside-in-Entwicklung vom sichtbaren Ergebnis zur technischen Verarbeitung und zurück.

Schritt	Leitfrage	erste Antwort / später ergänzen
1 · Anforderung	Was soll die Anwendung für Lernende leisten?	eine passende nächste Aufgabe empfehlen
2 · Ausgabe	Welche Ergebnisse kann die View anzeigen?	Grundlagen · Übung · Vertiefung
3 · Daten	Welche Informationen werden dafür benötigt?	Punkte · Versuche · Hilfen · Thema
4 · Verarbeitung	Wie entsteht aus Daten eine Empfehlung?	zunächst Regeln · später: _____
5 · Prüfung	Woran erkennen wir eine brauchbare Empfehlung?	richtig · erklärbar · fair · _____
6 · Rückmeldung	Was geschieht bei Unsicherheit oder Widerspruch?	_____ —

Unsere Entwicklungsfragen

- Wie speichern wir Lerndaten so, dass Python sie lesen kann?
- Wie prüfen wir, ob die Daten vollständig und plausibel sind?
- Welche Informationen werden Eingaben und welche Information wird Ausgabe?
- Welches Verfahren passt zu einer Zahlenvorhersage, einer Klasse oder unbekannten Gruppen?
- Wie messen wir Fehler und wie gehen wir mit Unsicherheit um?

- Wie schützen wir personenbezogene Daten und vermeiden vorschnelle Urteile?

Der rote Faden

CSV und pandas liefern die Datenbasis. Features und Labels beschreiben Eingabe und Ziel. ML-Verfahren erzeugen Vorhersagen oder Gruppen. Modellbewertung und Datenschutz prüfen, ob die sichtbare Empfehlung verantwortbar ist.

Fortschrittskarte

Baustein	vor dem Kapitel	nach dem Kapitel
View und Anforderung	☐	☐
CSV-Daten	☐	☐
DataFrame und Visualisierung	☐	☐
Features und Labels	☐	☐
ML-Verfahren	☐	☐
Modellprüfung	☐	☐

Arbeitsregel

Bewahre diese Seite auf. Nach jedem Kapitel ergänzt du mindestens eine technische Antwort und eine neue Prüffrage. So wächst aus der View schrittweise ein begründetes System.

Orientierung für die Lernenden [↩ Inhalt](#)

Du hast vor diesem Kurs bereits mit grundlegenden Bausteinen der imperativen Programmierung gearbeitet. Machine Learning beginnt deshalb nicht noch einmal bei null. Neu ist, dass Programme nun Daten aus Dateien übernehmen, tabellarisch untersuchen und daraus Modelle entwickeln.

Die zentrale Brücke

Bekannt: Ein Programm verarbeitet Werte nach festgelegten Anweisungen. Neu: Ein Machine-Learning-Verfahren bestimmt aus Beispieldaten ein Modell, das für neue Daten eine Vorhersage oder Zuordnung erzeugt.

Arbeitsregeln

- Arbeite zuerst mit der Aufgabenstellung und deinen eigenen Überlegungen.
- Öffne die Hilfen in der Reihenfolge Denkipuls → Strategiehilfe → Syntaxhilfe.
- Führe Code nicht nur aus: Erkläre mindestens die Zeile, die du verändert hast.
- Notiere Fehler und Korrekturen. Ein behobener Fehler ist ein sichtbarer Lernschritt.
- Vergleiche erst nach einem eigenen Versuch mit einer Musterlösung.

Zeitplan dieser Fassung

Kapitel	Schwerpunkt	Richtzeit	Lernprodukt
1	Python aktivieren und Datenlogik verstehen	1 UStd.	Regelbasierter Mini-Lernagent
2	CSV-Dateien verstehen und einlesen	2 UStd.	Geprüfte CSV-Datei
3	DataFrame untersuchen und visualisieren	2 UStd.	Datensteckbrief mit Grafik

Offlinepaket und verbindliche Aufgaben-IDs

Alle Pflichtaufgaben können ohne GitHub bearbeitet werden. Öffne zuerst ``02_Schuelermaterial/ML_Offlineprojekt/README.md``. Für Thonny stehen Python-Dateien bereit; inhaltlich gleiche Notebook-Fassungen können alternativ verwendet werden. Die Aufgaben ML-K01-A01 bis ML-K09-A01 sichern BPE 6 im ersten Schuljahr, ML-K10-A01 bis ML-K13-A01 sichern BPE 7 im zweiten Schuljahr.

Vorwissenscheck

- Ich kann einer Variablen einen Wert zuweisen.
- Ich kann Werte in einer Liste erkennen und verändern.
- Ich kann eine if-Bedingung in eigenen Worten erklären.

- Ich kann eine for-Schleife nachvollziehen.
- Ich kann eine Funktion aufrufen und ihren Rückgabewert verwenden.

Wichtig

Wenn du nicht alles sicher kannst, ist das kein Ausschluss vom ML-Kurs. Nutze in Kapitel 1 den Wiederholungspfad und dokumentiere, welcher Baustein wieder klar geworden ist.

1. Vom Pythonprogramm zur Datenanalyse ↵ Inhalt

Kapitelroute

BKW11 · Woche 25 · ca. 1 Unterrichtsstunde · Voraussetzung: Variablen, Listen, Funktionen, Schleifen und Bedingungen

Das kannst du danach

- bekannte Python-Bausteine in einem Datenprogramm wiedererkennen,
- Eingabe, Verarbeitung und Ausgabe eines Programms kennzeichnen,
- regelbasierte und datenbasierte Entscheidungen unterscheiden,
- erklären, warum ein Lernagent nicht mit einer einzigen starren Regel auskommt.

Startfrage

Eine Schülerin erreicht 5 von 10 Punkten. Soll sie automatisch eine Grundlagenaufgabe erhalten? Welche zusätzliche Information könnte die Entscheidung verändern?

1.1 Ein bekanntes Pythonprogramm

Das folgende Programm enthält keine neue Python-Syntax. Es verwendet eine Liste, eine Funktion, eine Bedingung, eine Schleife und eine Ausgabe. Neu ist die fachliche Frage: Wie wird aus Lerndaten eine Empfehlung?

```
punktzahlen = [8, 5, 10, 6]

def waehle_aufgabe(punkte):
    if punkte < 6:
        return 'Grundlagen wiederholen'
    return 'Vertiefungsaufgabe bearbeiten'

for wert in punktzahlen:
    print(wert, waehle_aufgabe(wert))
```

Arbeitsauftrag 1: Code lesen

6. Markiere die Liste, den Funktionskopf, die Bedingung, beide Rückgabewerte und den Funktionsaufruf.
7. Erkläre Zeile für Zeile, was das Programm macht.
8. Notiere die vier Ausgaben, bevor du den Code ausführst.
9. Führe den Code aus und vergleiche Prognose und tatsächliche Ausgabe.

Denkimpuls

Welche Werte durchlaufen die Schleife nacheinander? Welcher Wert wird jeweils an die Funktion übergeben?

Strategiehilfe

Arbeite mit einer Spurentabelle: aktueller Wert → Ergebnis der Bedingung → Rückgabewert.

aktueller Wert	Ist wert < 6?	Rückgabewert
8		
5		
10		
6		

Meine Erklärung

1.2 Eine Regel erweitern

Die erste Version unterscheidet nur zwei Fälle. Nun sollen drei Lernwege möglich sein: Grundlagen, Übung und Vertiefung.

Grundlagenaufgabe

Erweitere die Funktion so, dass bei weniger als 6 Punkten Grundlagen, bei 6 bis 8 Punkten eine Übung und ab 9 Punkten eine Vertiefung empfohlen wird.

```
def waehle_aufgabe(punkte):
    # Ergänze hier eine dreistufige Entscheidung.
    # Tipp: Beginne mit dem kleinsten Wertebereich.
    ...
```

Denkimpuls

Welche drei Wertebereiche dürfen sich nicht überschneiden und keine Lücke enthalten?

Strategiehilfe

Schreibe zuerst drei Sätze: Wenn ..., dann Übertrage sie anschließend in Python.

Syntaxhilfe

Für eine zweite Bedingung innerhalb derselben Auswahl verwendest du `elif` . Der letzte verbleibende Fall kann mit `else` behandelt werden.

Testfälle planen

Gute Tests prüfen nicht nur typische Werte, sondern auch Grenzen. Trage zuerst die erwartete Ausgabe ein und führe dann den Test aus.

Punkte	Warum wichtig?	Erwartete Empfehlung	Tatsächliche Empfehlung
0	kleinster möglicher Wert		
5	Grenze vor Übung		
6	erster Übungswert		

Punkte	Warum wichtig?	Erwartete Empfehlung	Tatsächliche Empfehlung
8	letzter Übungswert		
9	erster Vertiefungswert		
10	größter möglicher Wert		

1.3 Regelbasiert ist noch nicht Machine Learning

Die Funktion wurde vollständig von einem Menschen festgelegt. Das Programm lernt nichts aus früheren Fällen. Es führt die Bedingungen exakt aus – auch dann, wenn sie fachlich zu grob sind.

Regelbasiertes Programm	Machine-Learning-Modell
Menschen formulieren Regeln ausdrücklich.	Ein Verfahren bestimmt Modellparameter aus Beispieldaten.
Entscheidungsweg ist häufig direkt sichtbar.	Entscheidungsweg kann komplexer und schwerer erklärbar sein.
Neue Regel erfordert Programmänderung.	Neues Training kann das Modell verändern.
Qualität hängt von Regeln und Eingaben ab.	Qualität hängt von Daten, Verfahren, Training und Prüfung ab.

Merksatz

Nicht jedes Programm, das eine Empfehlung ausgibt, verwendet Künstliche Intelligenz. Eine if-Bedingung bleibt eine feste Regel.

1.4 Transfer: Was müsste ein Lernagent zusätzlich wissen?

Zwei Lernende können dieselbe Punktzahl erreichen und trotzdem unterschiedliche Unterstützung benötigen. Eine Person hat nur einen Flüchtigkeitsfehler gemacht; eine andere hat mehrere Hilfen verwendet und die Aufgabe noch nicht verstanden.

ML-K01-A01 · Arbeitsauftrag 2: Merkmale sammeln und prüfen

Material: `notebooks/01\_einfuehrung.ipynb` oder
programme/01_einfuehrung.py` · Abgabe: `abgaben/ML-K01-A01` · Prüfnachweis: Ergebnis, Deutung, mindestens ein Test- oder Grenzfall.

10. Sammle fünf mögliche Informationen für eine Lernempfehlung.
11. Kennzeichne, welche Information wirklich für den Lernzweck erforderlich ist.
12. Streiche Informationen, die zwar verfügbar, aber nicht notwendig oder unangemessen wären.
13. Formuliere drei starre Regeln und notiere jeweils einen Fall, in dem die Regel scheitern könnte.

mögliche Information	für Lernzweck nötig?	Begründung / Risiko
Punktzahl		
Anzahl der Versuche		
verwendete Hilfen		
Bearbeitungszeit		
eigener Vorschlag		

Lernagent-Transfer

Unser späterer Lernagent soll keine endgültigen Urteile über Lernende fällen. Er soll auf Basis weniger, erklärbarer Daten eine nächste Übung vorschlagen oder bei Unsicherheit eine menschliche Prüfung anfordern.

Checkpoint Kapitel 1

- Ich kann den Beispielcode ohne Ausführen erklären.
- Ich kann Regeln und gelerntes Modell unterscheiden.
- Ich kenne mindestens eine Grenze einer reinen Punktzahl-Regel.
- Ich kann begründen, warum Datenminimierung bereits bei der Planung beginnt.

Lernjournal

Welcher Python-Baustein war noch unsicher? Was ist jetzt klarer?

Offline-Praxis

Öffne ``notebooks/01_einfuehrung.ipynb``. Führe die Zellen nacheinander aus und verändere mindestens zwei Werte. Speichere das Notebook unter einem neuen Namen, damit die Ausgangsversion erhalten bleibt.

2. CSV-Dateien verstehen und sicher einlesen ↩ Inhalt

Kapitelroute

BKWI1 · Woche 25 · ca. 2 Unterrichtsstunden · neuer Kompetenzbereich
· Grundlage für alle folgenden ML-Kapitel

Das kannst du danach

- den Aufbau einer CSV-Datei erklären,
- Kopfzeile, Datensatz, Feld und Trennzeichen unterscheiden,
- eine CSV-Datei vor dem Import fachlich prüfen,
- relative Dateipfade nachvollziehen,
- Daten mit pandas einlesen und den Import kontrollieren.

Startfrage

Eine CSV-Datei lässt sich in Excel öffnen. Warum müssen wir trotzdem verstehen, dass sie eigentlich eine Textdatei ist?

2.1 Eine Tabelle ohne Formatierung

CSV bedeutet „Comma-Separated Values“. In der Praxis kann statt eines Kommas auch ein Semikolon verwendet werden. Jede Zeile entspricht meist einem Datensatz, jedes getrennte Feld einer Spalte. Farben, Formeln, Zellrahmen und mehrere Tabellenblätter werden nicht gespeichert.

```
lernender_id,thema,versuche,hilfen,bearbeitungszeit_min,punkte
101,Schleifen,2,1,18,8
102,Schleifen,4,3,31,5
103,Funktionen,1,0,14,10
```

Begriff	Bedeutung	Beispiel
Kopfzeile	Namen der Spalten	lernender_id, thema, ...
Datensatz	zusammengehörige Werte einer Zeile	102, Schleifen, 4, 3, 31, 5
Feld	ein einzelner Wert	31
Trennzeichen	grenzt Felder voneinander ab	, oder ;

ML-K02-A01 · Arbeitsauftrag 1: CSV-Datei lesen

Material: `notebooks/02\_daten\_erkunden.ipynb` und `notebooks/daten/lerndaten\_fehlerhaft.csv` · Abgabe: `abgaben/ML-K02-A01` · Prüfnachweis: Ergebnis, Deutung, mindestens ein Test- oder Grenzfall.

14. Wie viele Datensätze enthält das Beispiel?
15. Wie viele Spalten besitzt es?
16. Welche Spalten enthalten Text, welche Zahlen?
17. Welche Zeile beschreibt die höchste Punktzahl?
18. Warum ist `lernender\_id` keine Rechenzahl, obwohl sie aus Ziffern besteht?

Merksatz

Die technische Darstellung eines Wertes entscheidet nicht allein über seine fachliche Bedeutung. Eine ID dient der Zuordnung und wird normalerweise nicht addiert oder gemittelt.

2.2 Vor dem Import prüfen

Ein erfolgreicher Import bedeutet nur, dass Python etwas lesen konnte. Vor jeder Auswertung muss geprüft werden, ob die Daten fachlich und technisch so angekommen sind, wie sie gemeint waren.

Prüfroutine

19. Datei finden: Liegt sie wirklich am erwarteten Ort?
20. Als Text ansehen: Welche Kopfzeile und welches Trennzeichen werden verwendet?
21. Stichprobe lesen: Wirken einige Zeilen vollständig und plausibel?
22. Datentypen prüfen: Wurden Zahlen als Zahlen und Kategorien als Text erkannt?
23. Fehlende Werte suchen: Sind Felder leer oder mit Sonderzeichen gefüllt?
24. Wertebereiche prüfen: Sind beispielsweise Punkte nur zwischen 0 und 10 möglich?

Fehlerdetektiv

Eine Datei enthält `7,5` als Dezimalzahl und verwendet ebenfalls Kommas als Feldtrenner. Erkläre, warum diese Kombination ohne zusätzliche Regeln problematisch sein kann.

2.3 Dateipfade verstehen

Ein relativer Pfad wird ausgehend vom aktuellen Arbeitsort interpretiert. Das Notebook und die CSV-Datei können in unterschiedlichen Ordnern liegen. Deshalb muss der Pfad zur tatsächlichen Ordnerstruktur passen.

```
projekt/  
├─ notebooks/  
│   └─ 02_daten_erkunden.ipynb  
└─ notebooks/daten/  
    └─ haeuser.csv
```

Befindet sich das Notebook im Ordner `notebooks`, lautet der relative Pfad zur Datei im Unterordner `daten` normalerweise `daten/haeuser.csv`. Wird das Programm dagegen aus dem Projekt-

Hauptordner gestartet, kann `notebooks/daten/haeuser.csv` erforderlich sein.

Denkimpuls

Von welchem Ordner aus wird dein Programm oder Notebook tatsächlich ausgeführt?

Strategiehilfe

Prüfe zuerst den Arbeitsordner, dann die Ordnerstruktur und baue daraus den Pfad schrittweise.

Syntaxhilfe

Mit `from pathlib import Path` und `print(Path.cwd())` kannst du den aktuellen Arbeitsordner anzeigen.

```
from pathlib import Path
print(Path.cwd())
print(Path('notebooks/daten/haeuser.csv').exists())
```

2.4 CSV mit pandas einlesen

pandas ist eine Python-Bibliothek für tabellarische Daten. Mit dem üblichen Kurznamen `pd` kann eine CSV-Datei in einen DataFrame geladen werden. Ein DataFrame ähnelt einer Tabelle, besitzt aber zusätzliche Funktionen zum Prüfen, Filtern, Berechnen und Visualisieren.

```
import pandas as pd

daten = pd.read_csv('notebooks/daten/haeuser.csv')
print(daten.head())
```

Code in vier Fragen lesen

Codebestandteil	Leitfrage
import pandas as pd	Welche Bibliothek wird unter welchem Kurznamen eingebunden?
pd.read_csv(...)	Welche Funktion liest die Datei ein?
daten = ...	Unter welchem Namen wird der DataFrame gespeichert?

Codebestandteil	Leitfrage
<code>daten.head()</code>	Welche Stichprobe wird ausgegeben?

Arbeitsauftrag 2: Import durchführen

25. Öffne ``haeuser.csv`` zunächst als Text und notiere Spaltennamen und Trennzeichen.
26. Führe den Import aus.
27. Lasse die ersten fünf und anschließend die letzten drei Zeilen anzeigen.
28. Vergleiche die Zahl der sichtbaren Zeilen mit der tatsächlichen Dateigröße.
29. Verändere den Pfad absichtlich. Lies die Fehlermeldung und stelle den korrekten Zustand wieder her.

Fehler sind Daten

Dokumentiere bei einem Fehler: Fehlermeldung, vermutete Ursache, durchgeführte Prüfung und Lösung. Diese Viererstruktur ist später auch für Modellfehler hilfreich.

Mein Fehlerprotokoll

Fehlermeldung / Beobachtung	vermutete Ursache	Prüfung	Lösung

2.5 Trennzeichen und fehlende Werte

```
daten_semikolon = pd.read_csv('lerndaten.csv', sep=';')
print(daten_semikolon.head())
print(daten_semikolon.isna().sum())
```

Wenn das falsche Trennzeichen verwendet wird, kann der gesamte Inhalt einer Zeile als eine einzige Spalte erscheinen. Leere Felder werden häufig als ``NaN`` angezeigt. Vor einer automatischen Bereinigung muss geklärt werden, warum ein Wert fehlt.

Achtung

Fehlende Daten dürfen nicht gedankenlos durch 0 ersetzt werden. Null Punkte, keine Hilfen und „nicht erfasst“ haben unterschiedliche Bedeutungen.

Arbeitsauftrag 3: Datenqualität provozieren

30. Erstelle eine Kopie einer kleinen CSV-Datei.
31. Entferne einen Wert, vertausche ein Trennzeichen und schreibe ein Wort in eine Spaltenüberschrift.
32. Importiere die veränderte Datei.
33. Beschreibe für jeden Fehler, ob der Import scheitert oder erst die spätere Auswertung problematisch wird.
34. Stelle die Originaldatei wieder her.

2.6 Lernagent-Transfer: eine datensparsame CSV planen

Für den Unterricht wird ein künstlich erzeugter Lerndatensatz verwendet. Er enthält keine echten Namen und keine realen Leistungsprofile. Die Daten müssen trotzdem so aufgebaut sein, dass die spätere Fragestellung beantwortet werden kann.

Spalte	fachliche Bedeutung	Datentyp	zulässige Werte / Prüfung
lerner_id	pseudonyme Zuordnung	Text oder Ganzzahl	eindeutig, keine Klarnamen
thema	bearbeitetes Thema	Text	vorgegebene Themenliste
versuche	Anzahl der Versuche	Ganzzahl	mindestens 1
hilfen	genutzte Hinweise	Ganzzahl	0 bis Anzahl verfügbarer Hilfen
bearbeitungszeit_min	Arbeitszeit in Minuten	Zahl	positiv, Ausreißer prüfen
punkte	Ergebnis	Zahl	0 bis 10

Transferauftrag

Lege zehn künstliche Datensätze an. Baue zwei bewusst auffällige Werte ein und tausche die Datei mit einer Partnergruppe. Die Partnergruppe soll die Auffälligkeiten mithilfe der Prüfroutine finden.

Checkpoint Kapitel 2

- Ich kann CSV-Struktur und Trennzeichen erklären.

- Ich kann einen relativen Pfad prüfen.
- Ich kann eine CSV-Datei mit pandas einlesen.
- Ich kontrolliere nach dem Import Form, Spalten und Datentypen.
- Ich unterscheide fehlend, null und nicht zutreffend.

Lernjournal

Offline-Praxis

Verwendete Dateien: `notebooks/daten/haeuser.csv`, `haeuser\_80.csv`, `studierende\_noten.csv` und `notebooks/02\_daten\_erkunden.ipynb`.

3. Daten mit pandas untersuchen und visualisieren [↩ Inhalt](#)

Kapitelroute

BKWI1 · Woche 25 · ca. 2 Unterrichtsstunden · BPE 6.2 · Vorbereitung der ML-Verfahren

Das kannst du danach

- Aufbau und Inhalt eines DataFrames systematisch beschreiben,
- Spalten auswählen und einfache Kennzahlen berechnen,
- fehlende und auffällige Werte erkennen,
- eine Visualisierung aus einer fachlichen Frage ableiten,
- Beobachtung, Interpretation und Ursache unterscheiden.

Startfrage

Ein Datensatz besitzt 500 Zeilen und keine leeren Felder. Welche weiteren Prüfungen sind nötig, bevor daraus ein Modell trainiert wird?

3.1 Der Datensteckbrief

Bevor gerechnet wird, entsteht ein kurzer Datensteckbrief. Er beantwortet: Woher kommen die Daten? Was beschreibt eine Zeile? Welche Spalten gibt es? Welche Einheiten und Wertebereiche gelten? Welche Zielsetzung verfolgt die spätere Auswertung?

Befehl	liefert	Prüffrage
<code>daten.head()</code>	erste Zeilen	Sehen Spalten und Beispiele plausibel aus?
<code>daten.shape</code>	Zeilen und Spalten	Ist die erwartete Größe vorhanden?
<code>daten.columns</code>	Spaltennamen	Sind Namen vollständig und eindeutig?
<code>daten.info()</code>	Datentypen und Belegung	Gibt es fehlende Werte oder falsche Typen?
<code>daten.describe()</code>	statistische Kennzahlen	Sind Wertebereiche und Verteilungen plausibel?

```
print(daten.head())
print(daten.shape)
print(daten.columns)
daten.info()
print(daten.describe())
```

Merksatz

Ein DataFrame ist technisch lesbar, wenn pandas ihn laden kann. Er ist fachlich brauchbar, wenn Bedeutung, Qualität und Grenzen der Daten verstanden und dokumentiert sind.

3.2 Spalten auswählen und Kennzahlen berechnen

```
punkte = daten['punkte']
print(punkte.min())
print(punkte.max())
print(punkte.mean())
print(punkte.median())
```

Mittelwert und Median beantworten ähnliche, aber nicht identische Fragen. Der Mittelwert reagiert stärker auf extreme Werte. Deshalb sollte eine Kennzahl nie ohne Kenntnis der Verteilung berichtet werden.

ML-K03-A01 · Arbeitsauftrag 1: Datensteckbrief erstellen

Material: `notebooks/02\_daten\_erkunden.ipynb` und `notebooks/daten/lerndaten\_lernagent.csv` · Abgabe: `abgaben/ML-K03-A01` · Prüfnachweis: Ergebnis, Deutung, mindestens ein Test- oder Grenzfall.

35. Notiere Datenquelle und fachliche Bedeutung einer Zeile.
36. Bestimme Anzahl der Zeilen und Spalten.
37. Ordne jeder Spalte eine fachliche Bedeutung und einen erwarteten Datentyp zu.
38. Prüfe fehlende Werte und auffällige Minima oder Maxima.
39. Berechne für eine sinnvolle Zahlenspalte Minimum, Maximum, Mittelwert und Median.
40. Formuliere zwei offene Fragen, die der Datensatz nicht beantworten kann.

Merkmal	Ergebnis / Beobachtung
Datenquelle	
Eine Zeile beschreibt	
Anzahl Zeilen / Spalten	
fehlende Werte	
auffällige Werte	
Grenzen des Datensatzes	

Denkimpuls

Welche Kennzahl würde sich besonders verändern, wenn ein extrem großer Wert ergänzt wird?

Strategiehilfe

Vergleiche Mittelwert und Median und suche danach gezielt nach kleinen oder großen Ausreißern.

Syntaxhilfe

``daten['spalte'].min()`, `.max()`, `.mean()` und `.median()`.`

3.3 Erst die Frage, dann das Diagramm

Fragestellung	geeignete Darstellung	Beispiel
Wie sind Zahlenwerte verteilt?	Histogramm	Verteilung der Bearbeitungszeiten
Wie unterscheiden sich Kategorien?	Balkendiagramm	mittlere Punkte nach Thema
Hängen zwei Zahlenwerte zusammen?	Scatterplot	Bearbeitungszeit und Punkte
Wo liegen Ausreißer und Quartile?	Boxplot	Punkte je Themenbereich

Ein Diagramm ist keine Dekoration. Es ist eine visuelle Antwort auf eine vorher formulierte Frage. Titel, Achsen, Einheiten und gegebenenfalls Legende müssen die Aussage verständlich machen.

```
import matplotlib.pyplot as plt

daten['punkte'].plot(kind='hist', bins=6)
plt.title('Verteilung der erreichten Punkte')
plt.xlabel('Punkte von 0 bis 10')
plt.ylabel('Anzahl der Bearbeitungen')
plt.show()
```

Arbeitsauftrag 2: Visualisierung mit Aussage

41. Formuliere eine Frage zur Verteilung einer Zahlenspalte.
42. Wähle das Diagramm und begründe die Wahl.
43. Erstelle die Grafik mit Titel, Achsenbeschriftungen und Einheit.
44. Notiere eine direkt sichtbare Beobachtung.
45. Ergänze eine mögliche Interpretation.
46. Formuliere eine Aussage, die aus der Grafik ausdrücklich nicht abgeleitet werden darf.

Baustein	Dein Text
Fragestellung	

Baustein	Dein Text
Begründete Diagrammwahl	
Beobachtung	
vorsichtige Interpretation	
nicht zulässige Schlussfolgerung	

3.4 Zusammenhang ist nicht Ursache

Ein Scatterplot kann zeigen, dass zwei Merkmale gemeinsam steigen oder fallen. Daraus folgt noch nicht, dass das eine Merkmal das andere verursacht. Bei Lerndaten können Vorwissen, Aufgabenschwierigkeit, Unterbrechungen oder Messfehler die sichtbare Beziehung beeinflussen.

Beispiel

Wenn längere Bearbeitungszeiten mit niedrigeren Punktzahlen auftreten, beweist das nicht, dass langes Arbeiten schlechte Ergebnisse verursacht. Schwierige Aufgaben könnten gleichzeitig mehr Zeit benötigen und zu niedrigeren Punktzahlen führen.

Arbeitsauftrag 3: Aussagen sortieren

47. Markiere als Beobachtung: „Bei den sichtbaren Datenpunkten treten hohe Bearbeitungszeiten häufiger zusammen mit niedrigeren Punktzahlen auf.“
48. Markiere als unbewiesene Ursache: „Langes Arbeiten verursacht schlechte Leistungen.“
49. Formuliere zwei alternative Erklärungen.
50. Nenne zusätzliche Daten, die zur Prüfung hilfreich wären.

3.5 Gruppen bilden und vergleichen

```
mittelwerte = daten.groupby('thema')['punkte'].mean()  
print(mittelwerte)  
mittelwerte.plot(kind='bar')
```

Mit `groupby()` werden Datensätze nach Kategorien zusammengefasst. Ein Gruppenmittelwert kann Unterschiede sichtbar machen, verdeckt aber weiterhin Streuung und Einzelfälle. Deshalb wird die Zahl zusammen mit Gruppengröße und möglichst einer Verteilungsdarstellung betrachtet.

Prüffrage

Wie viele Fälle enthält jede Gruppe? Ein Mittelwert aus zwei Fällen ist anders zu beurteilen als ein Mittelwert aus zweihundert Fällen.

3.6 Transfer: Daten für den Lernagenten untersuchen

Der Lernagent soll später eine nächste Übung vorschlagen. Noch wird kein Modell trainiert. Zuerst wird geprüft, welche Muster sichtbar sind und welche Unsicherheiten bestehen.

Transferauftrag

51. Lade den künstlichen Lerndatensatz aus Kapitel 2.
52. Erstelle einen vollständigen Datensteckbrief.
53. Untersuche die Verteilung der Punkte.
54. Erstelle einen Scatterplot aus Bearbeitungszeit und Punkten.
55. Vergleiche mittlere Punktzahlen nach Thema.
56. Formuliere drei Beobachtungen, zwei mögliche Erklärungen und mindestens zwei Grenzen.
57. Entscheide, welche zusätzliche Information für eine Lernempfehlung besonders hilfreich wäre.

Datenschutz-Checkpoint

Arbeite ausschließlich mit künstlich erzeugten oder ausdrücklich freigegebenen Daten. Klarnamen, private Angaben und reale individuelle Leistungsprofile gehören nicht in das öffentliche Offlineprojekt.

Ergebnis: Management-Summary in fünf Sätzen

1. Der Datensatz beschreibt ...

2. Besonders auffällig ist ...

3. Eine mögliche, aber noch nicht bewiesene Erklärung ist ...

4. Für eine Lernempfehlung fehlt oder begrenzt ist ...

5. Als nächsten Analyseschritt schlage ich vor ...

Checkpoint Kapitel 3

- Ich beschreibe Daten, bevor ich ein Modell auswähle.
- Ich kann zentrale pandas-Prüfbefehle erklären.
- Ich wähle Diagramme aus Fragestellungen.
- Ich trenne Beobachtung, Interpretation und Ursache.
- Ich beachte Datenminimierung und Datenschutz.

Offline-Praxis

Bearbeite ``aufgaben/ML-K03-A01.md`` mit ``notebooks/02_daten_erkunden.ipynb``. Die Datei ``notebooks/daten/lerndaten_lernagent.csv`` im Offlinepaket wird denselben Prüfweg verwenden.

Orientierung: Vier Fragen - vier Verfahren [↩ Inhalt](#)

In den folgenden Kapiteln steht nicht der möglichst komplizierte Code im Mittelpunkt. Entscheidend ist, aus einer fachlichen Frage das passende Verfahren abzuleiten, die Datenrollen zu bestimmen und Ergebnisse vorsichtig zu beurteilen.

gewünschtes Ergebnis	Datenlage	Verfahren	Beispiel Lernagent
eine Zahl	Label als Zahlenwert vorhanden	Regression	erwartete Punktzahl
eine vorgegebene Klasse	Klassenlabel vorhanden	Entscheidungsbau m oder k-NN	Grundlagen / Übung / Vertiefung
unbekannte Gruppen	kein Ziel-Label	k-Means	ähnliche Arbeitsmuster

Leitregel

Zuerst wird das gewünschte Ergebnis formuliert. Erst danach wird ein Verfahren gewählt. Ein Algorithmus ist kein Selbstzweck.

Gemeinsamer ML-Workflow

1. Problem und gewünschte Ausgabe klären.
2. Datenquelle und Bedeutung einer Zeile beschreiben.
3. Features und gegebenenfalls Label festlegen.
4. Daten prüfen und vorbereiten.
5. Trainings- und Testdaten trennen.
6. Modell trainieren.
7. Vorhersagen oder Cluster erzeugen.
8. Ergebnis fachlich und mit Kennzahlen prüfen.
9. Grenzen und nächste Verbesserung dokumentieren.

Noch keine Magie

`fit()` bedeutet nicht, dass ein Programm menschlich versteht. Die Methode passt Modellparameter an Trainingsdaten an. Ob daraus eine sinnvolle Lösung entsteht, muss anschließend geprüft werden.

4. Vom Datensatz zum Machine-Learning-Modell ↩ Inhalt

Kapitelroute

BKW11 · Woche 26 · ca. 1 Unterrichtsstunde · BPE 6.1/6.2

Das kannst du danach

- Feature, Label, Modell, Training und Vorhersage erklären,
- überwachtes und unüberwachtes Lernen unterscheiden,
- eine Problemfrage in eine ML-Frage übersetzen,
- ungeeignete oder unzulässige Merkmale erkennen.

Startfrage

Welche bekannte Antwort müsste in vergangenen Lerndaten stehen, damit ein Modell später eine Unterstützungsstufe vorhersagen kann?

4.1 Aus Beispielen lernen

Beim überwachten Lernen enthält jeder Trainingsfall Eingaben und eine bekannte richtige oder gewünschte Ausgabe. Aus diesen Beispielen wird ein Modell angepasst. Bei neuen Fällen sind nur die Eingaben bekannt; das Modell schätzt die Ausgabe.

Begriff	Rolle	Beispiel
Feature	Eingabeinformation	Versuche, Hilfen, Bearbeitungszeit
Label	bekannte Zielgröße im Training	Unterstützungsstufe
Modell	gelernte mathematische Abbildung	ordnet neue Eingaben einer Stufe zu
Training	Anpassung an bekannte Beispiele	$\text{fit}(X_{\text{train}}, y_{\text{train}})$
Vorhersage	Anwendung auf neue Eingaben	$\text{predict}(X_{\text{test}})$

Merksatz

Features sind nicht automatisch alle verfügbaren Spalten. Verwendet werden nur fachlich sinnvolle, verfügbare und vertretbare Eingaben.

ML-K04-A01 · Arbeitsauftrag 1: Datenrollen bestimmen

Material: `Aufgabenblatt ML-K04-A01` · Abgabe: `abgaben/ML-K04-A01` · Prüfnachweis: Ergebnis, Deutung, mindestens ein

Test- oder Grenzfall.

Frage	mögliche Features	Label / Ausgabe	Verfahrensart
Wie viele Punkte werden erreicht?			
Welche Unterstützungsstufe passt?			
Welche Lernverläufe ähneln sich?			
Wie hoch wird der Monatsumsatz?			

Denkimpuls

Soll eine Zahl, eine bekannte Kategorie oder eine bislang unbekannte Gruppierung entstehen?

Strategiehilfe

Schreibe zuerst einen Satz: „Aus ... soll ... vorhergesagt beziehungsweise gefunden werden.“

4.2 Überwacht oder unüberwacht?

Regression und Klassifikation gehören in diesem Kurs zum überwachten Lernen: Das Ziel ist in Trainingsdaten bekannt. Clustering ist unüberwacht: Es gibt keine vorgegebenen Zielklassen. Gefundene Gruppen müssen nachträglich fachlich interpretiert werden.

Prüffrage	Ja	Nein
Ist für Trainingsfälle eine Zielantwort vorhanden?	überwachtes Lernen prüfen	unüberwachtes Lernen prüfen
Soll eine Zahl vorhergesagt werden?	Regression	weiterfragen

Prüffrage	Ja	Nein
Soll eine bekannte Kategorie vorhergesagt werden?	Klassifikation	Clustering möglich

4.3 Data Leakage: Wenn die Antwort schon in der Eingabe steckt

Data Leakage entsteht, wenn beim Training Informationen verwendet werden, die bei einer echten späteren Vorhersage noch nicht vorhanden sein dürften oder die Zielantwort praktisch verraten. Das Modell wirkt dann im Test außergewöhnlich gut, scheitert aber im Einsatz.

Beispiel

Soll vor der Bearbeitung der Unterstützungsbedarf geschätzt werden, darf die abschließende Punktzahl nicht als Feature verwendet werden. Sie ist zu diesem Zeitpunkt noch unbekannt.

Arbeitsauftrag 2: Merkmale prüfen

10. Formuliere eine Lernagent-Frage mit genau einer gewünschten Ausgabe.
11. Notiere fünf mögliche Features.
12. Kennzeichne, wann jedes Feature verfügbar ist.
13. Streiche Merkmale, die die Antwort vorwegnehmen oder nicht erforderlich sind.
14. Begründe die verbleibende Auswahl.

Lernagent-Transfer

Lege die erste Modellkarte an: Problem, Zielgruppe, Eingaben, Ausgabe, Zeitpunkt der Empfehlung, ausgeschlossene Daten und menschliche Kontrollstelle.

Checkpoint Kapitel 4

- Ich kann Feature und Label an einem Beispiel zeigen.
- Ich unterscheide überwachtes und unüberwachtes Lernen.
- Ich wähle die Verfahrensart aus der Frage.
- Ich kann Data Leakage an einem Beispiel erklären.

5. Regression: Eine Zahl vorhersagen ↵

Inhalt

Kapitelroute

BKWI1 · Woche 26 · ca. 1,5 Unterrichtsstunden · BPE 6.1

Das kannst du danach

- eine Regressionsfrage erkennen,
- X und y für eine einfache Regression bilden,
- Trainings- und Testdaten trennen,
- eine Vorhersage und Regressionsgerade interpretieren,
- Grenzen einer numerischen Vorhersage benennen.

Startfrage

Kann aus der Wohnfläche ein Hauspreis vorhergesagt werden? Welche wichtigen Informationen fehlen, wenn nur die Wohnfläche verwendet wird?

5.1 Von Punkten zu einer Geraden

Bei einer linearen Regression wird ein möglichst passender linearer Zusammenhang zwischen Eingaben und Zahlenziel bestimmt. Die Gerade ist ein vereinfachtes Modell: Sie fasst ein Muster zusammen, trifft aber nicht jeden beobachteten Wert.

Baustein	Bedeutung im Hausbeispiel	Bedeutung im Lernagenten
X / Features	Wohnfläche	z. B. Versuche und Hilfen

Baustein	Bedeutung im Hausbeispiel	Bedeutung im Lernagenten
y / Label	bekannter Hauspreis	bekannte spätere Punktzahl
Vorhersage	geschätzter Preis	geschätzte Punktzahl
Fehler	Abstand zum echten Preis	Abstand zur echten Punktzahl

5.2 Schrittfolge im Notebook

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression

daten = pd.read_csv('notebooks/daten/haeuser.csv')
X = daten[['groesse_m2']]
y = daten['preis_euro']

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)
modell = LinearRegression()
modell.fit(X_train, y_train)
vorhersagen = modell.predict(X_test)
```

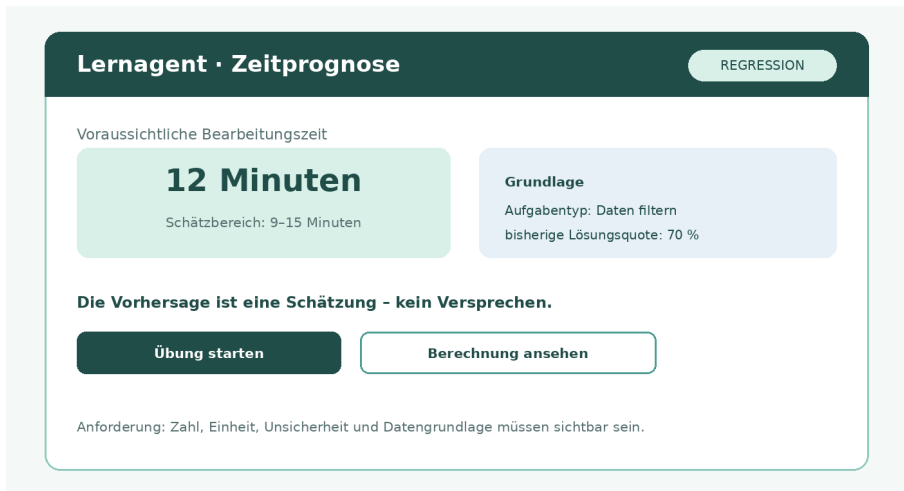
Code-Lesespur

Schritt	Was geschieht?	Warum nötig?
Daten laden		
X und y festlegen		
Train/Test-Split		
Modell erzeugen		
fit		
predict		

Syntaxhilfe

Doppelte eckige Klammern bei `daten[['groesse\_m2']]` halten X zweidimensional. scikit-learn erwartet Features als Tabelle - auch wenn nur eine Spalte verwendet wird.

5.3 Vorhersagen sichtbar machen



Mockup: Die Regressionsausgabe zeigt nicht nur einen Zahlenwert, sondern Einheit, Schätzbereich und Datengrundlage.

```
import matplotlib.pyplot as plt

plt.scatter(X_test['groesse_m2'], y_test, label='echte Preise')
plt.scatter(X_test['groesse_m2'], vorhersagen, label='Vorhersagen')
plt.xlabel('Wohnfläche in m²')
plt.ylabel('Preis in Euro')
plt.legend()
plt.show()
```

Beobachten vor Bewerten

Beschreibe zuerst, wo Vorhersagen nahe an echten Werten liegen und wo große Abstände auftreten. Erst danach wird beurteilt, ob das Modell für den Zweck ausreichend ist.

ML-K05-A01 · Arbeitsauftrag: Regression durchführen

Material: `notebooks/03\_lineare\_regression.ipynb` oder
programme/03_lineare_regression.py · Abgabe: `abgaben/ML-
K05-A01` · Prüfnachweis: Ergebnis, Deutung, mindestens ein
Test- oder Grenzfall.

15. Öffne `03\_lineare\_regression.ipynb`.

16. Kennzeichne Feature und Label.
17. Führe alle Zellen einzeln aus und erkläre ihre Aufgabe.
18. Notiere für fünf Testfälle echten und vorhergesagten Wert.
19. Erstelle oder ergänze die Visualisierung.
20. Formuliere zwei Beobachtungen und zwei Grenzen.

Testfall	echter Wert	Vorhersage	Abweichung	Kommentar
1				
2				
3				
4				
5				

5.4 Transfer zum Lernagenten

Eine geschätzte Punktzahl könnte helfen, Aufgaben passend zu dosieren. Sie darf aber weder als sichere zukünftige Leistung noch als Note behandelt werden. Ziel ist eine vorläufige Unterstützung, die durch tatsächliche Bearbeitung korrigiert wird.

Transferauftrag

Entwirf eine Regression zur Schätzung der nächsten Punktzahl. Lege Features fest, formuliere zwei erwartbare Fehlerquellen und bestimme eine Regel, wann keine automatische Empfehlung ausgegeben werden darf.

Checkpoint Kapitel 5

- Ich erkenne numerische Zielgrößen.
- Ich kann X, y, fit und predict erklären.
- Ich weiß, dass eine Gerade Wirklichkeit vereinfacht.
- Ich kann eine Modellvorhersage vorsichtig formulieren.

Offline-Praxis

Vorhanden: ``notebooks/03_lineare_regression.ipynb``,
``notebooks/daten/haeuser.csv``, ``aufgaben/ML-K05-A01.md``.

6. Klassifikation: Entscheidungsbaum und k-NN ↪ Inhalt

Kapitelroute

BKWI1 · Woche 26 · ca. 1,5 Unterrichtsstunden · BPE 6.1

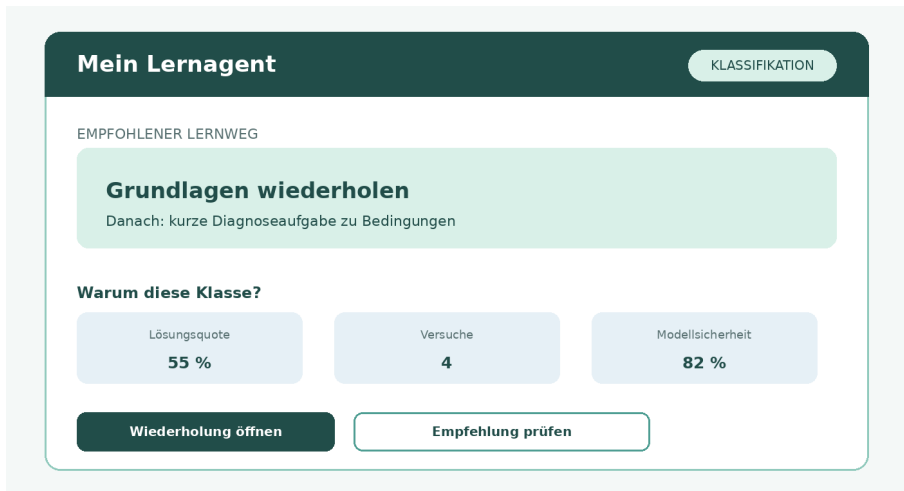
Das kannst du danach

- eine Klassifikationsfrage erkennen,
- einen einfachen Entscheidungsbaum lesen,
- die Nachbarschaftsidee von k-NN erklären,
- beide Verfahren anhand von Erklärbarkeit und Datenanforderungen vergleichen.

Startfrage

Der Lernagent soll zwischen Grundlagen, Übung und Vertiefung wählen. Ist das eine Zahlenvorhersage oder eine Zuordnung zu Klassen?

6.1 Klassen statt Zahlen



Mockup: Aus der sichtbaren Empfehlung lassen sich Zielklasse, Merkmale, Sicherheit und Prüfhandlung ableiten.

Bei einer Klassifikation gehört jeder Trainingsfall zu einer vorgegebenen Klasse. Das Modell lernt aus Features, wie neue Fälle diesen Klassen zugeordnet werden können. Die Klassen müssen fachlich sinnvoll und in den Trainingsdaten zuverlässig gekennzeichnet sein.

Problem	mögliche Klassen	geeignete Features
E-Mail prüfen	Spam / kein Spam	Wörter, Absendermerkmale
Kreditanfrage	genehmigen / prüfen / ablehnen	zulässige Finanzmerkmale
Lernempfehlung	Grundlagen / Übung / Vertiefung	Versuche, Hilfen, bisherige Ergebnisse

Achtung

Eine Klasse ist eine für den Zweck definierte Ausgabe – kein unveränderliches Etikett für einen Menschen.

6.2 Entscheidungsbaum

Ein Entscheidungsbaum teilt Trainingsfälle schrittweise anhand von Merkmalen. An jedem Knoten wird eine Frage geprüft. Die Blätter

enthalten die vorhergesagte Klasse. Kleine Bäume sind häufig gut erklärbar; sehr tiefe Bäume können Einzelfälle auswendig lernen.

```
versuche <= 2?
├─ ja: punkte_bisher >= 8?
│   ├── ja: Vertiefung
│   └── nein: Übung
└─ nein: Grundlagen
```

ML-K06-A01 · Arbeitsauftrag 1: Baum lesen und prüfen

Material: `notebooks/05\_entscheidungsbaum\_knn.ipynb` oder gleichnamiges Python-Programm · Abgabe: `abgaben/ML-K06-A01` · Prüfnachweis: Ergebnis, Deutung, mindestens ein Test- oder Grenzfall.

21. Bestimme die Empfehlung für drei vorgegebene Fälle.
22. Formuliere jeden Pfad als Wenn-dann-Satz.
23. Finde einen Fall, bei dem der Baum zu grob entscheidet.
24. Ergänze keine personenbezogenen Merkmale, sondern eine fachlich sinnvolle Lerninformation.

Fall	Versuche	bisherige Punkte	Pfad	Empfehlung
A	1	9		
B	2	6		
C	4	8		

Denkimpuls

Welche Frage wird zuerst geprüft und welcher Teilbaum bleibt danach noch relevant?

Strategiehilfe

Verfolge genau einen Weg von oben nach unten. Begründe jede Abzweigung mit den Werten des Falls.

6.3 k-Nearest Neighbors: Ähnliche bekannte Fälle

k-NN speichert die Trainingsfälle. Für einen neuen Fall werden die k ähnlichsten Fälle gesucht; bei Klassifikation entscheidet meist die häufigste Klasse in dieser Nachbarschaft. Das Verfahren hängt stark von der Definition von Nähe ab.

Einflussgröße	Warum wichtig?
Auswahl der Features	Nur diese Merkmale bestimmen Ähnlichkeit.
Skalierung	Große Zahlenbereiche können Abstände dominieren.
Wahl von k	Zu kleines k reagiert empfindlich; zu großes k verwischt lokale Muster.
Datenverteilung	Seltene Klassen können überstimmt werden.

Gedankenexperiment

Ein neuer Lernfall hat 3 Versuche und 2 Hilfen. Seine drei nächsten Nachbarn tragen die Klassen `Übung`, `Übung`, `Grundlagen`. Welche Klasse ergibt sich bei $k = 3$? Was könnte sich bei $k = 5$ ändern?

Fachhilfe

„Nah“ bedeutet nicht automatisch pädagogisch ähnlich. Die verwendeten Merkmale und ihre Skalierung legen fest, welche Fälle als Nachbarn gelten.

6.4 Die Verfahren vergleichen

Kriterium	Entscheidungsbaum	k-NN
Grundidee	aufeinanderfolgende Teilungsregeln	Mehrheit ähnlicher Fälle
Erklärbarkeit	bei kleinem Baum meist hoch	über Nachbarfälle erklärbar
Vorbereitung	weniger empfindlich für Skalierung	Skalierung meist wichtig

Kriterium	Entscheidungsbaum	k-NN
Risiko	zu tiefer Baum lernt Einzelfälle	ungeeignete Distanz oder k-Wahl
Vorhersage	schnell nach Training	muss Abstände zu Trainingsfällen berechnen

Transferauftrag

Lernagent-Transfer

Entwerft zu zweit einen kleinen Entscheidungsbaum und ein k-NN-Gedankenmodell für dieselbe Lernempfehlung. Entscheidet anschließend, welches Ergebnis einer Schülerin oder einem Schüler verständlicher erklärt werden kann.

Checkpoint Kapitel 6

- Ich erkenne Klassifikationsprobleme.
- Ich kann einen Baumpfad nachvollziehen.
- Ich kann k-NN ohne Formel erklären.
- Ich kann mindestens einen Unterschied begründen.

Offline-Praxis

Für dieses Kapitel muss das Repository um zwei geführte Mini-Notebooks samt Aufgaben, Lösungen und Tests ergänzt werden: Entscheidungsbaum und k-NN.

7. Clustering mit k-Means ↵ Inhalt

Kapitelroute

BKWI1 · Woche 26 · ca. 1 Unterrichtsstunde · BPE 6.1

Das kannst du danach

- Clustering von Klassifikation unterscheiden,
- die wiederholte Zuordnung bei k-Means erklären,
- den Einfluss von Skalierung und k beschreiben,
- Cluster vorsichtig und nicht stigmatisierend interpretieren.

Startfrage

Können ähnliche Lernverläufe gefunden werden, wenn noch niemand Kategorien wie „sicher“ oder „unsicher“ vergeben hat?

7.1 Gruppen ohne vorgegebenes Label

Beim Clustering gibt es kein Ziel-Label. Ein Algorithmus sucht Strukturen anhand ausgewählter Features. k-Means beginnt mit k Mittelpunkten, ordnet jeden Punkt dem nächsten Mittelpunkt zu und berechnet anschließend neue Mittelpunkte. Dieser Vorgang wird wiederholt.

25. k festlegen: Wie viele Cluster sollen gesucht werden?
26. Startmittelpunkte wählen.
27. Jeden Datenpunkt dem nächsten Mittelpunkt zuordnen.
28. Für jedes Cluster einen neuen Mittelpunkt berechnen.
29. Zuordnung wiederholen, bis sich kaum noch etwas ändert.

Merksatz

Der Algorithmus findet rechnerische Gruppen. Bedeutung und Namen der Gruppen entstehen erst durch fachliche Interpretation.

7.2 Abstand und Skalierung

Wenn ein Feature zwischen 0 und 5 liegt, ein anderes aber zwischen 0 und 10 000, bestimmt das zweite Merkmal den Abstand wesentlich stärker. Deshalb müssen Einheiten und Skalierung vor dem Clustering geprüft werden.

Gedankenexperiment

Feature A ist die Anzahl genutzter Hilfen (0–3), Feature B die Bearbeitungszeit in Sekunden (300–3600). Welches Feature würde einen ungeprüften Abstand wahrscheinlich dominieren?

7.3 Cluster visualisieren und beschreiben

Lernmuster · explorative Ansicht
CLUSTERING

Automatisch gefundene Gruppen – noch keine pädagogischen Kategorien

Gruppe A
hohe Quote · wenige Hilfen

Gruppe B
mittlere Quote · viele Versuche

Gruppe C
wenige Daten · uneindeutig

Erst beschreiben, dann vorsichtig deuten

Mockup: Cluster werden zunächst neutral beschrieben. Eine fachliche Bedeutung entsteht erst durch Prüfung und Interpretation.

```
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler

X = daten[['versuche', 'bearbeitungszeit_min']]
X_skaliert = StandardScaler().fit_transform(X)
modell = KMeans(n_clusters=3, random_state=42, n_init=10)
daten['cluster'] = modell.fit_predict(X_skaliert)
```

Code-Lesespur

Codebestandteil	Bedeutung
X	Features für Ähnlichkeit
StandardScaler	bringt Merkmale auf vergleichbare Größenordnung
n_clusters=3	gesuchte Anzahl der Gruppen
random_state	macht den Start reproduzierbar
fit_predict	lernt Cluster und gibt Zuordnungen zurück

ML-K07-A01 · Arbeitsauftrag: Clustering prüfen

Material: `notebooks/06\_kmeans.ipynb` oder
 programme/06_kmeans.py` · Abgabe: `abgaben/ML-K07-A01` ·
 Prüfnachweis: Ergebnis, Deutung, mindestens ein Test- oder
 Grenzfall.

30. Erstelle eine Punktwolke aus zwei Merkmalen.
31. Führe k-Means mit $k = 2$ und $k = 3$ aus.
32. Färbe Punkte nach Clusterzuordnung.
33. Beschreibe sichtbare Unterschiede, ohne die Gruppen sofort zu bewerten.
34. Formuliere für jedes Cluster eine sachliche, vorläufige Beschreibung.
35. Nenne eine Information, die das Verfahren nicht kennt.

Sprachregel

Verwende Beschreibungen wie „häufige Hilfen bei längerer Bearbeitungszeit“. Vermeide Etiketten wie „schwach“, „faul“ oder „hoffnungslos“.

7.4 Transfer zum Lernagenten

Cluster können Hinweise für unterschiedliche Unterstützungsangebote liefern. Sie dürfen nicht als feste Lerntypen behandelt werden. Eine Person kann sich je nach Thema, Tagesform und Aufgabe anders verhalten.

Transferauftrag

Untersuche künstliche Lernverläufe mit zwei bis drei Features. Schlage für jedes Cluster ein freiwilliges Unterstützungsangebot vor und formuliere zugleich eine Warnung vor einer möglichen Fehlinterpretation.

Cluster	beobachtbare Merkmale	mögliches Angebot	Warnung / Unsicherheit
0			
1			
2			

Checkpoint Kapitel 7

- Ich weiß, dass Clustering kein Label verwendet.
- Ich kann die Schritte von k-Means erklären.
- Ich prüfe Skalierung und Wahl von k.
- Ich beschreibe Cluster sachlich und vorläufig.

Offline-Praxis

Verwende das geführte Notebook ``notebooks/06_kmeans.ipynb`` mit künstlichen Lernagent-Daten. Die Lehrkraftfassung enthält Lösung und Funktionsprüfung.

Der rote Faden ← Inhalt

Wir entwickeln den Lernagenten weiter. Er soll Lernstände aus Daten ableiten und eine passende nächste Übung vorschlagen. Dabei genügt es nicht, dass ein Modell irgendein Ergebnis liefert: Wir müssen prüfen, wie gut, fair, nachvollziehbar und nützlich dieses Ergebnis ist.

Kapitel	Leitfrage	Ergebnis
8	Kann ich dem Modell vertrauen?	Prüfplan und Qualitätsurteil
9	Was leisten neuronale Netze und Sprachmodelle?	verantwortete Einsatzentscheidung
10	Wie werden viele Daten verständlich?	Analyse und Dashboard
11	Wie entsteht ein reproduzierbarer ML-Workflow?	Bibliotheksgestützte Pipeline
12	Welches KI-Werkzeug passt zum BWL-Problem?	begründete Toolentscheidung
13	Wie wird daraus ein Lernagent?	Prototyp, Test und Präsentation

Arbeitsweise

Bearbeite zunächst den Kernauftrag. Öffne Hilfen erst, wenn du sie brauchst. Dokumentiere Annahmen und Entscheidungen: Ein nachvollziehbarer Weg zählt ebenso wie ein korrektes Ergebnis.

Orientierung: vom Modell zur Anwendung

Ein ML-Projekt verläuft nicht geradlinig. Neue Ergebnisse führen oft zurück zu den Daten oder zur Fragestellung. Der folgende Arbeitszyklus begleitet alle Kapitel:

Schritt	Leitfrage	Produkt
1 · Problem	Welche Entscheidung soll unterstützt werden?	klarer Auftrag
2 · Daten	Sind die Daten passend und vertretbar?	Datensteckbrief
3 · Baseline	Was leistet eine sehr einfache Lösung?	Vergleichswert
4 · Modell	Welches Verfahren passt?	trainiertes Modell
5 · Prüfung	Wie gut ist es bei unbekannten Fällen?	Kennzahlen + Fehleranalyse
6 · View	Wie wird das Ergebnis verständlich?	Mockup/Prototyp
7 · Verantwortung	Wer entscheidet und kontrolliert?	Einsatzregeln

Zeitvorschlag

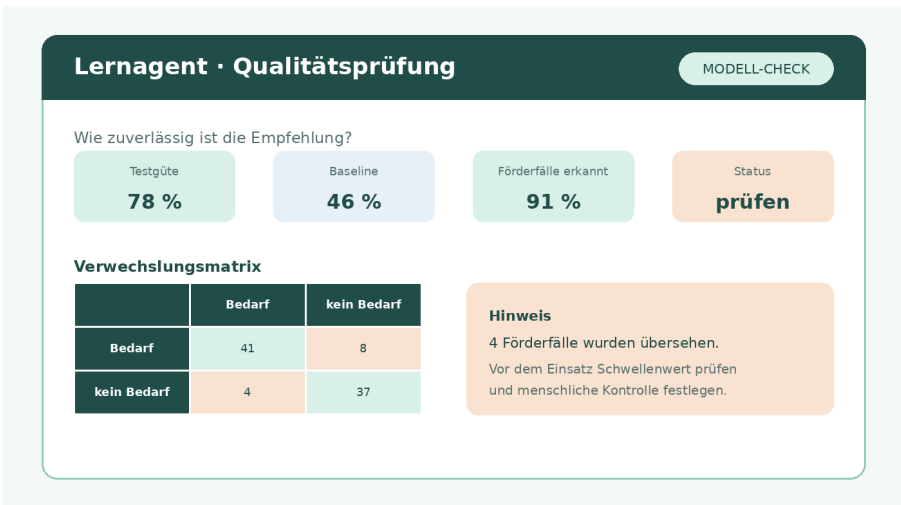
Unterrichtsblock	Schwerpunkt	Kapitel
BPE 6 · Abschluss	Modellgüte, neuronale Netze, Verantwortung	8–9
BPE 7 · Woche 1	große Daten und Visualisierung	10
BPE 7 · Woche 2	Programmbibliotheken und Workflow	11
BPE 7 · Woche 3	BWL-Werkzeugwahl und Transferprojekt	12–13

Wichtig

Der Lernagent trifft keine pädagogische Letztentscheidung. Er liefert einen begründeten Vorschlag. Lehrkraft und Lernende behalten Einsicht, Widerspruchsmöglichkeit und Kontrolle.

8 · Modelle prüfen statt Ergebnissen glauben [← Inhalt](#)

Der Lernagent meldet: „Nächste Empfehlung: CSV filtern.“ Aber woher wissen wir, ob diese Empfehlung zuverlässig ist? Ein Modell kann auf bekannten Daten hervorragend wirken und bei neuen Lernenden trotzdem scheitern. Modellprüfung ist deshalb keine Zusatzaufgabe, sondern Teil der Entwicklung.



Mockup: Die Qualitätsansicht verbindet Kennzahlen mit einer verständlichen Warnung und einer konkreten Prüfhandlung.

Das kannst du nach diesem Kapitel

- Trainings- und Testdaten begründet trennen.
- passende Kennzahlen für Regression und Klassifikation auswählen.
- eine Baseline verwenden und Über- bzw. Unteranpassung erkennen.
- Fehler nicht nur zählen, sondern fachlich und fair bewerten.

8.1 Bekannte und unbekannte Fälle

Beim Training lernt ein Modell Zusammenhänge aus Beispielen. Beim Test erhält es Daten, die es zuvor nicht gesehen hat. Nur so prüfen

wir, ob es verallgemeinert. Die Zielgröße darf beim Training bekannt sein; beim späteren Vorhersagen wird sie vom Modell geschätzt.

Teilmenge	Zweck	Typischer Anteil
Training	Muster lernen und Modellparameter bestimmen	z. B. 80 %
Test	Leistung an unbekannten Fällen prüfen	z. B. 20 %

Stolperstelle: Datenleck

Informationen aus dem Testdatensatz dürfen nicht in die Vorbereitung oder das Training einfließen. Sonst ist das Testergebnis zu optimistisch.

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)
```

random_state sorgt dafür, dass die zufällige Aufteilung reproduzierbar bleibt.

8.2 Kennzahlen passend zur Frage wählen

Die Kennzahl hängt davon ab, was das Modell vorhersagt. Bei einer Regression entsteht eine Zahl, etwa die erwartete Bearbeitungszeit. Bei einer Klassifikation entsteht eine Klasse, etwa „Grundlage wiederholen“ oder „Vertiefung starten“.

Aufgabe	Kennzahl	Bedeutung / Grenze
Regression	MAE	durchschnittlicher absoluter Fehler; gut verständlich
Regression	MSE	große Fehler zählen überproportional stark
Regression	R^2	erklärter Anteil der Streuung; allein nicht ausreichend
Klassifikation	Accuracy	Anteil korrekter Vorhersagen; bei ungleichen Klassen riskant
Klassifikation	Precision	Wie viele positive Meldungen waren richtig?
Klassifikation	Recall	Wie viele tatsächlich positive Fälle wurden gefunden?

Die Verwechslungsmatrix

Angenommen, „Förderbedarf“ ist die positive Klasse. Ein falsch negatives Ergebnis übersieht Förderbedarf; ein falsch positives Ergebnis empfiehlt unnötig eine Wiederholung. Beide Fehler haben unterschiedliche Folgen.

	Tatsächlich: Förderbedarf	Tatsächlich: kein Förderbedarf
Modell: Förderbedarf	richtig positiv	falsch positiv
Modell: kein Förderbedarf	falsch negativ	richtig negativ

Fachliche Entscheidung

Beim Lernagenten kann Recall wichtiger sein als reine Accuracy, wenn übersehener Förderbedarf besonders problematisch ist. Die Kennzahl folgt dem Zweck – nicht umgekehrt.

8.3 Baseline, Überanpassung und Datenqualität

Die Baseline

Eine Baseline ist eine bewusst einfache Vergleichslösung. Bei einer Klassifikation könnte sie immer die häufigste Klasse vorhersagen. Ein komplexeres Modell ist nur sinnvoll, wenn es diese einfache Lösung erkennbar verbessert.

Über- und Unteranpassung

Beobachtung	Deutung	Mögliche Reaktion
Training schlecht, Test schlecht	Unteranpassung: Modell erfasst Muster nicht	bessere Merkmale, anderes Modell
Training sehr gut, Test deutlich schlechter	Überanpassung: Modell merkt sich Besonderheiten	vereinfachen, mehr Daten, regulieren

Beobachtung	Deutung	Mögliche Reaktion
Training und Test ähnlich gut	Hinweis auf brauchbare Verallgemeinerung	Fehler und Teilgruppen prüfen

Mehr Daten sind nicht automatisch bessere Daten

- Repräsentativität: Kommen verschiedene Lerngruppen angemessen vor?
- Vollständigkeit: Fehlen Werte systematisch?
- Aktualität: Passt der Datensatz zur heutigen Lernsituation?
- Messqualität: Bedeutet „Bearbeitungszeit“ wirklich Lernfortschritt?
- Zulässigkeit: Dürfen die Daten für diesen Zweck verarbeitet werden?

Merksatz

Ein präzises Modell auf ungeeigneten Daten bleibt eine ungeeignete Lösung.

ML-K08-A01 · 8.4 Praxisauftrag: Qualitätsprüfung für den Lernagenten

Material: `notebooks/07\_modellpruefung.ipynb` · Abgabe: `abgaben/ML-K08-A01` · Prüfnachweis: Ergebnis, Deutung, mindestens ein Test- oder Grenzfall.

Erstelle einen Prüfplan für ein Modell, das aus Übungsergebnissen eine von drei Empfehlungen erzeugt: „wiederholen“, „weiterüben“, „vertiefen“. Lege Zielgröße, Datenaufteilung, Baseline, Kennzahlen und mindestens drei Fehlerprüfungen fest.

Denkimpuls

Welche Fehlentscheidung hätte für Lernende die stärkste negative Folge?

Strategiehilfe

Beginne mit einer Tabelle: Prüfrage - Methode - akzeptables Ergebnis - Reaktion.

Fach- oder Syntaxhilfe

Nutze bei Klassen nicht nur `accuracy_score`, sondern auch `confusion_matrix` und `classification_report`.

Prüfrage	Methode	Akzeptanzkriterium	Reaktion bei Abweichung
Schlägt das Modell die Baseline?			
Werden Förderfälle erkannt?			
Funktioniert es in Teilgruppen ähnlich?			
Sind Ergebnisse reproduzierbar?			

8.5 Transfer: Fairness und menschliche Kontrolle

Vergleiche die Fehlerquote beispielsweise nach Lernbereich – nicht nach Merkmalen, die für den Zweck unnötig oder besonders sensibel sind. Eine Abweichung beweist noch keine Benachteiligung, ist aber ein Signal für eine genauere Ursachenprüfung.

- Welche Daten benötigt der Lernagent wirklich?
- Welche Erklärung sieht die lernende Person?
- Wie kann sie einer Empfehlung widersprechen?
- Wann muss eine Lehrkraft eingreifen?

Lerncheck

Markiere: ✓ **sicher** · ~ **teilweise** · ? **noch unklar**

- ☐ Ich kann Training und Test unterscheiden.
- ☐ Ich kann eine Kennzahl fachlich begründen.
- ☐ Ich erkenne eine Baseline als notwendigen Vergleich.

☐ Ich kann Grenzen und Risiken eines Modells benennen.

9 · Neuronale Netze, Sprachmodelle und Verantwortung [↩ Inhalt](#)

Neuronale Netze können komplexe Muster in Bildern, Sprache oder großen Datenbeständen erkennen. Sie sind leistungsfähig, aber nicht für jedes Problem die beste Wahl. Für tabellarische Schuldaten kann ein einfaches, erklärbares Modell fachlich sinnvoller sein.

9.1 Vom künstlichen Neuron zum Netz

Ein künstliches Neuron erhält Eingabewerte, gewichtet sie, addiert einen Bias und wendet eine Aktivierungsfunktion an. Viele verbundene Neuronen bilden Schichten.

Baustein	Bedeutung im Lernagent-Beispiel
Eingabe	z. B. Lösungsquote, Bearbeitungszeit, Anzahl Versuche
Gewicht	erlernte Bedeutung eines Merkmals
Bias	verschiebt die Entscheidungsschwelle
Aktivierung	entscheidet, welches Signal weitergegeben wird
Ausgabe	z. B. Wahrscheinlichkeit für eine Empfehlung

Vereinfacht

Training bedeutet: Vorhersage erzeugen, Fehler bestimmen, Gewichte schrittweise so verändern, dass der Fehler kleiner wird.

Wann lohnt sich ein neuronales Netz?

- viele geeignete Beispiele stehen zur Verfügung,
- Muster sind komplex, etwa in Text, Bild oder Ton,
- Rechenaufwand und geringere Erklärbarkeit sind vertretbar,
- ein einfacheres Verfahren erreicht das Ziel nicht ausreichend.

9.2 Was Sprachmodelle tun

Ein Sprachmodell erzeugt Text, indem es auf Basis des bisherigen Kontexts wahrscheinliche nächste Texteinheiten berechnet. Es kann erklären, zusammenfassen und formulieren – besitzt aber keine Garantie auf Wahrheit. Plausibel klingende Falschaussagen bleiben möglich.

Kann unterstützen bei ...	Darf nicht ungeprüft entscheiden über ...
Formulierung einer Lernhilfe	Benotung oder Sanktion
Varianten einer Übungsfrage	Diagnose einer Person
Zusammenfassung eigener Notizen	Weitergabe personenbezogener Daten
Ideen für Feedback	pädagogische Letztentscheidung

Ein hybrider Lernagent

Eine robuste Lösung kann einfache Regeln, ein tabellarisches Modell und ein Sprachmodell kombinieren. Das tabellarische Modell schätzt einen Lernbedarf; das Sprachmodell formuliert eine Erklärung. Vor der Anzeige greifen Datenschutz-, Sicherheits- und Qualitätsregeln.

Schicht	Aufgabe	Kontrolle
Daten	zulässige Merkmale bereitstellen	Minimierung, Pseudonymisierung
Analyse	Lernbedarf schätzen	Testdaten, Kennzahlen, Baseline
Sprache	Hinweis verständlich formulieren	klare Vorgaben, Inhaltsprüfung
View	Empfehlung und Begründung zeigen	Widerspruch und menschliche Freigabe

Merksatz
Flüssige Sprache ist kein Beleg für fachliche Richtigkeit.

Mein Lernagent

UNSICHERES ERGEBNIS

DEINE NÄCHSTE EMPFEHLUNG

Noch keine sichere Empfehlung

Die vorhandenen Ergebnisse widersprechen sich.
Bitte bearbeite zuerst die kurze Diagnoseaufgabe.

Warum wird noch nicht entschieden?

Lösungsquote
8 von 10

Bearbeitungszeit
ungewöhnlich hoch

Datengrundlage
nur 1 Übung

Diagnose starten

Lehrkraft einbeziehen

Mockup: Der Agent darf Unsicherheit zeigen und eine weitere Diagnose oder menschliche Prüfung anfordern.

9.3 Chancen, Risiken und Einsatzregeln

Chance	Risiko	Schutzmaßnahme
individuelle Übungshinweise	vorschnelle Etikettierung	Empfehlung statt Festlegung
schnelle Rückmeldung	falsche oder erfundene Aussage	Quellen und Plausibilität prüfen
Skalierung wiederkehrender Hilfen	Automatisierungsbias	Unsicherheit sichtbar machen
Auswertung vieler Ergebnisse	Datenschutzverletzung	nur erforderliche Daten nutzen
barriereärmere Formulierungen	ungleiche Qualität für Gruppen	Teilgruppen und Randfälle testen

Aktuelle Entwicklungen untersuchen

Technologien ändern sich schnell. Statt eine feste Liste auswendig zu lernen, prüfst du eine aktuelle Entwicklung mit einem belastbaren

Raster: Was ist neu? Welche Quelle belegt es? Welches Problem löst es? Welche Grenzen bleiben? Was bedeutet es für Schule oder Betrieb?

Rechercheauftrag

Untersuche eine aktuelle Entwicklung bei Sprachmodellen oder Lernsystemen. Nutze mindestens zwei belastbare Quellen, notiere Veröffentlichungsdatum und Anbieterinteresse und trenne belegte Aussage, Anbieterbehauptung und eigene Schlussfolgerung.

Denkimpuls

Ist die Aussage messbar oder nur werblich?

Strategiehilfe

Bevorzuge Primärquellen wie technische Dokumentation, Forschungspapier oder offizielle Rechtsquelle.

ML-K09-A01 · 9.4 Fallentscheidung: Darf der Lernagent das?

Material: `Aufgabenblatt ML-K09-A01` · Abgabe: `abgaben/ML-K09-A01` · Prüfnachweis: Ergebnis, Deutung, mindestens ein Test- oder Grenzfall.

Bewerte vier Funktionen: (A) Übung vorschlagen, (B) Freitext automatisch benoten, (C) Lernhilfe formulieren, (D) Risiko für Nichtbestehen prognostizieren. Entscheide jeweils: einsetzen, nur mit Auflagen testen oder nicht einsetzen.

Denkimpuls

Betrachte Nutzen, Schadenshöhe, Datenbedarf, Erklärbarkeit und menschliche Kontrolle.

Strategiehilfe

Formuliere jede Entscheidung im Muster: Wir entscheiden ..., weil ...; das Risiko ... begrenzen wir durch ...

Funktion	Entscheidung	Begründung	Schutzmaßnahme
A			
B			
C			
D			

Mini-Debatte

These: „Wenn ein KI-System genauer als ein Mensch ist, sollte es automatisch entscheiden dürfen.“ Bereite eine Pro- und eine Contra-Position vor. Widerlege anschließend dein stärkstes eigenes Argument.

Offline-Praxis

Vertiefung und Recherche: [aufgaben/ML-K09-A01.md](#)

Lerncheck

Markiere: ✓ sicher · ~ teilweise · ? noch unklar

- ☐ Ich kann ein neuronales Netz vereinfacht erklären.
- ☐ Ich unterscheide Sprachproduktion und Wahrheitsprüfung.
- ☐ Ich kann einen KI-Einsatz mit Schutzmaßnahmen bewerten.

Teil B · BPE 7: KI-Systeme in betrieblichen Situationen [↪ Inhalt](#)

Jetzt wechselt die Perspektive: Aus einzelnen Verfahren wird ein nachvollziehbarer Arbeitsprozess. Daten werden verdichtet, Bibliotheken gezielt eingesetzt und Werkzeuge anhand betrieblicher Anforderungen ausgewählt. Der Lernagent bleibt das Transferprojekt.

BWL-Perspektive

Ein technisch mögliches System ist nicht automatisch wirtschaftlich oder organisatorisch sinnvoll. Nutzen, Kosten, Risiken, Integration und Verantwortlichkeit gehören gemeinsam in die Entscheidung.

10 · Große Datenmengen sichtbar machen [↪ Inhalt](#)

Bei vielen Zeilen hilft bloßes Durchscrollen nicht. Visualisierung reduziert Komplexität, darf sie aber nicht verfälschen. Gute Diagramme beantworten eine konkrete Frage und zeigen zugleich Unsicherheit oder Grenzen.

10.1 Vom Auftrag zur Grafik

Frage	Geeignete Darstellung	Beispiel Lernagent
Wie häufig kommen Werte vor?	Histogramm	Verteilung der Bearbeitungszeit
Wie unterscheiden sich Gruppen?	Boxplot / Balken	Ergebnisse nach Lernmodul
Hängen zwei Größen zusammen?	Streudiagramm	Zeit und Lösungsquote
Wie entwickelt sich etwas?	Liniendiagramm	Lernfortschritt über Wochen
Wie groß sind Anteile?	Balken; Kreis nur bei wenigen Teilen	Anteile der Empfehlungsklassen

10.2 Aggregieren, ohne Wichtiges zu verstecken

Viele Einzelwerte werden häufig gruppiert und zusammengefasst. Mittelwerte sind nützlich, können aber Ausreißer und ungleiche Verteilungen verdecken. Ergänze deshalb Anzahl, Median, Streuung oder ein Verteilungsdiagramm.

```
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv('daten/lernergebnisse.csv')
summary = df.groupby('modul').agg(
    anzahl=('punktzahl', 'size'),
    mittelwert=('punktzahl', 'mean'),
    median=('punktzahl', 'median')
)
print(summary)
```

Vier Prüfungen vor der Veröffentlichung

- Achsen: Beginnt die Skala sinnvoll und sind Einheiten sichtbar?
- Auswahl: Fehlen Gruppen oder Zeiträume?
- Verdichtung: Verdeckt der Durchschnitt eine problematische Streuung?
- Datenschutz: Lassen sich einzelne Personen indirekt erkennen?

Praxisregel

Zeige kleine Gruppen nicht gesondert, wenn dadurch Personen erkennbar werden könnten. Fasse zusammen oder unterdrücke die Darstellung.

Kurzauftrag

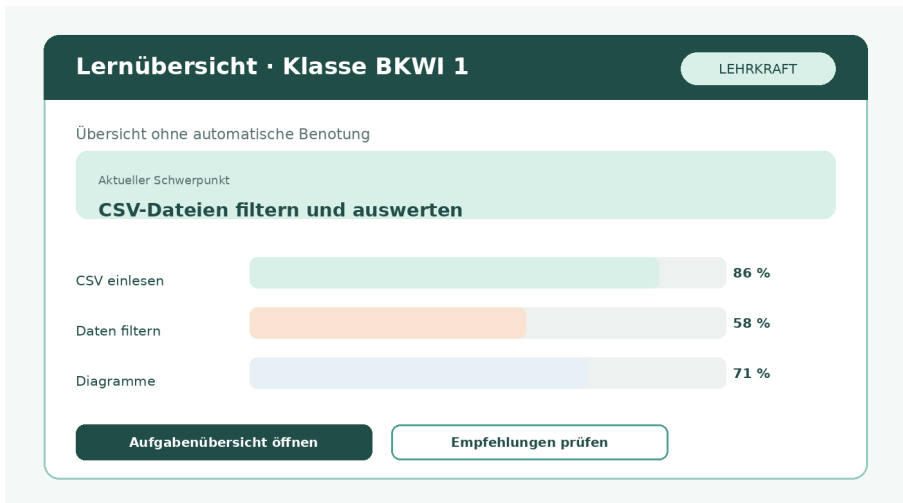
Wähle für jede Frage ein Diagramm und begründe deine Wahl: Verteilung der Punkte; Zusammenhang zwischen Versuchen und Punkten; durchschnittliche Punkte je Modul; Fortschritt einer Klasse über sechs Wochen.

Denkimpuls

Benenne zuerst Datentyp und Vergleichsrichtung: Verteilung, Zusammenhang, Gruppenvergleich oder Zeitverlauf.

10.3 View zuerst: Dashboard des Lernagenten

Bevor du Code schreibst, entwirfst du die Ergebnis-View. Dadurch wird sichtbar, welche Information die Software liefern muss und welche Daten dafür wirklich benötigt werden.



Mockup: Dashboard für die Lehrkraft. Die Übersicht zeigt Lernschwerpunkte, bleibt aber ausdrücklich eine Entscheidungshilfe und keine automatische Benotung.

Vom Mockup zur Anforderung

Element der View	Benötigte Daten	Logik
Lernstand	gelöste Aufgaben, Gewichtung	Score berechnen
Empfehlung	Score je Kompetenz, Schwellenwerte	nächsten Bedarf auswählen
Sicherheit	Datenmenge, Modellwahrscheinlichkeit	niedrig/mittel/hoch zuordnen
Begründung	stärkste Beobachtung	verständlichen Satz erzeugen

Transparenz

Die View zeigt nicht nur eine Empfehlung, sondern auch ihre Grundlage und Sicherheit. Das erleichtert Prüfung und Widerspruch.

ML-K10-A01 · 10.4 Praxisauftrag: Management- und Lernansicht

Material: `notebooks/08\_grosse\_daten\_visualisieren.ipynb` · Abgabe: `abgaben/ML-K10-A01` · Prüfnachweis: Ergebnis, Deutung, mindestens ein Test- oder Grenzfall.

Analysiere einen größeren Datensatz und erstelle zwei Ansichten: eine Detailansicht für Lernende und eine verdichtete Übersicht für Verantwortliche. Formuliere je Ansicht drei Fragen, zwei Diagramme und eine Handlungsempfehlung.

Denkimpuls

Welche Entscheidung soll die jeweilige Person treffen?

Strategiehilfe

Arbeite in dieser Reihenfolge: Frage → Kennzahl → Gruppierung → Diagramm → Aussage → Grenze.

Fach- oder Syntaxhilfe

Repository: `aufgaben/ML-K10-A01.md`; passende CSV-Dateien liegen unter `notebooks/daten/`.

Ansicht	Adressat	Kernfrage	Diagramm	Handlung
Detail				
Übersicht				

Diagramm-Review zu zweit

- Ist die Kernaussage in zehn Sekunden erkennbar?
- Sind Titel, Einheit, Zeitraum und Quelle vorhanden?
- Ist die gewählte Darstellung fachlich passend?
- Kann die Grafik zu einer falschen Schlussfolgerung verleiten?
- Welche Information fehlt für eine Entscheidung?

Lerncheck

Markiere: ✓ sicher · ~ teilweise · ? noch unklar

- ☐ Ich wähle Diagramme passend zur Frage.
- ☐ Ich prüfe Aggregationen und kleine Gruppen.
- ☐ Ich kann aus einer View Daten- und Logikanforderungen ableiten.

11 · Ein reproduzierbarer ML-Workflow mit Bibliotheken [↔ Inhalt](#)

Bibliotheken ersparen wiederkehrende Arbeit. Sie ersetzen aber nicht das fachliche Denken. Wer nur Methodenaufrufe kopiert, kann ein lauffähiges und trotzdem falsches Ergebnis erzeugen. Deshalb dokumentieren wir jeden Schritt.

Bibliothek	Rolle	Typischer Einsatz
pandas	Tabellendaten	einlesen, filtern, gruppieren
matplotlib / seaborn	Visualisierung	Verteilungen und Zusammenhänge
scikit-learn	Machine Learning	Aufteilung, Training, Kennzahlen, Pipeline
Jupyter	Arbeitsumgebung	Code, Erklärung und Ergebnis verbinden

11.1 Eine Pipeline mit überprüfbaren Stationen

Station	Artefakt	Prüffrage
Einlesen	DataFrame + Datensteckbrief	Spalten, Typen und Umfang korrekt?

Station	Artefakt	Prüffrage
Bereinigen	dokumentierte Regeln	Welche Werte wurden verändert?
Aufteilen	Train/Test-Split	Kein Datenleck?
Baseline	einfacher Vergleich	Schlägt das Modell die Baseline?
Trainieren	Modell + Parameter	Ist der Lauf reproduzierbar?
Evaluieren	Kennzahlen + Fehlerfälle	Passt die Kennzahl zum Zweck?
Kommunizieren	View + Modellkarte	Sind Grenzen verständlich?

11.2 Code lesen: von Daten zu einer Klassifikation

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.dummy import DummyClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import classification_report

df = pd.read_csv('daten/lernergebnisse.csv')
X = df[['loesungsquote', 'versuche', 'bearbeitungszeit']]
y = df['empfehlung']
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42, stratify=y
)

baseline = DummyClassifier(strategy='most_frequent')
modell = DecisionTreeClassifier(max_depth=3, random_state=42)
for kandidat in [baseline, modell]:
    kandidat.fit(X_train, y_train)
    print(classification_report(y_test, kandidat.predict(X_test)))
```

Erkläre statt kopiere

- Warum werden nur drei Merkmale in X aufgenommen?
- Warum gehört die Empfehlung in y?
- Wozu dienen random_state und stratify?
- Warum wird die Baseline mittrainiert?
- Welche Folge hätte ein sehr tiefer Entscheidungsbaum?

Syntaxhilfe

X bezeichnet üblicherweise die Eingabemerkmale, **y** die Zielgröße. Diese Namen sind Konventionen, keine besonderen Python-Schlüsselwörter.

Reproduzierbarkeit

Notiere Datenstand, Bibliotheksversionen, Zufallsstartwert, Merkmale, Zielgröße, Parameter und Kennzahlen. Eine andere Person soll deinen Lauf nachvollziehen können.

ML-K11-A01 · 11.3 Praxisauftrag: Bibliotheken gezielt einsetzen

Material: ``notebooks/07_modellpruefung.ipynb`` · Abgabe: ``abgaben/ML-K11-A01`` · Prüfnachweis: Ergebnis, Deutung, mindestens ein Test- oder Grenzfall.

Erstelle oder ergänze ein Notebook, das einen Datensatz einliest, untersucht, aufteilt, mit einer Baseline und einem zweiten Modell bearbeitet und bewertet. Jede Codezelle erhält davor eine kurze fachliche Absicht und danach eine Ergebnisdeutung.

Denkimpuls

Wenn du eine Zeile nicht erklären kannst, zerlege sie in kleinere Schritte.

Strategiehilfe

Lege zuerst sieben Markdown-Überschriften für die Pipeline an. Fülle dann schrittweise Code und Deutung ein.

Fach- oder Syntaxhilfe

Repository-Einstieg: `notebooks/02_daten_erkunden.ipynb`, `notebooks/03_lineare_regression.ipynb` und `notebooks/04_ki_tools_bwl.ipynb`.

Minimaler Notebook-Check

- ☐ Kernel neu starten und alle Zellen in Reihenfolge ausführen.
- ☐ Keine versteckten Variablen aus früheren Läufen verwenden.
- ☐ Relative Dateipfade nutzen.
- ☐ Fehlende Werte sichtbar prüfen.
- ☐ Baseline und Modell auf denselben Testdaten vergleichen.
- ☐ Diagramme beschriften und Aussagen formulieren.

11.4 Modellkarte

Feld	Eintrag
Zweck und Nutzergruppe	
Trainingsdaten und Zielgröße	
Merkmale	
Kennzahlen und Baseline	
bekannte Grenzen	
nicht vorgesehene Nutzung	
menschliche Kontrolle	

11.5 Fehlerwerkstatt

Analysiere nicht nur Fehlermeldungen des Programms, sondern auch fachliche Fehler.

Symptom	Mögliche Ursache	Prüfung
Datei nicht gefunden	Pfad oder Arbeitsordner falsch	aktuellen Ordner und relativen Pfad prüfen

Symptom	Mögliche Ursache	Prüfung
Text statt Zahl	Datentyp oder Dezimalzeichen	dtypes und Beispielwerte ansehen
sehr hohe Testgüte	Datenleck oder Duplikate	Merkmale und Splitting untersuchen
stark schwankende Güte	kleiner Datensatz / Zufall	mehrere Splits oder Cross-Validation
gute Kennzahl, schlechte Praxis	falsches Ziel oder falsche Metrik	Stakeholder und Fehlerfolgen prüfen

Debug-Auftrag

Wähle einen technischen und einen fachlichen Fehler. Dokumentiere Beobachtung, Hypothese, Test, Ergebnis und Korrektur.

Denkimpuls

Ändere pro Test nur eine Sache. So erkennst du, welche Änderung wirkt.

Lerncheck

Markiere: ✓ sicher · ~ teilweise · ? noch unklar

- ☐ Ich kann die Rollen zentraler Bibliotheken erklären.
- ☐ Ich dokumentiere einen reproduzierbaren Workflow.
- ☐ Ich vergleiche ein Modell mit einer Baseline.
- ☐ Ich trenne technische und fachliche Fehler.

12 · KI-Werkzeuge für BWL-Probleme auswählen ↵ Inhalt

Nicht das Werkzeug steht am Anfang, sondern der betriebliche Bedarf. „Wir wollen KI einsetzen“ ist noch keine Problemdefinition. Eine gute Auswahl macht Kriterien, Gewichtungen und Unsicherheit sichtbar.

12.1 Das Problem schärfen

Unklar	Prüfbar formuliert
Kundenservice verbessern	Antwortzeit für Standardanfragen senken, ohne Fehlerquote zu erhöhen
Lernen personalisieren	passende Folgeübung vorschlagen und Begründung anzeigen
Verkäufe steigern	Nachfrage pro Produktgruppe für vier Wochen prognostizieren
Dokumente automatisieren	Rechnungsdaten extrahieren und zur Prüfung markieren

Problem-Canvas

Nutzer · Entscheidung · gewünschtes Ergebnis · verfügbare Daten · Fehlerfolgen · Prozessintegration · Verantwortliche

Build, Buy oder No-AI?

- Build: selbst entwickeln; hohe Anpassbarkeit, aber Entwicklungs- und Betriebsaufwand.
- Buy: vorhandenes Produkt nutzen; schneller Start, aber Abhängigkeit und Integrationsfragen.
- No-AI: Regeln, Suche oder Prozessverbesserung; oft transparenter und günstiger.

12.2 Kriterien gewichten und bewerten

Eine Nutzwertanalyse zwingt dazu, Prioritäten offenzulegen. Die Punktzahl ersetzt keine Entscheidung, strukturiert aber die Diskussion.

Kriterium	Gewicht	Tool A (1-5)	Tool B (1-5)	Eigenbau (1-5)
fachliche Eignung	25			
Datenschutz	20			
Erklärbarkeit	15			

Kriterium	Gewicht	Tool A (1-5)	Tool B (1-5)	Eigenbau (1-5)
Integration	15			
Kosten über 3 Jahre	15			
Anbieterabhängigkeit	10			
Summe	100			

gewichteter_nutzen = sum(gewicht * punkte for gewicht, punkte in bewertungen)
 # Bei Gewichten in Prozent anschließend durch 100 teilen.

Sensitivitätsanalyse

Ändere ein wichtiges Gewicht, etwa Datenschutz von 20 auf 30 Prozent, und gleiche andere Gewichte aus. Bleibt die Rangfolge stabil? Wenn eine kleine Änderung die Entscheidung dreht, ist das Ergebnis empfindlich und sollte nicht als eindeutig dargestellt werden.

Achtung Scheingenauigkeit

Punktwerte beruhen auf Annahmen. Belege, Unsicherheiten und Ausschlusskriterien gehören neben die Rechnung.

ML-K12-A01 · 12.3 Fallstudie: Werkzeug für einen Lernagenten

Material: `notebooks/04\_ki\_tools\_bwl.ipynb` · Abgabe: `abgaben/ML-K12-A01` · Prüfnachweis: Ergebnis, Deutung, mindestens ein Test- oder Grenzfall.

Eine Schule möchte Übungen empfehlen und kurze Lernhinweise erzeugen. Vergleiche drei Lösungswege: regelbasiert, klassisches ML plus Textbausteine, ML plus externes Sprachmodell. Erstelle Problem-Canvas, Muss-Kriterien, Nutzwertanalyse und Sensitivitätsprüfung.

Denkimpuls

Welche Daten müssten die drei Varianten an wen übertragen?

Strategiehilfe

Definiere zuerst Ausschlusskriterien. Nur verbleibende Varianten erhalten Punkte.

Fach- oder Syntaxhilfe

Repository: [aufgaben/ML-K12-A01.md](#) sowie [notebooks/04_ki_tools_bwl.ipynb](#).

Variante	Stärke	Risiko	Offene Evidenz
Regeln			
ML + Textbausteine			
ML + Sprachmodell			

Entscheidungsvorlage auf einer Seite

- Entscheidungssatz: Welche Variante wird wofür empfohlen?
- Begründung: drei wichtigste Kriterien und Belege.
- Risiken: wichtigste Fehlerfolge und Schutzmaßnahmen.
- Wirtschaftlichkeit: Einmal- und laufende Aufwände.
- Pilot: kleinster sinnvoller Test mit Abbruchkriterien.

12.4 Wirtschaftlichkeit und Verantwortung zusammendenken

Dimension	Fragen
Nutzen	Welche Zeit, Qualität oder Wirkung verbessert sich messbar?
Kosten	Lizenz, Entwicklung, Integration, Schulung, Kontrolle, Exit?
Risiko	Welche Schäden sind möglich und wer trägt sie?

Dimension	Fragen
Organisation	Wer pflegt Daten, Modell, Regeln und Dokumentation?
Recht & Datenschutz	Welche Grundlage, Zweckbindung und Schutzmaßnahmen gelten?
Akzeptanz	Können Betroffene Ergebnis und Widerspruchsweg verstehen?

Pitch in 90 Sekunden

Stelle deine Entscheidung einer Schulleitung oder Geschäftsführung vor. Nenne Problem, Empfehlung, wichtigsten Beleg, Hauptrisiko und nächsten Pilotschritt.

Denkimpuls

Beginne mit dem Entscheidungssatz - nicht mit der Technik.

Lerncheck

Markiere: ✓ sicher · ~ teilweise · ? noch unklar

- ☐ Ich beginne bei Problem und Entscheidung.
- ☐ Ich kann Kriterien gewichten und eine Sensitivität prüfen.
- ☐ Ich berücksichtige laufende Kosten, Datenschutz und Exit.

13 · Abschlussprojekt: einen verantworteten Lernagenten entwickeln

↪ Inhalt

Im Abschlussprojekt verbindest du Datenarbeit, Modellprüfung, View-Entwurf und betriebliche Entscheidung. Ziel ist kein allwissendes System, sondern ein enger, überprüfbarer Prototyp.

ML-K13-A01 · Projektauftrag

Entwickle einen Lernagenten, der auf Basis geeigneter Übungsdaten genau eine klar abgegrenzte Entscheidung unterstützt, zum Beispiel die Auswahl der nächsten Übung. Zeige Ergebnis, Begründung und Unsicherheit in einer verständlichen View.

Material: `notebooks/09\_abschlussprojekt\_start.ipynb`, `aufgaben/ML-K13-A01.md` und `notebooks/daten/testfaelle\_lernagent.csv` · Abgabeordner: `abgaben/ML-K13-A01`.

13.1 Projektgrenzen

Im Umfang	Außerhalb des Umfangs
CSV-Datensatz oder synthetische Daten	reale personenbezogene Leistungsdaten ohne Freigabe
eine Zielgröße	vollständige Schulplattform
Baseline und ein Modell	beliebig viele unbewertete Modelle
Mockup oder einfacher Prototyp	produktiver Dauerbetrieb
dokumentierte menschliche Kontrolle	automatische Benotung oder Sanktion

Erfolgskriterien

- fachlich sinnvolle Problemdefinition,
- nachvollziehbare Daten- und Modellentscheidung,
- ehrliche Evaluation gegen eine Baseline,
- verständliche Ergebnis-View,
- konkrete Grenzen und Schutzmaßnahmen.

13.2 Von der View zur Architektur



Mockup: Lernendenansicht mit Empfehlung, Begründungsdaten und einer Möglichkeit zur Prüfung.

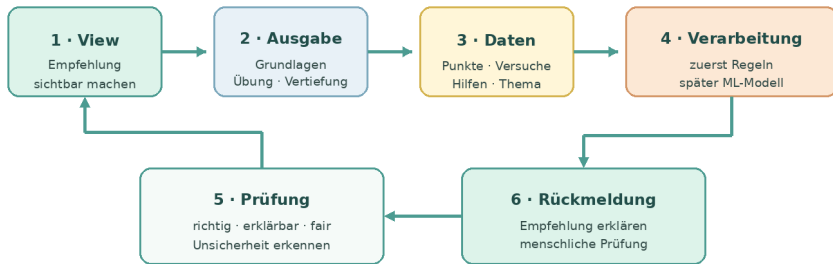
Markiere im Mockup jedes sichtbare Ergebnis. Leite daraus die nötigen Eingaben und Berechnungen ab. Erst danach entscheidest du, ob eine Regel, ein ML-Modell oder eine Kombination sinnvoll ist.

Baustein	Verantwortung	Beispiel
Datenquelle	zulässige Rohdaten liefern	synthetische Übungsergebnisse
Vorbereitung	prüfen und Merkmale bilden	Lösungsquote je Kompetenz
Entscheidungslogik	Empfehlung berechnen	Regel oder Klassifikator
Erklärung	Grundlage verständlich machen	„weil 2 von 3 Filteraufgaben ...“
View	Ergebnis und Unsicherheit zeigen	Dashboard-Mockup
Kontrolle	Freigabe und Widerspruch	Lehrkraft / Lernende

Architekturprinzip

Jeder Baustein erhält eine klare Aufgabe. Dadurch können Daten, Modell, Erklärung und Oberfläche getrennt geprüft und später ausgetauscht werden.

Was muss hinter der View entwickelt werden?



Die View ist Ausgangspunkt und Prüfstein der gesamten Entwicklung.

System-Mockup: Die sichtbare View wird auf Aufgaben, Daten, Verarbeitung, Rückmeldung und Prüfung zurückgeführt.

13.3 Projektphasen und Abnahmen

Phase	Liefergegenstand	Abnahmefrage
1 · Auftrag	Problem-Canvas + Nicht-Ziele	Ist die Entscheidung eng genug?
2 · View	beschriftetes Mockup	Versteht die Zielgruppe das Ergebnis?
3 · Daten	Datensteckbrief + Prüfbericht	Sind Daten geeignet und zulässig?
4 · Baseline	Regel oder Dummy-Modell	Gibt es einen ehrlichen Vergleich?
5 · Modell	Notebook / Skript	Ist der Ablauf reproduzierbar?
6 · Evaluation	Kennzahlen + Fehlerfälle	Sind Folgen und Teilgruppen geprüft?
7 · Übergabe	Modellkarte + Präsentation	Sind Grenzen und Betrieb geklärt?

Rollen im Team

- Product Owner: schärft Bedarf und Erfolgskriterien.
- Data Steward: prüft Datensatz, Qualität und Datenschutz.
- ML Developer: implementiert Baseline, Modell und Evaluation.
- UX/QA: prüft View, Verständlichkeit, Fehlerfälle und Dokumentation.

In kleinen Teams können Rollen kombiniert werden. Entscheidungen bleiben trotzdem sichtbar zugeordnet.

13.4 Sprint 1: Auftrag, Daten und View

Erstellt Problem-Canvas, Datensteckbrief und ein beschriftetes Mockup. Führt anschließend einen 5-Minuten-Nutzertest durch: Eine Person erklärt laut, was sie sieht und welche Handlung sie daraus ableitet.

Denkimpuls

Wenn zwei Personen die View unterschiedlich verstehen, fehlt eine Beschriftung oder Begründung.

Strategiehilfe

Notiert Beobachtungen, nicht nur Meinungen: „Person suchte 20 Sekunden nach ...“.

Testfrage	Beobachtung	Änderung
Welche Empfehlung wird gegeben?		
Warum wird sie gegeben?		
Wie sicher ist das Ergebnis?		
Wie kann widersprochen werden?		

13.5 Sprint 2: Baseline, Modell und Prüfung

Implementiert eine Baseline und ein passendes Modell. Trennt Training und Test, begründet Kennzahlen und untersucht mindestens fünf Fehlfälle. Vergleicht nicht nur Zahlen: Welche Lösung ist für den schulischen Einsatz besser erklärbar und kontrollierbar?

Denkimpuls

Der beste Kennzahlenwert gewinnt nicht automatisch.

Strategiehilfe

Erstellt eine Ergebnistabelle: Baseline - Modell - Differenz - fachliche Bedeutung.

13.6 Testfälle des Lernagenten

Testfall	Erwartung	Warum wichtig?
viele korrekte Lösungen, wenig Zeit	Vertiefung möglich	klarer Standardfall
wenige Lösungen, viele Versuche	Wiederholung / Hilfe	Förderfall nicht übersehen
widersprüchliche Signale	Unsicherheit anzeigen	keine Scheinsicherheit
fehlende Werte	keine ungeprüfte Empfehlung	Robustheit
unbekannte Kategorie	kontrollierter Hinweis	Eingabe außerhalb Training
Grenzwert knapp erreicht	stabile, erklärbare Reaktion	Schwellenwert prüfen

Abbruch- und Übergaberegeln

- Keine Empfehlung, wenn erforderliche Daten fehlen.
- Menschliche Prüfung bei geringer Sicherheit oder hohem Risiko.
- Protokollierung der Modellversion und Eingaben ohne unnötige Personendaten.
- Regelmäßige Kontrolle auf veränderte Daten und sinkende Güte.
- Klarer Weg zur Korrektur oder Deaktivierung.

Definition of Done

Der Prototyp ist fertig, wenn eine andere Person ihn ausführen, das Ergebnis verstehen, Grenzen erkennen und die wichtigsten Tests nachvollziehen kann.

13.7 Präsentation und Projektdokumentation

Demo-Ablauf: sieben Minuten

Zeit	Inhalt
0:00–0:45	Problem, Zielgruppe und Entscheidung
0:45–1:45	View und Nutzungsszenario
1:45–3:00	Daten und Merkmale
3:00–4:30	Baseline, Modell und Kennzahlen
4:30–5:30	Fehlerfall und Grenze
5:30–6:30	Schutzmaßnahmen und menschliche Kontrolle
6:30–7:00	Fazit und nächster Schritt

Bewertungsraster

Bereich	Anteil	Stark erfüllt, wenn ...
Problem & Nutzen	15 %	Auftrag eng, relevant und messbar ist
Datenarbeit	20 %	Qualität, Herkunft und Grenzen dokumentiert sind
Modell & Prüfung	25 %	Baseline, Kennzahlen und Fehleranalyse überzeugen
View & Erklärung	15 %	Ergebnis, Grund und Unsicherheit verständlich sind
Verantwortung	15 %	Risiken, Datenschutz und Kontrolle konkret sind
Dokumentation & Team	10 %	Projekt reproduzierbar und Entscheidungen sichtbar sind

Offline-Praxis

Projektgrundlage: aufgaben/ML-K13-A01.md. Nutzt vorhandene Notebooks und Datensätze. Alle Übungsdateien, Testfälle und Starterfassungen liegen im Offlinepaket vor.

13.8 Persönliche Reflexion

Beantworte die Fragen einzeln. Die Reflexion bewertet nicht nur das Ergebnis, sondern dein fachliches Lernen.

Welche Entscheidung im Projekt war fachlich am schwierigsten?

Welche Annahme hat sich als falsch oder zu einfach erwiesen?

Was zeigt eure Kennzahl - und was zeigt sie nicht?

Welcher Fehlerfall hat eure Lösung verändert?

Wann sollte euer Lernagent ausdrücklich keine Empfehlung geben?

Welche Kompetenz möchtest du im nächsten Projekt vertiefen?

Lerncheck

Markiere: ✓ sicher · ~ teilweise · ? noch unklar

- ☐ Ich kann ein enges ML-Problem in ein Projekt übersetzen.
- ☐ Ich leite Daten und Logik aus einer View ab.
- ☐ Ich bewerte Modell, Fehlerfolgen und Verantwortung gemeinsam.

☐ Ich kann den Prototyp verständlich demonstrieren.

Abschluss: Begriffe sicher verwenden ↵

Inhalt

Begriff	Kurzdefinition
Baseline	einfache Vergleichslösung
Trainingsdaten	Beispiele, aus denen das Modell lernt
Testdaten	zur unabhängigen Prüfung zurückgehaltene Fälle
Überanpassung	Modell passt zu stark zu Trainingsbesonderheiten
Verwechslungsmatrix	Übersicht richtiger und falscher Klassenentscheidungen
Aggregation	Zusammenfassung vieler Einzelwerte
Pipeline	definierte Folge von Verarbeitungsschritten
Reproduzierbarkeit	ein Ergebnis lässt sich unter dokumentierten Bedingungen wiederholen
Nutzwertanalyse	gewichteter Vergleich mehrerer Alternativen
Sensitivitätsanalyse	Prüfung, wie Annahmeänderungen das Ergebnis beeinflussen
Modellkarte	kompakte Dokumentation von Zweck, Leistung und Grenzen
Human in the Loop	Mensch bleibt an festgelegten Stellen beteiligt

Ein Satz, der bleibt

Vom Ergebnis zur Verantwortung

Ein gutes ML-System liefert nicht nur eine Vorhersage. Es macht sichtbar, wofür sie gedacht ist, worauf sie beruht, wie sicher sie ist, wo ihre Grenzen liegen und wer verantwortlich entscheidet.