

01 Einleitung

Kursübersicht > [KI-Technologien verstehen](#)

Einleitung: Warum Daten und Informationsverarbeitung die Grundlage für KI-Verständnis sind

Um KI-Systeme sinnvoll einsetzen und bewerten zu können, müssen wir verstehen, wie sie Informationen aufnehmen, verarbeiten und daraus Entscheidungen oder Antworten generieren. In diesem Modul treten wir daher einen Schritt zurück - bevor wir über konkrete Anwendungen oder Ergebnisse sprechen - und betrachten die Grundlagen: **Wie gelangen Systeme überhaupt an Informationen, und was passiert, wenn sie diese „verstehen“ sollen?**

Wenn wir über Künstliche Intelligenz sprechen, sprechen wir im Kern über **Informationsverarbeitung**. Ein KI-System kann nur so gute Ergebnisse liefern, wie die Informationen, auf die es Zugriff hat, sowie deren Strukturierung, Interpretation und Verknüpfung es zulassen.

Leitgedanke: KI verstehen heißt, Informationsverarbeitung verstehen.

Zielsetzung des Moduls

Im Rahmen dieses Moduls erhalten Sie daher nicht nur einen Überblick über zentrale technische Grundlagen, wie etwa die Aufbereitung und Strukturierung von Daten, verschiedene Lernarten oder die Generierung von Ergebnissen (Output), sondern wir verknüpfen diese Aspekte auch mit aktuellen Forschungserkenntnissen zur Informationsverarbeitung bei Systemen und Menschen als Kooperationspartnern.

Wie in den vorherigen Modulen bereits gezeigt wurde, reicht eine rein technische Perspektive auf KI nicht aus. Systeme agieren nicht im luftleeren Raum, sondern werden von Menschen entwickelt, trainiert und genutzt. Ihre Wirksamkeit hängt also immer auch davon ab, wie gut Menschen und Systeme miteinander interagieren, Informationen austauschen und interpretieren.

Fokus: Gemeinsame Informationsverarbeitung von Mensch und System

In diesem Modul bieten wir daher einen **systematischen Einstieg in das Thema Informationsverarbeitung in KI-Systemen** und in die Rolle, die der Mensch in dieser Kooperation spielt. Ausgangspunkt ist das **Modell der integrierten Informationsverarbeitung (Integrated Information Processing)**, das Technik und menschliches Denken gemeinsam betrachtet.



Dabei beschäftigen wir uns unter anderem mit folgenden Fragen:

- Wie müssen Informationen aufbereitet sein, damit Systeme sie „verstehen“ und verarbeiten können?
- Wie lernen Systeme, welche Informationen relevant sind, und wie treffen sie auf dieser Basis Entscheidungen?
- Und schließlich: Wie generieren KI-Systeme Ergebnisse, die für Menschen nachvollziehbar, verständlich und nutzbar sind?

Dieses Verständnis bildet die Grundlage für alle weiteren Themen in der Modulreihe - von Datenqualität und Bias bis hin zu transparenter und vertrauenswürdiger KI. Denn wer versteht, wie Systeme Informationen verarbeiten, kann besser beurteilen, wann ihre Ergebnisse hilfreich, fehlerhaft oder verzerrt sind - und wie Mensch und KI gemeinsam zu guten Entscheidungen kommen.

Kapitelübersicht

1

Input - Technik

Wie funktioniert der Input in ein KI-System auf technischer Ebene und welche Rolle spielen die verfügbaren Daten dabei?

2

Input - Gestaltung

Wie werden Informationen als Input genutzt? Wie beeinflusst dies die Gestaltung von KI-Systemen?

3

Verarbeitung - Technik

Die Frage wie KI-Systeme Daten verarbeiten wird in diesem Kapitel betrachtet.

4

Verarbeitung - Gestaltung

Wie kann die Verarbeitung von KI-Systemen gestaltet werden, damit einzuordnen ist, ob es so arbeitet wie gewünscht?

5

Output - Technik

Mehr als nur ein Ergebnis: KI-Outputs kritisch verstehen und richtig deuten.

6

Output - Gestaltung

Ein guter KI-Output beantwortet mehr als nur die Frage nach dem Ergebnis. Er erklärt das 'Warum', das 'Wie sicher' und das 'Was wäre wenn' für Nutzervertrauen.

LLMs

Dieses Kapitel erklärt, wie LLMs durch Wortvorhersage plausible Texte erzeugen, warum sie aber nichts wirklich verstehen, deshalb 'halluzinieren' und welche Kriterien bei der Auswahl des richtigen Modells entscheidend sind.

02

Input - Technik

Kursübersicht > [KI-Technologien verstehen](#)

1. Einleitung: Warum es wichtig ist, Daten zu verstehen

Wer mit KI-Systemen arbeitet, arbeitet immer auch mit Daten. Ob ein Chatbot Anfragen von Bürger:innen beantwortet, eine KI eingereichte Anträge prüft oder ein Analyse-Tool soziale Trends erkennen soll - der Ausgangspunkt all dieser Systeme sind Daten. Doch was genau sind eigentlich „Daten“? Und warum ist es so entscheidend, ihre Struktur, Herkunft und Qualität zu verstehen, bevor sie in ein KI-System eingespeist werden?

In gemeinwohlorientierten Organisationen wird KI oft eingesetzt, um Prozesse zu entlasten, Zugänge zu erleichtern oder faire Entscheidungen zu unterstützen. Aber ohne Verständnis dafür, was in ein System hineingeht, bleibt unklar, wie man seine Ergebnisse bewerten oder verbessern kann. **Input verstehen heißt, Daten verstehen.**

2. Was sind Daten und was nicht?

Daten sind zunächst nur Zeichen, Zahlen oder Symbole, die etwas in der Welt abbilden. Erst durch Interpretation, also durch das Einordnen dieser Zeichen in einen Kontext, entsteht Bedeutung.

Vom Zeichen zum Wissen

Zeichen: Einzelne Symbole oder Werte (z.B. „25“, „grün“, „Ja“).



Daten: Zeichen, die systematisch erfasst wurden (z.B. „25 Jahre alt“, „grüne Ampel“, „Ja/Nein Antwort“).



Information: Bedeutung, die sich aus Daten in einem bestimmten Kontext ergibt (z.B. „Die Person ist 25 Jahre alt“).



Wissen: Das Verständnis, wie diese Information einzuordnen ist („Menschen zwischen 18 und 30 gelten hier als junge Erwachsene“).

In der Praxis bedeutet das: Wenn eine KI Informationen zu eingereichten Förderanträgen verarbeitet, arbeitet sie nicht mit Wissen, sondern mit Daten - etwa Texten, Zahlenfeldern oder Kategorien. Das Wissen, wie

diese zu interpretieren sind, liegt beim Menschen oder in den Regeln, mit denen das KI-System trainiert wurde.

3. Wie müssen Daten gestaltet sein, um von KI-Systemen genutzt zu werden?

KI-Systeme können nur so gut arbeiten, wie die Daten es zulassen. Damit ein System Daten verarbeiten kann, müssen diese in einer bestimmten Struktur und einem Format vorliegen.

Datenstrukturen und -formate

Strukturierte Daten: Klar definierte Spalten, Werte und Datentypen (z.B. Tabellen, Datenbanken).

Unstrukturierte Daten: Texte, Bilder, Audiodateien, PDFs - also Informationen ohne feste Ordnung.

Halbstrukturierte Daten: Mischformen wie JSON oder XML, die Strukturmerkmale enthalten, aber flexibel bleiben. Beides sind textbasierte Formate, die Informationen in festen Strukturen speichern: JSON vor allem als Schlüssel-Wert-Paare, XML in verschachtelten Tags.

Viele gemeinwohlorientierte Organisationen arbeiten in ihrem Alltag überwiegend mit unstrukturierten Daten - etwa Antragsdokumenten, Berichten oder selbsterstellten Dokumenten mit Freitextantworten.

Diese unstrukturierten Daten können wertvoll sein, müssen aber in strukturierte oder maschinenlesbare Form gebracht werden, bevor sie als Input für eingesetzte KI-Systeme dienen können.

4. Möglichkeiten der Kategorisierung von Daten

Um zu verstehen, welche Art von Daten man einem KI-System zur Verfügung stellt, ist es hilfreich, Daten nach bestimmten Kriterien zu **kategorisieren**. Diese Kategorisierungen helfen dabei einzuschätzen, ob Daten geeignet sind, welche Form der Aufbereitung sie benötigen und welche Schlussfolgerungen sich später aus ihnen ziehen lassen.

Syntax

Ein erster Aspekt betrifft die **Syntax** - also die formale Struktur der Daten. Syntax beschreibt, in welcher Form ein Wert vorliegt: als Zahl, Text, Kategorie oder Wahr/Falsch-Angabe. Diese Unterscheidung ist entscheidend, weil viele KI-Modelle bestimmte Formate erwarten. Ein Text wie „Ja“ oder „Nein“ muss etwa in 0/1-Werte umgewandelt werden, wenn das System nur mit numerischen Eingaben arbeiten kann.

Erscheinung oder Form

Darüber hinaus spielt die **Erscheinung oder Form** der Daten eine Rolle. Damit ist gemeint, wie die Daten erfasst oder dargestellt sind - etwa als Antwortfeld in einem Formular, als Sensormessung oder als Fließtext in einem Bericht. Diese Form bestimmt häufig, wie leicht oder schwer Daten automatisiert weiterverarbeitet werden können. Während eine standardisierte Eingabemaske klare Werte liefert, sind handgeschriebene Dokumente oder unstrukturierte E-Mails für eine KI nur schwer zu deuten.

Zeitlicher Bezug

Ein weiterer wichtiger Aspekt ist der **zeitliche Bezug** der Daten. Daten sind Momentaufnahmen einer bestimmten Realität, die sich mit der Zeit verändern kann. Angaben zu Einkommensverhältnissen, Bevölkerungsdaten oder Nutzungszahlen können nach einigen Monaten oder Jahren bereits veraltet sein. Wenn eine KI also auf Basis alter Daten trainiert wurde, spiegelt sie möglicherweise eine Realität wider, die so gar nicht mehr existiert.

Skalenniveau

Daten lassen sich außerdem nach ihrem **Skalenniveau** unterscheiden - also danach, wie genau sie messbar sind.

- Nominale Daten (z.B. „Farbe der Karte“) lassen sich nicht in einer Rangfolge bringen.
- Ordinale Daten (z.B. „Zufriedenheit: niedrig - mittel - hoch“) hingegen schon.
- Intervall- und Rationskalen (z.B. Temperatur in °C oder Einkommen in Euro) ermöglichen präzise mathematische Berechnungen.

Das richtige Verständnis dieser Unterschiede ist essenziell, weil sie bestimmen, welche statistischen Verfahren und KI-Modelle überhaupt sinnvoll angewendet werden können.

Datentyp

Schließlich ist auch der **Datentyp** selbst von Bedeutung. Ein Datentyp legt fest, ob eine Information als Zahl, Text, boolescher Wert (wahr/falsch) oder komplexere Struktur vorliegt. Ein scheinbar einfacher Unterschied - etwa zwischen einer Zahl, die als Text gespeichert wurde („15“), und einer echten numerischen Variable - kann bei der Verarbeitung durch eine KI große Auswirkungen haben.

Praxisbezug

In der Praxis ist es hilfreich, diese verschiedenen Kategorien im Blick zu behalten. Wer etwa in einer Organisation arbeitet, die mithilfe von KI eingereichte Förderanträge analysiert, sollte sich fragen:

- Sind die Eingabefelder in den Formularen konsistent aufgebaut (Syntax)?
- Liegen die Anträge in digitaler oder gescannter Form vor (Erscheinung)?
- Beziehen sich die Daten auf aktuelle oder ältere Förderperioden (zeitlicher Bezug)?
- Sind die Bewertungskategorien der Gutachter*innen ordinal oder numerisch (Skalenniveau)?
- Und schließlich: Sind die einzelnen Werte korrekt als Text oder Zahl gespeichert (Datentyp)?

Erst wenn diese Grundlagen verstanden und überprüft sind, kann ein KI-System sinnvoll mit den Daten arbeiten und die Organisation sicherstellen, dass die Ergebnisse nachvollziehbar und belastbar bleiben.

5. Datenqualität

Viele Probleme in KI-Projekten entstehen nicht durch den Algorithmus, sondern durch **mangelhafte Datenqualität**. Gründe können z.B. menschliche **Eingabefehler**, fehlende oder uneinheitliche **Standardisierung** (z. B. unterschiedliche Formate wie Datum 01.02.24 vs. 2024-02-01) oder **Systemumbrüche** (wenn verschiedene Systeme Daten unterschiedlich speichern, z. B. fehlende Vorwahlen beim Wechsel des Systems) sein.

Vier zentrale Qualitätsmerkmale

1

Vollständigkeit: Sind alle notwendigen Informationen vorhanden?

2

Genauigkeit: Sind die Daten korrekt und überprüfbar?

3

Konsistenz: Stimmen Daten innerhalb eines Systems überein (z.B. gleiche Schreibweisen, gleiche Einheiten)?

4

Aktualität: Sind die Daten noch relevant oder bereits veraltet?

Ein Beispiel dazu

Eine Organisation möchte mithilfe von KI prüfen, ob Förderanträge vollständig ausgefüllt sind. Wenn jedoch alte Formulare im Umlauf sind oder Einträge unterschiedlich benannt wurden („Straße“ vs. „Str.“), kann die KI falsche Lücken oder Dubletten erkennen und so zusätzliche Arbeit für Mitarbeitende erzeugen, die diese falschen Positive dann händisch filtern müssen.

6. Beziehungen zwischen Daten: Abhängigkeiten, Korrelationen und Kausalität

KI-Systeme analysieren Daten nicht isoliert, sondern immer in ihren **Beziehungen zueinander**. Diese Beziehungen zu verstehen ist zentral, um beurteilen zu können, **was eine KI tatsächlich erkennt - und was sie nur zu erkennen scheint**.

Zunächst lohnt sich ein Blick auf den Unterschied zwischen **abhängigen** und **unabhängigen Variablen**. Eine unabhängige Variable ist ein Faktor, der andere Werte beeinflussen kann, während eine abhängige Variable das Ergebnis oder die Reaktion auf diesen Einfluss darstellt.

Ein einfaches Beispiel: Wenn eine Organisation untersucht, ob das Einkommen einer Person (unabhängige Variable) beeinflusst, ob sie finanzielle Unterstützung beantragt (abhängige Variable), dann kann eine KI diese Beziehung nur dann korrekt erkennen, wenn beide Variablen klar definiert und sauber erfasst sind.

Solche Zusammenhänge bilden die Grundlage vieler KI-Modelle. Doch wichtig ist: **Eine statistische Beziehung bedeutet nicht automatisch, dass ein echter ursächlicher Zusammenhang besteht.** KI-Systeme identifizieren häufig **Korrelationen**, also gleichzeitige Muster oder Bewegungen in den Daten, ohne zu verstehen, **warum** sie auftreten. Kausalität hingegen beschreibt, dass eine Veränderung in einer Variablen tatsächlich eine Veränderung in einer anderen verursacht.

Ein klassisches Beispiel verdeutlicht das: Wenn eine KI in Daten erkennt, dass in Monaten mit höherem Eisverkauf auch mehr Badeunfälle gemeldet werden, besteht zwar eine Korrelation, aber keine Kausalität. Der eigentliche Grund liegt in einer dritten Variable - dem warmen Wetter, das sowohl den Eisverkauf als auch die Zahl der Badeunfälle beeinflusst.

In gemeinwohlorientierten Projekten kann ein ähnliches Risiko auftreten. Eine KI, die Bürger:innenanfragen auswertet, könnte feststellen, dass bestimmte Stadtteile häufiger Beschwerden einreichen. Ohne Kontext könnte dies fälschlicherweise als „höhere Unzufriedenheit“ interpretiert werden - dabei könnten schlicht **unterschiedliche Kommunikationswege** oder **bessere digitale Zugänge** die Ursache sein.

Wer mit KI arbeitet, sollte daher immer fragen:

- Welche Variablen hängen logisch miteinander zusammen und welche nur zufällig?
- Welche Faktoren könnten im Hintergrund wirken, ohne in den Daten sichtbar zu sein?
- Und wie sicher kann ich sein, dass ein Muster tatsächlich eine Ursache-Wirkung-Beziehung darstellt?

Ein KI-System kann Muster sichtbar machen - aber die Interpretation dieser Muster bleibt menschliche Aufgabe.

7. Bias

Ein weiteres zentrales Thema im Umgang mit Daten für KI-Systeme ist **Bias**, also eine **Verzerrung oder Schieflage in den Daten**. Biases sind nicht immer auf den ersten Blick erkennbar, können aber große Auswirkungen auf die wahrgenommene Fairness, Zuverlässigkeit und Akzeptanz eines KI-Systems haben.

Im Kern entsteht ein Bias dann, wenn die Daten, mit denen ein System trainiert oder gefüttert wird, **nicht die tatsächliche Vielfalt oder Verteilung der Realität widerspiegeln**. Die KI „lernt“ dann ein einseitiges Bild - und reproduziert es bei jeder Entscheidung oder Empfehlung.

Man unterscheidet dabei zwei grundsätzliche Arten, wie ein Bias entstehen kann:

Falsche Abbildung der Realität

Hier sind die Daten schlicht **fehlerhaft, unvollständig oder falsch erhoben**. Vielleicht wurden bestimmte Gruppen gar nicht befragt, Datensätze ungleichmäßig aktualisiert oder Eingabefehler nie korrigiert. Ein Chatbot, der Anfragen von Bürger:innen beantwortet, könnte etwa eine Schieflage aufweisen, wenn die zugrunde liegenden Textbeispiele überwiegend aus einer bestimmten Altersgruppe stammen.

Abbildung einer ungleichen Realität

In diesem Fall spiegeln die Daten die reale Welt korrekt wider - doch diese Welt ist selbst **ungleich oder diskriminierend**. Wenn eine KI beispielsweise historische Personaldaten analysiert, in denen Männer häufiger Führungspositionen innehatten, dann „lernt“ sie diese Ungleichheit mit, selbst wenn niemand sie absichtlich eingebaut hat. Sie läuft so Gefahr, diese Ungleichheit zu reproduzieren.

Bias ist deshalb nicht nur ein technisches, sondern vor allem ein **gesellschaftliches Problem**, das sich in die Technologie einschreibt. In gemeinwohlorientierten Projekten ist der Umgang damit besonders wichtig, weil Entscheidungen hier direkt über **Zugang zu Unterstützung, Sichtbarkeit oder Teilhabe** entscheiden können.

Leitfragen zum Erkennen von Bias

- Wer oder was ist in den Daten **überrepräsentiert**?
- Wer oder was **kommt kaum oder gar nicht vor**?
- Welche historischen oder strukturellen Ungleichheiten könnten sich in den genutzten Daten widerspiegeln?
- Welche Werte oder Annahmen liegen in der Datenerhebung selbst verborgen (z.B. Sprache, Begrifflichkeiten, Klassifikationen)?

Bias lässt sich nie vollständig vermeiden - aber er lässt sich erkennen, benennen und abmildern.

Dazu gehört, die Herkunft und Zusammensetzung der Daten kritisch zu prüfen, verschiedene Perspektiven in die Entwicklung einzubeziehen und den Kontext der Datennutzung offenzulegen. Gerade für Organisationen, die im Dienst des Gemeinwohls arbeiten, ist dies ein entscheidender Schritt, um sicherzustellen, dass KI-Systeme nicht unbeabsichtigt bestehende Ungleichheiten fortschreiben, sondern dazu beitragen, **fairere und inklusivere Entscheidungsprozesse** zu fördern.

03

Input - Gestaltung

[Kursübersicht](#) > [KI-Technologien verstehen](#)

1. Einleitung: Methoden integrierter Informationsverarbeitung

Damit KI-Systeme Menschen sinnvoll unterstützen können, müssen beide Seiten dieselbe „Sprache“ sprechen und sich darüber austauschen, was notwendig ist, um eine Aufgabe zu lösen. Während Maschinen Informationen als strukturierte Daten, Gewichte und Wahrscheinlichkeiten verarbeiten, deuten Menschen dieselben Informationen in Bedeutungen, Erfahrungen und Zielen.

Die Kunst besteht darin, diese beiden Arten der Informationsverarbeitung miteinander zu verbinden. Genau hier setzt der Gedanke der integrierten Informationsverarbeitung an: Informationen fließen nicht nur *vom Menschen ins System*, sondern auch *vom System zum Menschen zurück*. Nutzende verstehen dadurch, wie ein System arbeitet, können ihre Eingaben korrigieren, und lernen mit der Zeit, wie sie bessere, passgenauere Informationen bereitstellen.

In diesem Kapitel stellen wir drei Methoden vor, die diese Zusammenarbeit zwischen Mensch und Maschine besonders unterstützen:

1

Information Disclosure - das System teilt kontextrelevante Informationen über seine Entscheidungen mit den Nutzer:innen so, dass diese ihren Input anpassen können.

2

Informationen editieren - Nutzer:innen können Eingaben in das System verändern und so beobachten, wie sich diese Veränderungen auf das System auswirken.

3

Zeitverlauf - das System zeigt, wie sich Informationen und Bewertungen über die Zeit hinweg verändern.

Alle drei Methoden sollen die Informationsverarbeitung so gestalten, dass sie verständlich, nachvollziehbar und auf gegenseitiges Lernen ausgelegt ist, sodass Mensch und System bestmöglichen Input für die weitere Verarbeitung generieren.

2. Information Disclosure

Information Disclosure bedeutet, dass ein KI-System mehr Informationen bereitstellt, als nur das bloße Ergebnis.

Statt lediglich „Wohnung geeignet: Ja“ oder „Score: 0,73“ auszugeben, gibt das System auch Einblick in die Gründe und Sicherheiten seiner Einschätzung und die Faktoren, die zu dieser Entscheidung führen könnten. Dadurch können Nutzende besser nachvollziehen, wie Entscheidungen zustande kommen und ob sie auf soliden Daten beruhen oder Unsicherheiten bestehen.

Das Beispiel „Wohnbrücke e. V.“

Die Organisation *Wohnbrücke e. V.* unterstützt Menschen in Not bei der Wohnungssuche in einer Großstadt.

Um Wohnungsangebote systematisch zu bewerten, nutzt sie ein KI-System, das jeder Immobilie einen Eignungswert zwischen 0 und 1 zuweist.

Ein Angebot in der Lindenstraße 12 erhält etwa den Wert 0,82.

Das System zeigt außerdem an:

- **Hauptfaktoren:** Barrierefreiheit (+0,15), Nähe zu Betreuungseinrichtungen (+0,12), moderate Miete (+0,08).
- **Unsicherheiten:** Kein aktuelles Bildmaterial vorhanden, fehlende Angaben zur Heizungsart.
- **Confidence Score:** 0,76 (relativ hohe Sicherheit).

So erkennen die Mitarbeitenden: Das System bewertet das Objekt positiv, aber mit gewissen Unsicherheiten, die auf fehlende Daten zurückzuführen sind.

Durch solche Offenlegungen wird die Entscheidungslogik greifbarer. Mitarbeitende lernen, welche Merkmale besonders wichtig sind und wann sie die Bewertung besser hinterfragen sollten. So können sie ihren Input in das System bestmöglich anpassen.

Gleichzeitig entsteht eine Transparenz, die Vertrauen schafft, sowohl in das System selbst als auch in die Entscheidungen, die darauf basieren.

Reflexionsfrage

Wann wäre es in Ihrem Arbeitskontext hilfreich, zu sehen, wie sicher sich ein System bei seiner Einschätzung ist?

Vorteile von Information Disclosure

- Erhöht Transparenz und Nachvollziehbarkeit.
- Fördert Vertrauen in KI-gestützte Prozesse.
- Hilft, Unsicherheiten zu erkennen und gezielt zu beheben.
- Unterstützt Lernprozesse bei Nutzenden („Wie denkt das System?“).

Grenzen von Information Disclosure

- Gefahr der Überforderung: Zu viele Zahlen oder Indikatoren können verwirren.
- Missverständnisse möglich: Ein hoher Confidence Score heißt nicht automatisch, dass die Aussage des Systems „richtig“ ist, sondern nur, dass das System sich dahingehend sehr sicher ist.
- Datenschutz und Wettbewerbsinteressen können Offenlegungen einschränken, beispielsweise könnte ein Tool zur Bewertung von Krediten nicht ohne größere Probleme die Daten anderer Nutzer:innen offenlegen, um seine Entscheidung zu verdeutlichen.

3. Informationen editieren

Während Disclosure Transparenz schafft, lädt die Methode des Information Editierens zur aktiven Auseinandersetzung ein. Nutzende können Eingaben verändern, um zu sehen, wie das System reagiert. Diese „Was-wäre-wenn“-Szenarien helfen, ein intuitives Verständnis für die Informationsverarbeitung der KI zu entwickeln.

Editieren bei Wohnbrücke e. V.

Die Mitarbeitenden testen verschiedene Annahmen, um die Logik des Systems besser zu verstehen.

Sie wählen wieder das Objekt in der Lindenstraße 12, das aktuell den Eignungswert 0,82 hat.

Nun verändern sie einzelne Eingaben:

Veränderung	Neuer Eignungswert	Wert-Veränderung
Miete sinkt von 850€ auf 700€	0,88	+0,06
Entfernung zur nächsten Sozialstation steigt von 1km auf 2km	0,75	-0,07
Barrierefreiheit entfernt	0,68	-0,14

Die Ergebnisse machen sichtbar:

- Das System legt großen Wert auf Barrierefreiheit (-0,14 Punkte Verlust).
- Auch Entfernung zu Betreuungseinrichtungen wirkt stark.
- Mietkosten sind relevant, aber mit kleinerem Einfluss.

Mitarbeitende verstehen nun, welche Eingaben kritisch sind und wo das System Schwerpunkte setzt. Das hilft ihnen, Daten gezielter zu prüfen oder neue Objekte realistischer einzuschätzen.

Reflexionsfrage

Wie könnten Sie in Ihrem Projekt mit „Was-wäre-wenn“-Szenarien prüfen, ob Ihr System nachvollziehbar arbeitet?

Vorteile des Editierens von Informationen

- Fördert aktives, exploratives Lernen über Systemverhalten.
- Erlaubt es, Hypothesen zu prüfen („Was, wenn das Objekt kleiner wäre?“).
- Macht Zusammenhänge greifbar - besonders hilfreich bei komplexen Modellen.

Grenzen und Risiken des Editierens von Informationen

- Nicht jede Veränderung lässt sich eindeutig interpretieren - insbesondere bei stark vernetzten Merkmalen.
- Gefahr der Überanpassung: Nutzer:innen könnten versuchen, Eingaben „zu optimieren“, statt realistische Daten zu liefern.
- Zusätzlicher technischer Aufwand, da das System flexibel auf Eingabeveränderungen reagieren und Veränderungen visualisieren muss.

4. Zeitverlauf - Entscheidungen nachvollziehbar machen

Die dritte Methode erweitert die Perspektive:

Zeitverlauf meint die Möglichkeit, Veränderungen von Eingaben und Ergebnissen über die Zeit zu beobachten.

Damit wird nachvollziehbar, *wie* und *warum* sich Systementscheidungen entwickeln - ein zentrales Element für Vertrauen und Verantwortlichkeit.

Gerade in Organisationen, in denen Daten laufend aktualisiert werden (z.B. Mietspiegel, Infrastruktur, Energiepreise), kann der Zeitverlauf wichtige Hinweise geben.

Zeitverlauf bei Wohnbrücke e. V.

Das Team verfolgt über drei Monate hinweg, wie sich die Bewertung der **Wohnung Lindenstraße 12** entwickelt:

Datum	Änderungen in den Daten	Eignungswert	Bemerkung
01. Februar	Ursprüngliche Bewertung	0,82	Barrierefrei, mittlere Miete
15. Februar	Neue Info: Heizkosten steigen um 30€	0,79	Leichter Abfall
01. März	Zusatzdaten: Nähe zu Schule (0,8km)	0,83	Verbesserung
01. April	Neue Konkurrenzangebote in der Umgebung	0,76	Sinkende relative Attraktivität

Dieser Verlauf zeigt: Das System reagiert auf neue Informationen dynamisch. Für die Organisation bedeutet das Transparenz - sie kann sehen, warum ein Objekt heute anders bewertet wird als vor einem Monat.

Zudem können die Daten genutzt werden, um Trends zu erkennen:
Verändern sich Bewertungen ganzer Stadtteile?
Wie stark beeinflussen steigende Energiekosten die Eignungswerte insgesamt?

Reflexionsfrage

Wie könnte ein Verlaufs- oder Änderungsprotokoll in Ihrem Projekt helfen, Entwicklungen besser zu verstehen oder zu kommunizieren?

Vorteile von Zeitverläufen

- Erhöht Nachvollziehbarkeit: Zeigt, wie Eingaben und Ergebnisse sich über die Zeit entwickeln.
- Stärkt Verantwortlichkeit: Dokumentiert Entscheidungen und Änderungen, sodass nachvollziehbar ist, wer welche Schritte unternommen hat.
- Macht Lerneffekte sichtbar: Zeigt, wie das System auf Feedback oder Veränderungen reagiert und daraus lernt.
- Fördert Vertrauen: Transparente Entwicklungen schaffen Sicherheit und Vertrauen in das System.

Grenzen und Herausforderungen von Zeitverläufen

- Erhöhter Speicher- und Dokumentationsaufwand.
- Datenschutzfragen (Protokolle enthalten oft sensible Informationen).
- Zu viele Details können den Überblick erschweren.

5. Fazit

Diese drei Methoden - Disclosure, Editieren und Zeitverlauf - verdeutlichen, dass Informationsverarbeitung in KI-Systemen kein einseitiger Vorgang ist.

Sie sind Werkzeuge, um den Dialog zwischen Mensch und Maschine zu fördern.

1

Disclosure schafft Verständnis und Transparenz

2

Editieren ermöglicht aktives Lernen und kontrolliertes Experimentieren

3

Zeitverlauf bietet Nachvollziehbarkeit und Reflexion über Zeit

Gemeinsam bilden sie die Grundlage für KI-Systeme, die nicht nur technisch effizient, sondern auch sozial und ethisch handhabbar sind.

Verarbeitung - Technik

Kursübersicht > KI-Technologien verstehen

Auf technischer Ebene beschreibt „Verarbeiten“, wie ein KI-System Eingabedaten aufnimmt, analysiert und daraus Ergebnisse oder Entscheidungen ableitet. Verschiedene technische Varianten dieser Verarbeitung werden im Folgenden betrachtet.

Überwachtes Lernen

Ein Verfahren des maschinellen Lernens, bei dem ein System anhand von gelabelten Daten trainiert wird, um später neue Eingaben korrekt zu klassifizieren oder Vorhersagen über Zahlenwerte zu treffen.

1. Einführung in das überwachte Lernen

Überwachtes Lernen (engl.: supervised learning) ist eine Methode des maschinellen Lernens, bei der ein System anhand von gelabelten Daten

trainiert wird. Das bedeutet: Für jede Eingabe gibt es ein bekanntes Ergebnis, das als Orientierung dient.

Beispiel:

Stellen Sie sich vor, wir möchten ein System trainieren, handschriftliche Buchstaben zu erkennen. Jede Buchstabendarstellung im Datensatz ist bereits mit dem richtigen Buchstaben **labelt**, also beschriftet. Das System lernt so, die Muster der Buchstaben zu erkennen.

0 7 1 1 4 9 4 3 4 8 2 2 1 8 6 9 0 3 4 0 2 9

0 7 1 1 4 9 4 3 4 8 2 2 1 8 6 9 0 3 4 0 2 9

2. Wichtige Bestandteile des Lernprozesses

Beim überwachten Lernen spielen zwei zentrale Aspekte eine wichtige Rolle: **Labels** (also die Zielwerte) und die **Aufteilung des Datensatzes** in verschiedene Teile.

Labels - die „richtigen Antworten“

Labels sind die Ergebnisse oder Kategorien, die wir unseren Eingaben (den sogenannten Features) zuordnen.

Beispiel Bilderkennung: Wenn wir ein Foto von einem Verkehrsschild in ein System eingeben, dann ist das passende Label z. B. „Stoppschild“ oder „Geschwindigkeitsbegrenzung 50 km/h“. Die Aufgabe des Modells besteht darin, die Eingabe (das Foto) mit dem richtigen Ausgabewert (dem Label) zu verknüpfen.

Einfache Ja/Nein-Fragen: Manchmal ist die Klassifikation binär. Wir möchten ein System trainieren, das entscheidet, ob eine Person für einen Kredit geeignet ist (Ja) oder nicht (Nein). Dazu nutzen wir einen Datensatz mit vielen Finanzinformationen (z. B. Einkommen, Schulden, Zahlungsverhalten). Das System lernt dann, welche dieser Merkmale entscheidend für die Kreditwürdigkeit sind.

Ohne Labels kann das Modell nicht lernen, ob seine Vorhersagen korrekt sind. Sie sind sozusagen die „Lösungsschablonen“, an denen das Modell sein Wissen überprüft.

Aufteilung des Datensatzes - Trainieren und Testen

Damit das Modell nicht nur die vorhandenen Daten auswendig lernt, sondern allgemein gültige Regeln erkennt und Fehler frühzeitig identifiziert werden, wird der Datensatz in verschiedene Teile aufgeteilt:

Trainingsdaten: Dies ist der größte Teil des Datensatzes. Das Modell „sieht“ diese Daten während der Trainingsphase und versucht, Muster darin zu erkennen.

Testdaten: Dieser Teil wird während des Trainings nicht verwendet. Erst wenn das Modell fertig trainiert ist, werden die Testdaten eingesetzt, um zu überprüfen, wie gut das Modell mit neuen, unbekannten Daten zurechtkommt. So können wir feststellen, ob das Modell wirklich gelernt hat oder ob es sich nur die Trainingsbeispiele „gemerkt“ hat.

In vielen Fällen wird zusätzlich noch ein Validierungsdatensatz genutzt, mit dem die Zwischenergebnisse während des Trainings kontrolliert werden.

Beispiel: Bei einem Fruchtbilder-System lernt das Modell an den Trainingsbildern („Apfel“, „Birne“, „Banane“). Anschließend wird mit den Testbildern geprüft, ob es neue Früchte korrekt erkennt.

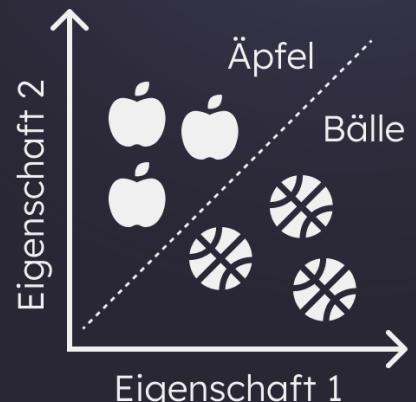
3. Anwendungsbeispiele des überwachten Lernens

Beim überwachten Lernen gibt es zwei zentrale Anwendungsarten: **Klassifikation** und **Regression**.

Klassifikation - Daten in Kategorien einordnen

Bei der Klassifikation werden Eingaben bestimmten Kategorien zugeordnet.

- Beispiel Bilderkennung: Ein Bild wird automatisch die beiden runden Objekte „Äpfel“ oder „Bälle“ erkannt.
- Beispiel Bewertung: Ein System kann Tests oder Aufgaben daraufhin beurteilen, ob sie „vollständig“ oder „unvollständig“ sind.

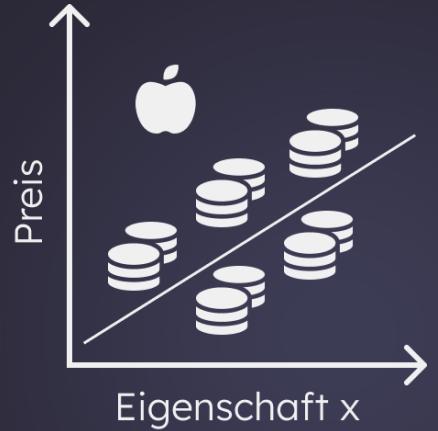


Das Ziel ist es also, qualitative Unterschiede zu erkennen und Daten in klar definierte Gruppen einzuteilen. Beim überwachten Lernen werden diese Klassifikationsregeln nicht von Hand aufgeschrieben. Stattdessen

stellt der Mensch eine Reihe von Beispielen mit den richtigen Labels bereit. Das Modell lernt anhand dieser Beispiele, wie es selbstständig auch neue Daten richtig kategorisieren kann. Der Mensch, der die Labels liefert, fungiert dabei gewissermaßen als „Aufsichtsperson“, die den Algorithmus in die richtige Richtung lenkt.

Regression - Vorhersage von Zahlenwerten

Während es bei der Klassifikation um Kategorien geht, beschäftigt sich die **Regression** mit der Vorhersage **kontinuierlicher Zahlenwerte**, auch von Regressionsproblemen. Statt Labels wie „Apfel“ oder „Ball“ wird also eine Zahl vorhergesagt, z. B. ein Preis für Äpfel im zeitlichen Verlauf, die möglichst nahe am realen Wert liegt.



Besonders wichtig ist dabei, dass die Vorhersage meist nicht von einer einzelnen Einflussgröße, sondern vom Zusammenspiel mehrerer Variablen abhängt. Die Regression untersucht also, wie unterschiedliche Faktoren gemeinsam auf die Zielgröße wirken.

Beispiele

- **Einkommen:** Schätzung des Gehalts einer Person auf Basis von Alter, Ausbildung, Berufserfahrung und Branche.
- **Werbung:** Vorhersage der Klickrate einer Online-Anzeige in Abhängigkeit von Text, Gestaltung, Zielgruppe und bisherigen Nutzerverhalten.
- **Verkehr:** Prognose der Anzahl von Unfällen, wobei Faktoren wie Straßenbedingungen, Wetter, Tageszeit und Geschwindigkeitsbegrenzungen einfließen.
- **Immobilien:** Schätzung des Verkaufspreises einer Wohnung oder eines Hauses anhand von Lage, Wohnfläche, Baujahr, Ausstattung und energetischem Zustand.

4. Herausforderungen beim überwachten Lernen

Beim überwachten Lernen gibt es zwei zentrale Probleme, die die Leistungsfähigkeit eines Modells stark beeinflussen können: **Overfitting** und **Underfitting**.



Overfitting

Das Modell lernt die Trainingsdaten zu genau, erkennt keine allgemeinen Muster. Es liefert sehr gute Ergebnisse bei Trainingsdaten, aber schlechte Vorhersagen bei neuen Daten.

Beispiel: Ein Modell wurde nur mit Daten älterer Menschen trainiert und liefert für jüngere Menschen ungenaue Vorhersagen.

Underfitting

Das Modell erkennt die Zusammenhänge in den Daten nicht ausreichend und liefert ungenaue Vorhersagen, auch bei Trainingsdaten.

Beispiel: Eine KI zur Spam-Erkennung, die nur kurze Texte als Spam markiert, ignoriert andere wichtige Merkmale wie Schreibweisen oder Handlungsaufforderungen.

Das Ziel beim überwachten Lernen ist, ein ausgewogenes Modell zu entwickeln, das die richtigen Muster erkennt, ohne sich zu sehr an die Trainingsdaten zu klammern. Nur so kann es auf neue Daten gut generalisieren und verlässliche Vorhersagen treffen.

1

Gelabelte und ausreichende Datenmenge: Damit ein Modell erfolgreich überwacht lernen kann, müssen einige Voraussetzungen erfüllt sein. Zunächst sind **gelabelte Daten** erforderlich, denn ohne bekannte Ergebnisse kann das System nicht lernen. Außerdem ist eine **ausreichende Datenmenge** wichtig, um zu verhindern, dass das Modell unteranpasst und keine sinnvollen Muster erkennt (Underfitting).

2

Qualität und Vielfalt der Daten: Neben der Menge spielen auch die **Datenqualität und Vielfalt** eine entscheidende Rolle. Ein vielfältiger Datensatz hilft, dass das Modell nicht nur die Trainingsdaten auswendig lernt, sondern die zugrunde liegenden Zusammenhänge erkennt und auf neue Daten übertragen kann. So wird Overfitting vermieden. Dabei ist zu beachten, dass natürlich nicht "irgendwelche" Daten genutzt werden sollten. Es geht vielmehr darum, einen möglichst diversen Datensatz zu dem spezifischen Problem zu haben, das das System lösen soll.

3

Schließlich ist eine **klare Zieldefinition** notwendig: Soll das Modell Eingaben klassifizieren, also Kategorien zuordnen, oder numerische Werte vorhersagen, also eine Regression durchführen? Eine präzise Zielsetzung bestimmt den Aufbau des Modells und die Auswahl der geeigneten Daten.

Einsatz von Testdaten: Um zu überprüfen, wie gut ein Modell auf neue Daten reagiert, werden **Testdaten** eingesetzt. Diese Daten wurden beim Training nicht verwendet und ermöglichen eine realistische Einschätzung der Leistungsfähigkeit. Zur Bewertung des Modells werden verschiedene **Kennzahlen** herangezogen, wie z.B. Genauigkeit oder Fehlermaße.

Zusammenfassung

Überwachtes Lernen funktioniert nur mit **gelabelten Daten**. Modelle unterscheiden sich je nach Ziel in **Klassifikation**, also der Einordnung in Kategorien, und **Regression**, also der Vorhersage von Zahlenwerten. Ziel ist immer ein ausgewogenes Modell, das weder Overfitting noch Underfitting zeigt. Eine **gute Datenbasis** - qualitativ hochwertig, vielfältig und ausreichend groß - ist entscheidend für den Erfolg eines Modells.

Unüberwachtes Lernen

Beschreibt ein Verfahren des maschinellen Lernens, bei dem ein System ohne vorgegebene Labels in den Daten eigenständig Strukturen, Muster oder Gruppen erkennt.

1. Einführung in das unüberwachte Lernen

Im Gegensatz zum überwachten Lernen arbeitet das **unüberwachte Lernen** (engl.: unsupervised learning) mit Daten, die **keine Labels** enthalten. Das bedeutet: Es gibt keine vorgegebenen Kategorien oder Ergebnisse, an denen sich das System orientieren kann. Stattdessen versucht das Modell, selbstständig Strukturen und Muster in den Daten zu erkennen.

Beispiel für unlabeled Daten sind etwa große Mengen von PDF-Antragsformularen oder Videoaufzeichnungen von Parlamentsdebatten (z.B. auf openparliament.tv). Diese Daten liegen zwar in großer Zahl vor, sind aber nicht vorab in Kategorien eingeteilt oder beschriftet.

Unüberwachtes Lernen kommt immer dann zum Einsatz, **wenn noch keine Kategorien existieren** oder wenn es darum geht, Daten **neu zu ordnen oder in Gruppen einzuteilen**.

Die zentralen Fragestellungen lauten daher:

- Wie lässt sich Struktur in einer großen Menge unlabeled Daten entdecken?
- Wie können diese Daten sinnvoll gruppiert oder zusammengefasst werden?

2. Clustering

Eine der wichtigsten Methoden im unüberwachten Lernen ist das **Clustering**. Darunter versteht man die **Gruppierung von Daten nach ihrer Ähnlichkeit**. Ziel ist es, Datensätze so anzurufen, dass ähnliche Daten in einer Gruppe - einem sogenannten Cluster - zusammengefasst werden.

Beispiel: Stellen Sie sich vor, es liegen viele verschiedene Antragsformulare vor. Ohne vorherige Labels könnte das System sie automatisch in Cluster einteilen, etwa in:

- **Sozialhilfe**
- **Wohngeld**
- **Elterngeld**

So entstehen sinnvolle Gruppen, die eine spätere Analyse oder Verarbeitung erleichtern.

Ein wichtiger Aspekt beim Clustering ist die **Anzahl der Cluster**:

- Mit **mehr Clustern** entstehen feinere Gruppen (hohe Granularität), die Unterschiede sehr detailliert darstellen.
- Mit **weniger Clustern** entstehen gröbere Gruppen (geringe Granularität), die nur die wichtigsten Unterschiede berücksichtigen.

3. Beispiel aus der Praxis

Ein praktisches Beispiel für unüberwachtes Lernen ist die **Auswertung handschriftlicher Dokumente ohne Labels**. Das Ziel besteht darin, verschiedene Buchstaben voneinander zu unterscheiden, ohne dass diese zuvor beschriftet wurden.

Das System geht dabei so vor, dass sie ähnliche Formen automatisch in Gruppen, also **Cluster**, einteilt. So könnten beispielsweise alle „A“s in einem Cluster landen, alle „B“s in einem anderen, und so weiter. Auf diese Weise lassen sich Strukturen in den Daten erkennen, ohne dass vorherige Kennzeichnungen nötig sind.



4. Herausforderungen beim unüberwachten Lernen

Im Gegensatz zum überwachten Lernen bringt das unüberwachte Lernen besondere Schwierigkeiten mit sich. Da **keine Labels** vorhanden sind, gibt es auch keinen klassischen **Test-Datensatz**, mit dem die Ergebnisse überprüft werden könnten.

Das bedeutet: Es gibt keine eindeutig „richtigen“ oder „falschen“ Antworten. Stattdessen zeigt das Modell lediglich mögliche Strukturen auf, die sinnvoll erscheinen können - oder auch nicht. Die **Bewertung der Ergebnisse** ist daher deutlich komplexer und erfordert meist zusätzliches Fachwissen oder weitere Analysen.

Einsatzgebiete: Während überwachte Lernverfahren vor allem dann sinnvoll sind, wenn konkrete Vorhersagen oder Klassifikationen benötigt werden (z.B. Kreditwürdigkeitsprüfung, medizinische Diagnose), eignet sich unüberwachtes Lernen besonders für Exploration, Mustererkennung und Clusterbildung, wenn keine Labels vorliegen und man zunächst Strukturen oder Zusammenhänge in den Daten entdecken möchte.

Zusammenfassung

Unüberwachtes Lernen kommt ohne Labels aus. Sein Ziel ist es, **Strukturen oder Gruppen (Cluster)** in Daten zu erkennen. Die häufigste Methode dabei ist das **Clustering**, bei dem ähnliche Daten automatisch zusammengefasst werden. Allerdings ist die Bewertung der Ergebnisse wesentlich schwieriger als beim überwachten Lernen, da es keine klaren Antworten gibt.

Bestärkendes Lernen

Ein Verfahren des maschinellen Lernens, bei dem ein System durch Versuch und Irrtum lernt und sein Verhalten anhand von Belohnungen oder Bestrafungen schrittweise verbessert, um langfristig den größtmöglichen Erfolg zu erzielen.

1. Einführung in das bestärkende Lernen

Beim **bestärkenden Lernen** (engl.: reinforcement learning) lernt ein KI-System durch **Versuch und Irrtum**. Es erhält **Belohnungen**, wenn es gute Entscheidungen trifft, und **Bestrafungen**, wenn sein Verhalten nicht zielführend ist. Dieser Ansatz ähnelt dem **operanten Konditionieren** in der Psychologie, bei dem Verhalten durch positives oder negatives Feedback geformt wird.

Ein zentrales Merkmal ist, dass es **keine direkten „richtigen“ Lösungen** gibt. Das System lernt stattdessen aus **Feedback, das oft erst nach mehreren Schritten erfolgt**.

Typische Anwendungsbereiche sind Situationen, in denen ein KI-Akteur eigenständig handeln muss, wie etwa:

- **Selbstfahrende Autos**, die erst nach einer Reihe von Entscheidungen Rückmeldung über deren Qualität erhalten.
- **Spiele**, bei denen der Erfolg erst am Ende des Spiels sichtbar wird.

2. Funktionsweise

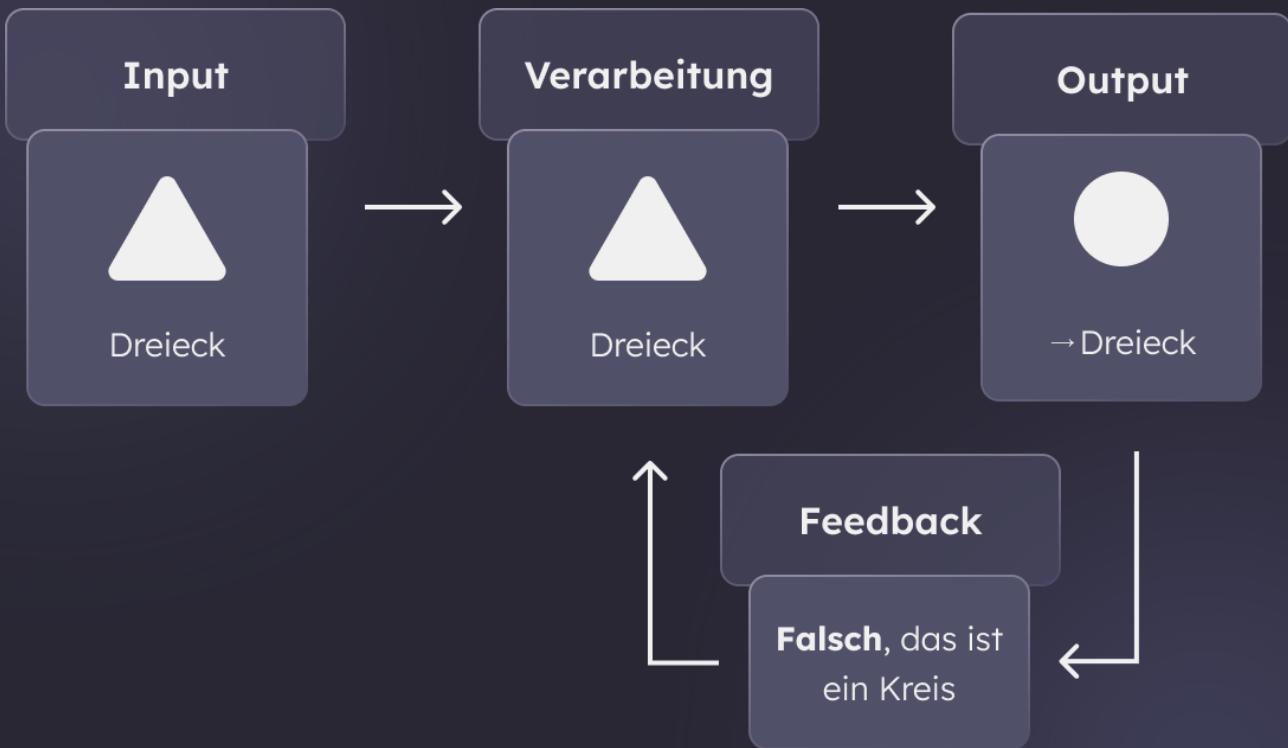
Das System befindet sich in einer **Umgebung**, mit der es ständig interagiert.

- Für jede Aktion erhält es eine Rückmeldung in Form einer Belohnung oder Bestrafung.
- Diese Erfahrungen werden gespeichert und genutzt, um die **Strategie (Policy)** zu verbessern.
- Das Ziel besteht darin, **langfristig die maximale Belohnung** zu erreichen - nicht nur kurzfristig richtige Entscheidungen zu treffen.

3. Besonderheiten

Bestärkendes Lernen unterscheidet sich deutlich vom überwachten und unüberwachten Lernen:

- Es werden **keine großen Datensätze** benötigt, da das System durch direkte Interaktion mit seiner Umgebung lernt.
- Es gibt **keine festen Labels oder Testdaten**.
- Stattdessen basiert der gesamte Lernprozess auf **Erfahrungen**.
- Diese Methode ist besonders geeignet für **dynamische Umgebungen**, in denen sich Situationen ständig ändern und Flexibilität gefragt ist.



Zusammenfassung

Bestärkendes Lernen basiert auf dem Prinzip von **Belohnung und Bestrafung**. Das System lernt schrittweise aus seinen Erfahrungen und passt sein Verhalten immer weiter an. Es eignet sich besonders für **komplexe, dynamische Umgebungen** wie die Robotik. Das übergeordnete Ziel ist dabei stets die **Maximierung der langfristigen Belohnung**.

Verarbeitung - Gestaltung

Kursübersicht > [KI-Technologien verstehen](#)

1. Shapley-Werte - Wer trägt welchen Anteil an der Entscheidung?

Wenn KI-Systeme Entscheidungen treffen, ist das Ergebnis oft nur eine Zahl oder Bewertung. Aber was steckt dahinter? Welche Eingaben haben welchen Anteil daran, dass ein bestimmtes Resultat herauskommt? Genau diese Frage versuchen **Shapley-Werte** zu beantworten - eine Methode, die ursprünglich aus der **Spieltheorie** stammt und heute zu den wichtigsten Werkzeugen gehört, um **Entscheidungsprozesse in KI-Systemen transparent** zu machen.

Vom Spiel zur Entscheidung - die Grundidee

Der Name geht auf den Mathematiker **Lloyd Shapley** zurück, der sich mit fairer Verteilung von Gewinnen in kooperativen Spielen beschäftigte. Stellen wir uns ein Spiel vor, in dem mehrere Personen gemeinsam ein Ziel erreichen - zum Beispiel ein Team, das zusammen einen Gewinn erspielt. Shapleys Frage lautete: *Wie lässt sich dieser Gewinn gerecht auf die Mitspielenden verteilen, abhängig davon, wie stark jede Person zum Gesamterfolg beigetragen hat?*

Diese Idee lässt sich erstaunlich gut auf **KI-Modelle** übertragen:

- Das „**Spiel**“ ist in unserem Fall das Modell, das eine Entscheidung trifft oder eine Vorhersage berechnet.
- Die „**Spieler**“ sind die **Eingabeveriablen**, also beispielsweise Lage, Größe und Kosten einer Wohnung.
- Der „**Gewinn**“ ist die tatsächliche Vorhersage - etwa der Eignungswert eines Gebäudes für eine Wohngruppe - im Vergleich zur durchschnittlichen Bewertung aller untersuchten Objekte.

Das Ziel der Methode: herauszufinden, **wie stark jede einzelne Variable zum Ergebnis beigetragen hat** - und zwar fair, indem alle möglichen Kombinationen von Merkmalen berücksichtigt werden.

Ein praktisches Beispiel

Eine gemeinwohlorientierte Organisation möchte mit Unterstützung eines KI-Systems einschätzen, **welche Immobilien sich für Menschen in Not eignen.**

Das System bewertet verschiedene Objekte anhand von Merkmalen wie:

- Größe der Immobilie
- Entfernung zu einer Betreuungseinrichtung
- Ausstattung und Zustand
- monatliche Mietkosten

Das Modell berechnet für jede Immobilie einen Eignungswert. Eine Wohnung erhält z. B. **nur 0,35 Punkte** (auf einer Skala von 0 bis 1), eine andere **0,75 Punkte**.

Die Verantwortlichen möchten nun verstehen: Warum schneidet die erste so viel schlechter ab?

Mit **Shapley-Werten** lässt sich nachvollziehen, wie stark jedes Merkmal dazu beigetragen hat - etwa:

Merkmal	Shapley-Wert (Beitrag zur Vorhersage)
Größe	+0,15
Lage (Nähe zu Betreuungseinrichtung)	+0,25
Kosten	-0,30
Zustand	-0,05

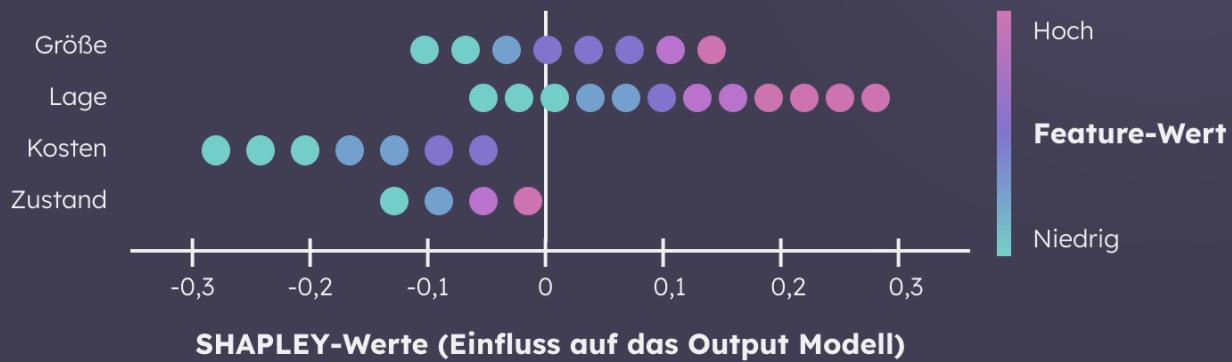
Solche Werte zeigen: Die höheren Kosten und der schlechte Zustand haben die Bewertung stark gedrückt, während Lage und Größe positiv gewirkt haben. Damit lässt sich nicht nur die Entscheidung des Systems besser verstehen, sondern auch diskutieren, **ob die Gewichtung dieser Faktoren fair und sinnvoll ist.**

Wie Shapley-Werte berechnet werden

In der Praxis wird für jede Variable berechnet, **wie sich das Ergebnis verändert**, wenn sie zum Modell „hinzugefügt“ oder „weggelassen“ wird - und das über alle möglichen Kombinationen von Variablen hinweg.

Da die vollständige Berechnung bei vielen Variablen sehr aufwändig ist, arbeiten Anwendungen meist mit **Näherungsverfahren** oder **Stichproben**.

Viele gängige KI-Frameworks (z. B. SHAP in Python) bieten fertige Implementierungen, die diese Berechnungen automatisiert durchführen. Hier eine beispielhafte Darstellung:



Worauf Sie achten sollten

Die **Interpretation von Shapley-Werten** hängt immer vom **Referenzdatensatz** ab - also den Daten, die als Vergleich herangezogen werden, wenn einzelne Merkmale „fehlen“. Ein unausgewogener oder nicht repräsentativer Datensatz kann zu **verzerrten Ergebnissen** führen. Außerdem gilt:

Shapley-Werte sagen **nichts darüber aus, wie sich das Ergebnis**

verändern würde, wenn ein Merkmal tatsächlich geändert würde. Sie beschreiben nur, **welchen Einfluss es im aktuellen Modell** hat.

Beispiel: Der Shapley-Wert sagt nicht, „wenn die Wohnung größer wäre, würde sie besser bewertet“, sondern nur, „im aktuellen Datensatz tragen größere Wohnungen im Schnitt positiv zur Bewertung bei“.

Grenzen und Aufwand

Die Methode ist **rechenintensiv**, besonders bei komplexen oder hochdimensionalen Modellen. Für große Sprachmodelle (LLMs) oder neuronale Netze lassen sich daher meist nur **Teilsets von Merkmalen** untersuchen. Dennoch sind Shapley-Werte eines der **robustesten und anerkanntesten Verfahren**, um **Nachvollziehbarkeit und Fairness** in KI-Entscheidungen zu fördern.

2. Partial Dependence Plots

Während **Shapley-Werte** uns zeigen, *welchen Anteil* einzelne Faktoren an einer Entscheidung haben, helfen **Partial Dependence Plots (PDPs)** dabei, *wie genau* diese Faktoren den Ausgang eines Modells beeinflussen.

Mit anderen Worten: Während Shapley-Werte erklären, **wer wie stark mitspielt**, zeigen PDPs, **wie das Zusammenspiel aussieht**.

Was zeigt ein PDP?

Ein **Partial Dependence Plot** stellt visuell dar, **wie sich der vorhergesagte Wert eines Modells verändert**, wenn sich ein oder mehrere Eingabefaktoren verändern - und alle anderen Faktoren konstant gehalten werden.

Dadurch lässt sich nachvollziehen, **welche Beziehung zwischen einem Merkmal und dem Ergebnis** besteht:

- Steigt der Wert des Merkmals → steigt oder fällt dann auch die Bewertung durch das System?
- Gibt es Schwellenwerte, ab denen sich der Effekt ändert?
- Wie wirken zwei Merkmale in Kombination aufeinander?

Beispiel: Immobilienbewertung für eine gemeinwohlorientierte Organisation

Kommen wir noch einmal zu unserem Beispiel zurück:

Eine Organisation nutzt ein KI-System, um zu beurteilen, welche Immobilien sich für Menschen in Not eignen. Das System analysiert Merkmale wie Größe, Mietkosten, Baujahr und Zustand der Immobilie und gibt am Ende einen Eignungswert zwischen 0 (ungeeignet) und 1 (sehr geeignet) aus.

Ein **Partial Dependence Plot** könnte nun etwa zeigen, **wie das Baujahr der Immobilie** den Eignungswert beeinflusst.

In der Abbildung verläuft beispielhaft eine aufsteigende Kurve:

Je neuer das Gebäude, desto höher die Bewertung durch das System.

Das Modell scheint also neuere Immobilien systematisch positiver zu bewerten.



Eine zweite, erweiterte beispielhafte Darstellung zeigt zusätzlich den **Zustand der Immobilie** als zweiten Faktor in einer farbigen **Heatmap**. Darin erkennen wir, dass der Effekt des Baujahrs **abhängig von der Qualität** ist:

- Bei **sehr guter Qualität** spielt das Alter kaum noch eine Rolle - selbst ältere Gebäude erhalten hohe Bewertungen.
- Bei **schlechter Qualität** verstärkt sich dagegen der Effekt des Baujahres deutlich - alte und schlecht erhaltene Gebäude werden klar abgewertet.

Solche Visualisierungen helfen, **Zusammenhänge intuitiv zu erkennen** - auch ohne tief in die Modelllogik einzusteigen.

Wie ein PDP funktioniert

Ein PDP wird erstellt, indem man **den Wert eines Merkmals systematisch variiert** (z. B. das Baujahr von 1950 bis 2020) und dabei beobachtet, **wie sich die Modellvorhersage verändert**, während alle anderen Merkmale konstant bleiben.

Diese Veränderungen werden dann **als Kurve oder Fläche** visualisiert.

- **1D-PDPs** zeigen die Auswirkung **eines einzelnen Faktors** (z.B. Baujahr).
- **2D-PDPs** zeigen die **Wechselwirkung zweier Faktoren** (z.B. Baujahr und Zustand).

Vorteile und Grenzen von PDPs

Vorteile

- Sie sind **intuitiv und leicht verständlich**.
- Sie helfen, **nichtlineare Zusammenhänge** zu erkennen (z.B. Schwellen oder Sättigungseffekte).
- Sie unterstützen Teams dabei, **Modellentscheidungen visuell zu prüfen** - auch ohne technisches Detailwissen.

Grenzen

- PDPs setzen voraus, dass **Faktoren unabhängig voneinander** betrachtet werden können.
In der Realität sind Variablen aber oft **korreliert** - etwa, dass neuere Gebäude meistens teurer sind.
Dadurch können im PDP **unrealistische Kombinationen** entstehen, z.B. „sehr altes Gebäude mit extrem hohem Zustand“, die im echten Datensatz gar nicht vorkommen.
Das Modell zeigt dann zwar eine scheinbare Abhängigkeit, die **in der Praxis aber keine Bedeutung** hat.
- Zudem können PDPs **nur ein oder zwei Merkmale gleichzeitig** darstellen. Für komplexere Interaktionen braucht es andere Verfahren (z.B. Shapley- oder ICE-Plots).

Beispiel für eine Fehlinterpretation

Angenommen, ein PDP zeigt, dass die Bewertung einer Immobilie **stark mit steigender Wohnfläche** zunimmt.

Gleichzeitig sind in den Daten aber größere Wohnungen fast immer **auch teurer**.

Der Plot könnte dann fälschlicherweise suggerieren, dass nur die Größe entscheidend ist, obwohl in Wahrheit die **Mietkosten** der eigentliche Treiber sind.

Solche Effekte zu erkennen, ist Teil einer reflektierten Anwendung dieser Methode - und genau deshalb sind PDPs besonders nützlich, **wenn sie gemeinsam mit anderen Verfahren** wie Shapley-Werten eingesetzt werden.

3. Permutation Feature Importance

Wenn wir verstehen wollen, welche Eingabefaktoren für die Entscheidungen eines KI-Systems tatsächlich wichtig sind, reicht es nicht immer, nur zu wissen, *wie* sie wirken. Oft möchten wir auch wissen, *wie stark* sich das Wegfallen oder Verfälschen eines Faktors oder dessen Reihenfolge im System auf das Ergebnis auswirkt.

Eine Methode, die genau das messbar macht, ist die Permutation Feature Importance (PFI) - zu Deutsch: die Bedeutung eines Merkmals durch Zufallsdurchmischung.

Grundidee: Was passiert, wenn wir die Reihenfolge verändern?

Die Idee hinter der Permutation Feature Importance ist erstaunlich einfach und intuitiv:

Wenn ein bestimmtes Merkmal für die Vorhersage eines Modells wichtig ist, sollte das Ergebnis schlechter werden, sobald wir die Werte dieses Merkmals zufällig durcheinander mischen.

Dadurch wird die ursprüngliche Beziehung zwischen diesem Merkmal und der Zielgröße zerstört.

Je stärker sich der Vorhersagefehler dadurch erhöht, desto wichtiger ist dieses Merkmal für das Modell.

Ein Merkmal, dessen Vertauschung keine oder kaum Veränderungen bewirkt, hat dagegen wenig Einfluss auf die Entscheidungen des Systems - das Modell „ignoriert“ es in gewisser Weise.

Beispiel: Bewertung von Wohnraum für Menschen in Not

Bleiben wir bei unserem laufenden Beispiel.

Das KI-System bewertet Immobilien danach, wie gut sie sich als Wohnraum für Menschen in Not eignen. Eingabefaktoren sind unter anderem:

- Wohnfläche
- Mietkosten
- Entfernung zur Betreuungseinrichtung
- Baujahr
- Ausstattung (Zustand, Barrierefreiheit, etc.)

Nun wollen die Verantwortlichen verstehen, welche dieser Merkmale die Entscheidung des Systems am stärksten beeinflussen.

Dazu wird folgende Vorgehensweise angewandt:

1

Das Modell sagt zunächst mit den echten Daten die Eignungswerte der Immobilien vorher.

2

Anschließend wird eine Spalte (z.B. das Baujahr) zufällig durchmischt.

Damit verliert das Modell die echte Verbindung zwischen Baujahr und Bewertung.

3

Das Modell erstellt erneut Vorhersagen - diesmal mit den „vertauschten“ Werten.

4

Nun wird gemessen, wie stark die Vorhersage an Genauigkeit verliert.

5

Dieser Unterschied wird für jedes Merkmal berechnet, oft mehrfach wiederholt und gemittelt.

Das Ergebnis zeigt, wie sehr das Modell auf jedes Merkmal angewiesen ist.

Beispielsweise könnte sich herausstellen:

Merkmal	Anstieg des Fehlers (in %)	Bedeutung
Mietkosten	+22%	Sehr wichtig
Entfernung zur Betreuungseinrichtung	+15%	Wichtig
Baujahr	+7%	Mittelwichtig
Barrierefreiheit	+4%	Gering
Klimaanlage vorhanden	+1%	Unwichtig

In diesem Fall zeigt sich:

Das Alter der Immobilie hat einen deutlich stärkeren Einfluss auf die Bewertung als etwa die Ausstattung mit Klimaanlage. Das System gewichtet also manche Merkmale stark, andere kaum.

Der technische Ablauf in Kürze

1

Vorhersage mit Originaldaten: Das Modell schätzt, wie geeignet jede Immobilie ist.

2

Permutation eines Merkmals: Die Werte eines Faktors (z. B. Baujahr) werden zufällig neu angeordnet.

3

Vorhersage mit permutierten Daten: Das Modell trifft erneut Entscheidungen, nun ohne die echte Beziehung zwischen Baujahr und Bewertung.

4

Vergleich der Fehler: Wie sehr hat sich der Vorhersagefehler erhöht?

5

Wiederholung: Die Schritte werden mehrfach wiederholt und gemittelt, um zufällige Schwankungen auszugleichen.

Das Ergebnis: ein Wichtigkeitswert pro Merkmal, der zeigt, welche Faktoren das Modell wirklich nutzt, um seine Entscheidungen zu treffen.

Warum das relevant ist

PFI ist besonders dann nützlich, wenn Sie wissen wollen, ob Ihr System auf die richtigen Dinge schaut.

Wenn etwa das Modell in unserem Beispiel feststellt, dass die Postleitzahl oder der Mietpreis besonders großen Einfluss hat, könnte das ein Hinweis auf versteckte soziale oder geografische Verzerrungen (Bias) sein.

Die Methode kann also helfen, Fairness-Probleme frühzeitig zu erkennen und zu adressieren.

Zudem berücksichtigt PFI automatisch auch **Wechselwirkungen zwischen Variablen**:

Wenn sich zwei Merkmale gegenseitig beeinflussen (z. B. Baujahr und Zustand), wird dieser Effekt mitgemessen - denn durch das Durchmischen werden alle Abhängigkeiten zwischen diesem Merkmal und den anderen gleichzeitig aufgehoben.

Vorteile und Grenzen

Vorteile

- **Einfach und modellunabhängig:** Sie funktioniert mit fastem jedem KI-Modell, ohne dass es neu trainiert werden muss.
- **Erhöht Transparenz und Fairness:** Zeigt auf, welche Faktoren in der Praxis tatsächlich Einfluss nehmen.
- **Berücksichtigt Interaktionen:** Auch Kombinationseffekte zwischen Variablen fließen mit ein.
- **Verständlich für Nicht-Expert:innen:** Das Prinzip „wir mischen und schauen, was passiert“ ist leicht nachvollziehbar.

Grenzen und Herausforderungen

- **Kein Verständnis der Richtung:** PFI zeigt nur, *wie stark* ein Faktor wirkt - nicht *ob er positiv oder negativ* wirkt.
- **Zufälligkeit und Streuung:** Da die Methode auf Zufallsdurchmischung basiert, können Ergebnisse schwanken. Eine Mehrfach-Wiederholung (und Mittelung) stabilisiert die Ergebnisse, kostet aber **mehr Rechenzeit**.
- **Keine Aussage über Ursache und Wirkung:** PFI misst Bedeutung, nicht Kausalität. Ein hoher Wert bedeutet nicht, dass dieses Merkmal *verursacht*, dass das Ergebnis so ausfällt - nur, dass es eng damit verknüpft ist.

Beispielhafte Fehlinterpretation

Wenn die Organisation feststellt, dass das Merkmal „Postleitzahl“ sehr wichtig für die Bewertung ist, bedeutet das **nicht**, dass die Lage per se problematisch ist. Es kann sein, dass in bestimmten Stadtteilen schlicht häufiger Immobilien mit schlechter Ausstattung vorkommen – und das Modell diesen Zusammenhang gelernt hat.

PFI hilft also, solche **versteckten Muster sichtbar zu machen**, erfordert aber immer **eine menschliche Interpretation**, um Fehlschlüsse zu vermeiden.

06 Output - Technik

[Kursübersicht](#) > [KI-Technologien verstehen](#)

Einleitung: Warum der Output entscheidend ist

Der Output eines KI-Systems ist das sichtbare Ergebnis aller vorhergehenden Verarbeitungsschritte - und damit die Grundlage, auf der Menschen und andere Systeme weiterarbeiten. Er entscheidet darüber, wie nützlich, verständlich und anschlussfähig ein System im konkreten Anwendungskontext ist.

Besonders in gemeinwohlorientierten Organisationen, in denen Entscheidungen oft soziale Folgen haben, ist es wichtig, nicht nur das Ergebnis selbst, sondern auch dessen Art, Herkunft und Aussagekraft zu verstehen. Denn nicht jeder Output ist gleich: Systeme können Texte bewerten, Wahrscheinlichkeiten berechnen, Prognosen abgeben oder sogar neue Daten erzeugen.

Dieses Kapitel gibt einen Überblick über die wichtigsten Output-Formen von KI-Systemen und erläutert, wie sie gelesen, interpretiert und kritisch hinterfragt werden können.

1. Kategorische Outputs

Viele KI-Systeme ordnen Daten in Kategorien ein. Diese Form des Outputs findet sich häufig bei Klassifikationsaufgaben - etwa, wenn ein System E-Mails als „Spam“ oder „Nicht“-Spam“ markiert, oder wenn ein Textanalysetool die Stimmung eines Textes als „positiv“, „neutral“ oder „negativ“ einstuft.

Für gemeinwohlorientierte Organisationen können solche Modelle zum Beispiel eingesetzt werden, um eingehende Anträge zu sortieren oder Texte nach Themen zu gruppieren. Wichtig ist dabei zu verstehen, dass eine Kategorie nicht immer eindeutig „richtig“ ist: ein analysierter Text kann sowohl sachlich als auch emotional gefärbt sein und somit zwei Kategorien zugeordnet werden.

Ein zentrales Merkmal kategorialer Outputs ist die Wahrscheinlichkeit, mit der eine Zuordnung vorgenommen wird. Ein Modell kann etwa schätzen, dass eine Nachricht mit 70 % Wahrscheinlichkeit „positiv“ ist, mit 20 % „neutral“ und mit 10 % „negativ“.

Beispiel aus der Praxis

Eine Organisation, die Bürgeranfragen automatisch vorsortieren möchte, nutzt ein KI-Modell, das E-Mails in die Kategorien „Lob“, „Beschwerde“, „Antrag“ und „Sonstiges“ einteilt. Eine Nachricht wird als „Antrag“ klassifiziert, mit einer Wahrscheinlichkeit von 55 %. Auf den ersten Blick mag das ausreichend erscheinen - doch die zweitwahrscheinlichste Kategorie „Beschwerde“ liegt bei 40 %. Es wäre also riskant, die E-Mail automatisch einem Bearbeitungsprozess zuzuweisen, ohne diesen Unsicherheitsbereich zu berücksichtigen.

Merksatz: Kategorische Outputs sollten nie als absolute Wahrheiten interpretiert werden. Ein Blick auf die zweit- oder drittwahrscheinlichste Kategorie kann helfen, Fehlentscheidungen zu vermeiden.

2. Pattern Matching

Unter *Pattern Matching* versteht man das Erkennen wiederkehrender Muster in Daten. Dabei sucht ein KI-System nach regelmäßigen Abfolgen, Beziehungen oder Ähnlichkeiten zwischen Datenpunkten.

Diese Methode wird vor allem dort eingesetzt, wo es um zeitliche oder sequentielle Zusammenhänge geht - etwa in der Analyse von Verlaufsmustern, Ereignisfolgen oder Textstrukturen.

Beispiel aus der Praxis

Ein gemeinnütziges Gesundheitsprojekt analysiert Gesprächsverläufe aus einer Online-Beratung. Das System erkennt, dass Anfragen, in denen Wörter wie „*überfordert*“, „*allein*“ oder „*nicht mehr weiter*“ vorkommen, häufig in einer Eskalation enden, wenn innerhalb von 24 Stunden keine Antwort erfolgt. Diese Mustererkennung hilft der Organisation, Prioritäten zu setzen und gefährdete Fälle schneller zu identifizieren.

Pattern Matching liefert also keine Bewertung, sondern erkennt Strukturen, die menschlichen Entscheidenden Hinweise geben. Dabei gilt: Muster sind immer statistisch - sie zeigen Wahrscheinlichkeiten, keine Notwendigkeiten.

Reflexionsfrage: Welche Risiken könnten entstehen, wenn eine Organisation erkannte Muster als feste Regeln interpretiert?

3. Numerische Prädiktion

Numerische Prädiktionen gehören zu den wichtigsten Outputs vieler KI-Systeme. Statt Kategorien liefert das Modell hier **Zahlenwerte**, die als Bewertung, Wahrscheinlichkeit oder Score dienen.

Ziel ist es, eine mathematische Funktion zu finden, die auf Basis der Eingabedaten einen quantitativen Output erzeugt - zum Beispiel den geschätzten Wert einer Immobilie, die Wahrscheinlichkeit eines Ereignisses oder einen Prioritätsscore.

Beispiel aus der Praxis

Eine soziale Einrichtung möchte geeigneten Wohnraum für Familien in Not finden. Das KI-Modell bewertet 100 verfügbare Wohnungen anhand von Größe, Lage, Zustand und Mietkosten.

Der Output sieht vereinfacht so aus:

Wohnung	KI-Bewertung (Score 0-100)
A	82
B	76
C	41
D	59

Je höher der Wert, desto besser die Eignung. Die Organisation kann diese Bewertung als Orientierungshilfe nutzen - sollte aber immer prüfen, **welche Merkmale den Score am stärksten beeinflusst haben** (z. B. über Methoden wie Shapley-Werte oder Permutationsanalysen).

Numerische Prädiktionen ermöglichen es auch, die **Abweichung** eines Werts zu quantifizieren - ein Vorteil, wenn Systeme durch Lernen schrittweise optimiert werden sollen.

Merksatz: Numerische Outputs machen Unterschiede messbar - aber nicht automatisch erklärbar. Transparente Modelle helfen, Zahlen richtig einzuordnen.

4. Synthetische Ergebnisse

Synthetische Outputs entstehen, wenn ein KI-System **neue Daten erzeugt**, anstatt vorhandene zu bewerten. Dazu gehören automatisch generierte Texte, Bilder, Musik oder Simulationen.

Im gemeinwohlorientierten Bereich kann diese Art des Outputs beispielsweise genutzt werden, um **Situationen zu simulieren oder alternative Szenarien zu prüfen**.

Beispiel aus der Praxis

Ein Stadtentwicklungsprojekt möchte ermitteln, wie sich neue Grünflächen auf die Lebensqualität in einem Viertel auswirken könnten. Das KI-System erzeugt auf Basis vorhandener Umweltdaten und Bürgerbefragungen synthetische Szenarien, die verschiedene Kombinationen von Bebauungsdichte, Verkehrsaufkommen und Grünanteil zeigen. Diese Simulationen helfen, Entscheidungen über die Stadtplanung zu unterstützen, ohne reale Eingriffe vornehmen zu müssen.

Synthetische Ergebnisse bieten also große Chancen für Planung, Simulation und Bildung. Gleichzeitig stellt sich die Frage nach **ethischer Verantwortung**: Je realistischer synthetische Daten sind, desto größer ist das Risiko, dass sie mit echten verwechselt oder missbräuchlich verwendet werden.

Merksatz: Synthetische Daten sind Werkzeuge zur Exploration -
keine Abbilder der Realität.

5. Forecasting

Forecasting ist die Vorhersage zukünftiger Entwicklungen auf Basis vergangener und aktueller Daten. Anders als bei numerischen Prädiktionen liegt hier der Fokus auf Trends über Zeiträume hinweg.

Beispiel aus der Praxis

Eine Organisation, die Lebensmittelpenden koordiniert, nutzt Forecasting, um den künftigen Bedarf an bestimmten Produkten zu planen.

Das Modell zeigt:

- Wenn die Temperaturen im Winter unter 0°C fallen, steigt die Nachfrage nach warmen Mahlzeiten um durchschnittlich 18%.
- In Ferienzeiten sinkt die Spendenbereitschaft um rund 12%.

Diese Informationen ermöglichen es, Ressourcen effizienter einzuplanen und Engpässe frühzeitig zu vermeiden.

Forecasting erlaubt so die Antizipation von Bedarfen und Risiken - ist jedoch immer von der Qualität der zugrundeliegenden Daten abhängig.

Unerwartete Ereignisse (z. B. Pandemien, politische Krisen) können die Genauigkeit solcher Vorhersagen erheblich beeinträchtigen.

6. Metadaten: Wie gut ist der Output?

Neben den inhaltlichen Ergebnissen liefern viele KI-Systeme sogenannte Metadaten - also Informationen über die Güte ihrer eigenen Entscheidungen.

Zu den wichtigsten gehören Accuracy, Precision und Recall.

Accuracy

Wie viele Vorhersagen des Systems waren insgesamt korrekt?

Beispiel: Von 100 Anträgen erkennt ein System 70 korrekt →
Accuracy = 70%.

Precision

Wie viele der als „positiv“ eingestuften Fälle waren tatsächlich positiv?

Beispiel: 50 Anträge wurden als „dringend“ markiert, aber nur 40 waren es tatsächlich → Precision = $40/50 = 80\%$.

Recall

Wie viele der tatsächlich positiven Fälle wurden erkannt?

Beispiel: Es gab 60 wirklich dringende Anträge, 40 davon wurden richtig erkannt → Recall = $40/60 = 66,7\%$.

Bedeutung in der Praxis

Eine Organisation, die Anträge nach Dringlichkeit sortiert, sollte dann auf einen hohen **Recall** achten, wenn das Übersehen eines Falls (*False Negative*) gravierende Folgen hätte.

Beispiel: Ein Sozialamt prüft Notfallhilfen für obdachlose Personen. Wenn das System einen wirklich dringenden Antrag übersieht, erhält jemand in akuter Not keine schnelle Hilfe. Deshalb ist es wichtiger, möglichst alle echten Notfälle zu erkennen, auch wenn einige weniger dringende Anträge fälschlicherweise als dringend markiert werden.

Wenn hingegen Falschalarme (*False Positives*) problematisch sind – etwa weil sie Ressourcen binden – ist eine **hohe Precision** wichtiger.

Beispiel: Dieselbe Organisation hat nur begrenzte Notfallbetten.

Wenn zu viele nicht dringende Fälle fälschlich als dringend markiert werden, könnten echte Notfälle keinen Platz bekommen. Hier ist es entscheidend, dass fast alle als dringend eingestuften Fälle tatsächlich dringend sind.

Reflexionsfragen

Fragen, die man sich im Rahmen des Outputs stellen könnte:

1

Welche Form von Output produziert das KI-System, mit dem Sie arbeiten (z.B. Textklassifikation, Score, Simulation)?

2

Wie könnte die Darstellung der Ergebnisse verbessert werden, um sie für die Zielgruppe verständlicher oder nützlicher zu machen?

3

Wie stark würden Sie sich auf die Ergebnisse verlassen, wenn das System zusätzlich seine Accuracy oder Confidence mitliefert?

Fazit

Das Verständnis verschiedener Output-Formen ist entscheidend, um KI-Systeme sinnvoll in gemeinwohlorientierten Kontexten zu nutzen. Ob kategoriale Zuordnung, numerische Prädiktion, Forecasting oder Simulation - der Output ist immer nur so gut wie seine Interpretation. Die Herausforderung liegt darin, Ergebnisse nicht als absolute Wahrheiten, sondern als Hilfsmittel zur Entscheidungsunterstützung zu begreifen. Nur dann kann KI in gemeinwohlorientierten Organisationen das leisten, was sie soll: Prozesse verbessern, ohne Verantwortung zu ersetzen.

Output - Integrierte Informationsverar- beitung

Kursübersicht > KI-Technologien verstehen

Outputs von KI-Systemen sind die Ergebnisse, die ein KI-Modell produziert, nachdem es Eingabedaten verarbeitet hat. Beispiele: Vorhersagen, Klassifikationen, Wahrscheinlichkeiten, Texte, Bilder etc.

Kurz: Was das Modell am Ende „ausspuckt“.

Was ist bei Interaktion mit KI-Systemen im Bezug auf den Output relevant?

Wenn Nutzer:innen mit einem KI-System interagieren, möchten sie oft verstehen:

- **Warum** hat die KI diese Entscheidung getroffen? → *Local Feature Relevance*
- **Wie sicher** ist sich die KI in ihrer Antwort? → *Confidence Estimation*
- **Was könnte anders sein**, damit das Ergebnis anders ausfällt? → *Counterfactual Explanation*

1. Local Feature Relevance (Lokale Merkmalsbedeutung)

Hier wird **erklärt, welche Eingabemerkmale für eine einzelne spezifische Vorhersage wichtig waren.**

Beispiel bei einer medizinischen Diagnose-KI:

„Für diese eine Vorhersage waren Alter und Blutdruck besonders einflussreich, Geschlecht dagegen weniger.“

Kurz: Welche Faktoren haben in genau diesem Fall das Ergebnis bestimmt?

Typische Anwendung im Interface: Heatmaps, Balkendiagramme, Tooltipps „Dieses Merkmal hatte den größten Einfluss“

Beispiel mit Diabetes-Diagnose

Eine Ärztin gibt die Patientendaten in die KI ein. Das Modell gibt eine Vorhersage aus: „**Der Patient hat ein hohes Risiko, in den nächsten Jahren Diabetes zu entwickeln.**“

Die **Local Feature Relevance** zeigt für **diesen einen Patienten**:

Merkmal	Einfluss auf die KI-Vorhersage
Langfristiger Blutzuckerwert (HbA1c)	sehr Hoch
BMI (Übergewicht)	hoch
Alter	moderat
Bewegung pro Woche	gering
Geschlecht	kaum Einfluss

Interpretation: Die Ärztin kann jetzt sehen, dass der hohe HbA1c-Wert und das Übergewicht die Hauptgründe für die hohe Risiko-Vorhersage sind. Das bedeutet nicht, dass die KI „hohe Wahrscheinlichkeit für hohe Wahrscheinlichkeit“ ausgibt, sondern dass die KI das **konkrete Risiko für diesen Patienten** als hoch einschätzt und erklärt, warum.

2. Confidence Estimation (Konfidenzschätzung)

Das ist die **Einschätzung des Modells, wie sicher es sich bei seiner eigenen Vorhersage ist**. Meist wird dies als Wahrscheinlichkeit oder Score ausgegeben.

Wichtig: Hohe Confidence bedeutet nicht automatisch, dass die Vorhersage korrekt ist - nur, dass das Modell „glaubt“, dass sie korrekt ist.

Kurz: Wie sicher ist das Modell bei seiner Antwort?

Typische Anwendung im Interface: Wahrscheinlichkeitsanzeigen, Farbcodierung (z.B. grün = sicher, rot = unsicher)

Beispiel mit Diabetes-Diagnose

Nach der Risiko-Prognose zeigt das System:

Diabetes-Risiko: 78% Wahrscheinlichkeit

Vertrauensniveau des Modells: 92%

Die Ärztin erkennt dadurch:

- Die KI ist **sehr sicher**, obwohl die Entscheidung komplex ist.
- Sie kann die Informationen vorsichtig weitergeben: „Die Daten deuten stark auf ein erhöhtes Risiko hin.“
- Wäre die **Confidence z.B. nur 40%**, wäre Vorsicht angesagt. Eventuell müssten weitere Tests gemacht werden.

3. Counterfactual Explanation (Was-wäre-wenn-Erklärung)

Eine counterfaktische Erklärung zeigt, **welche minimale Veränderung an der Eingabe nötig wäre, damit das Modell zu einem anderen Ergebnis kommt.**

Beispiel:

„Die Kreditbewerbung wurde abgelehnt. **Hätte ihr Einkommen 400€ höher gelegen, wäre sie angenommen worden.**“

Kurz: Welche kleine Änderung hätte das Ergebnis verändert?

Typische Anwendung im Interface:

Interaktive „Was-wäre-wenn“-Slider: Nutzer:innen können Werte verändern und sieht direkt, wie sich das Ergebnis ändert.

Beispiel mit Diabetes-Diagnose

Die KI gibt zusätzlich eine counterfaktische Empfehlung:

„ **Wenn der BMI um 2 Punkte reduziert wird oder der Patient 2 zusätzliche Sporneinheiten pro Woche durchführt, sinkt das Diabetes-Risiko von 78% auf 45%.** “

Im Interface kann die Ärztin z.B. einen Schieberegler bewegen:

- BMI von 31 → 29 → Risiko sinkt sichtbar
- Bewegung 1h/Woche → 3h/Woche → Risiko sinkt weiter

Diese Erklärung ist **handlungsorientiert**, weil sie nicht nur zeigt, was das Risiko ist, sondern auch, was man konkret tun könnte.

UX-Beispiel: Kreditbewilligung

Stellen Sie sich ein Dashboard für Kreditentscheidungen vor:

1

Local Feature Relevance → Balken zeigt: „Einkommen +400€, Schulden -100€ → stärkster Einfluss auf Entscheidung“

2

Confidence Estimation → Ampel/Prozentangabe: „KI ist zu 85% sicher, dass der Antrag abgelehnt wird“

3

Counterfactual Explanation → Interaktives Widget: „Wenn Sie ihr Einkommen um 400€ erhöhen oder Schulden reduzieren, würde der Antrag genehmigt werden“

Resultat für UX:

- Nutzer:innen verstehen die Entscheidung (Local Feature Relevance)
- Nutzer:innen wissen, wie zuverlässig die KI ist (Confidence Estimation)
- Nutzer:innen können potenzielle Maßnahmen ausprobieren (Counterfactual Explanation)

Fazit

Damit Nutzer:innen die Ergebnisse eines KI-Systems verstehen und ihnen vertrauen können, reicht der reine Output (z. B. ein Risiko-Wert oder eine Entscheidung) nicht aus. Erst durch ergänzende Erklärungsmechanismen wird der Output **interpretierbar und nutzbar**.

- **Local Feature Relevance** zeigt, **warum** das Modell zu genau diesem Ergebnis kam.
- **Confidence Estimation** macht transparent, **wie sicher** das Modell in seiner Vorhersage ist.
- **Counterfactual Explanation** eröffnet **konkrete Handlungsmöglichkeiten**, indem sie zeigt, wie das Ergebnis durch kleine Änderungen beeinflusst werden kann.

In der UX führt die Kombination dieser drei Aspekte zu **Vertrauen, Nachvollziehbarkeit und Kontrolle**: zentrale Voraussetzungen für verantwortungsbewusste und gebrauchstaugliche KI-Systeme.

08

LLMs

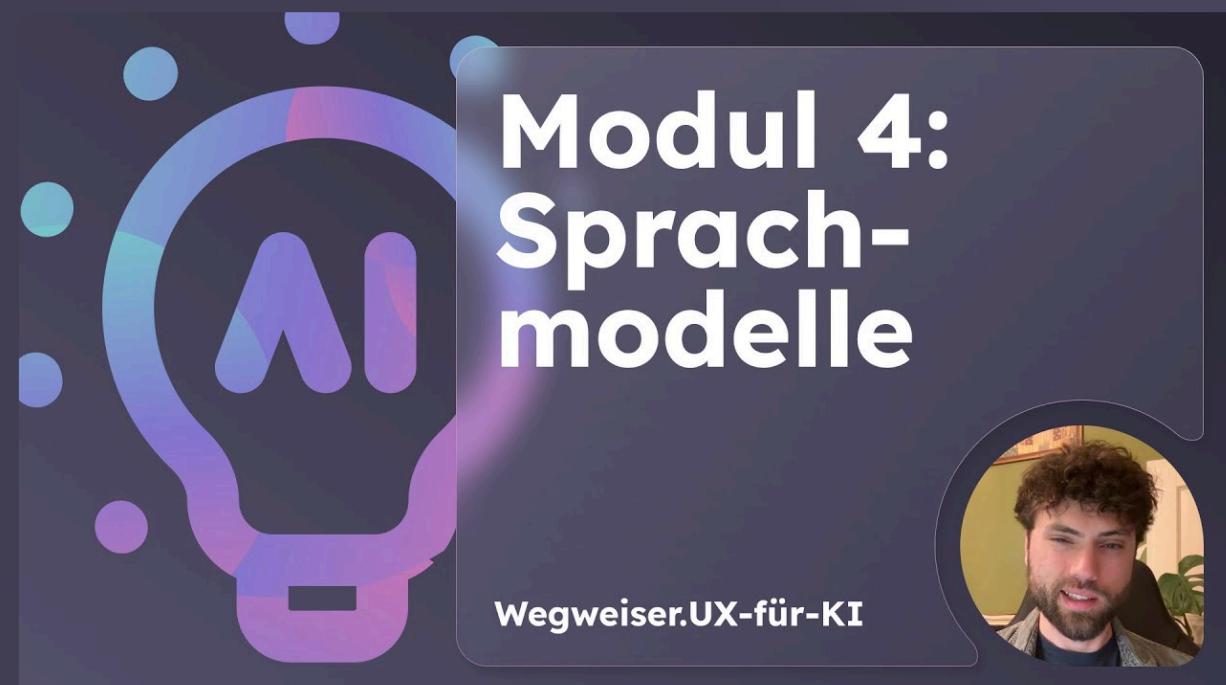
Kursübersicht > KI-Technologien verstehen

LLMs sind große Sprachmodelle, die vorhersagen, wie ein Text fortgesetzt werden könnte, basierend auf einer riesigen Textmenge.

1. Einführung: Was sind Large Language Models (LLMs)?

Large Language Models (LLMs) sind KI-Systeme, die auf der Grundlage großer Textmengen trainiert werden, um Sprache zu verstehen, zu verarbeiten und selbstständig Texte zu erzeugen. Sie bilden die Grundlage vieler moderner Anwendungen wie Chatbots, automatische Übersetzungen oder Textanalysen.

Das folgende Video gibt einen kurzen Überblick über die Funktionsweise von LLMs sowie typische Anwendungsszenarien und deren Grenzen. Diese Aspekte werden in den nächsten Abschnitten vertieft.



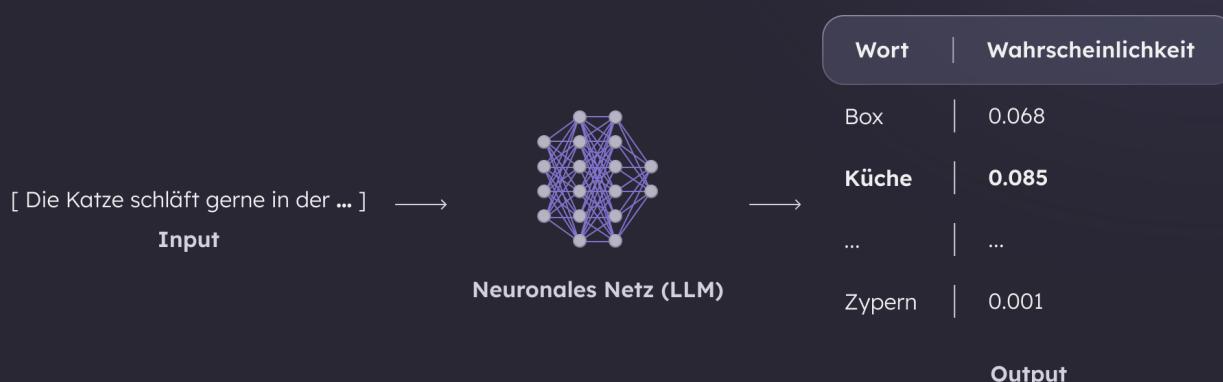
2. Funktionsweise von Large Language Models (LLMs)

Ein **Large Language Model (LLM)** lernt, Sprache zu verstehen und selbstständig Texte zu erzeugen. Im Kern basiert es auf einer einfachen, aber wirkungsvollen Idee:

Es wird darauf trainiert: **das nächste Wort in einem Satz vorherzusagen.**

Zum Beispiel:

[Die Katze schläft gerne in der ...] → Welches Wort kommt als nächstes?



Dazu benötigt das Modell eine enorme Menge an Textdaten - etwa aus dem Internet, aus Büchern, Artikeln oder anderen Quellen. Diese Daten müssen **nicht manuell beschriftet** werden, da das Modell selbst lernt, Sprachmuster zu erkennen.

Im Training entdeckt das LLM typische **Muster und Strukturen** der Sprache, beispielsweise:

- Welche Wörter häufig zusammen vorkommen
- Wie unterschiedliche Sätze aufgebaut sind
- Welche Bedeutungen und Zusammenhänge zwischen Wörtern bestehen

Sobald das Modell trainiert ist, kann es **neue Texte generieren**, Wort für Wort - oder genauer gesagt, **Token für Token**. Aber was genau ist ein **Token**?

Tokens - die Bausteine der Sprache

Ein **Token** ist eine kleine Einheit von Sprache, die das Modell verarbeitet. Das kann ein ganzes Wort, ein Wortteil oder sogar ein Satzzeichen sein.

Beispiele:

- Das Wort „*Haus*“ ist ein Token
- Das Wort „*Häuserbau*“ könnte in zwei Tokens zerlegt werden: „*Häuser*“ und „*bau*“

LLMs verarbeiten Sprache anders als Menschen also **nicht in ganzen Sätzen oder Silben**, sondern in diesen kleineren Einheiten. Durch diese Tokenisierung wird die Sprache sehr flexibel und kann detaillierter verwendet werden.

Kreativität und Variation bei der Texterzeugung

Wenn das LLM ein neues Wort (oder Token) vorhersagt, wählt es nicht immer die **wahrscheinlichste** Option. Stattdessen kann es aus mehreren **guten Möglichkeiten** auswählen. Das sorgt für **abwechslungsreiche und kreative Texte**. Deshalb können zwei Antworten auf dieselbe Frage leicht unterschiedlich ausfallen.

Kann KI verstehen? Das Gedankenexperiment des Chinesischen Zimmers

Ein bekanntes Gedankenexperiment, das hilft, die Grenzen von LLMs zu verstehen, ist das „**Chinesische Zimmer**“ des Philosophen **John Searle** (1980).

Stellen Sie sich vor, eine Person sitzt in einem geschlossenen Raum. Sie versteht **kein Chinesisch**, hat aber ein Handbuch mit **Regeln**, wie sie auf chinesische Zeichen richtig mit anderen Zeichen reagieren kann. Durch diese Regeln kann sie auf Fragen in chinesischer Schrift **korrekte Antworten** geben. So könnten Außenstehende denken, dass die Person **Chinesisch** versteht.

Tatsächlich folgt sie aber nur **formalen Anweisungen**, ohne den **Bedeutungsinhalt** der Sprache zu begreifen.

Searle nutzte dieses Gedankenexperiment, um zu zeigen:

Auch wenn ein Computer (oder ein LLM) scheinbar intelligente Antworten gibt, bedeutet das **nicht**, dass er **wirklich versteht**, was er sagt. Das Modell verarbeitet nur **Symbole nach Regeln**, ähnlich wie die Person im chinesischen Zimmer.

Bezug zu LLMs

LLMs funktionieren ganz ähnlich: Sie erkennen Muster in Sprache und erzeugen darauf basierend plausibel klingende Texte.

Doch sie **verstehen keine Bedeutungen im menschlichen Sinn**. Sie haben **kein Bewusstsein, keine Absichten und keine eigenen Gedanken**.

Das „Chinesische Zimmer“ regt dazu an, über menschliches und maschinelles Verstehen nachzudenken. Searle stellt die Idee in den Raum, dass ein System zwar auf sprachliche Eingaben sinnvoll reagieren kann, aber ohne dabei tatsächlich zu *verstehen*, was es sagt. Im Kontext moderner LLMs wird diese Frage erneut relevant: Wenn ein Modell Texte analysiert und Antworten generiert, zeigt es dann Intelligenz und Verstehen oder lediglich die Fähigkeit, sprachliche Muster zu erkennen und zu reproduzieren?

Der Chinese Room fordert uns also heraus, die Grenze zwischen echter Erkenntnis und bloßer Symbolverarbeitung von Systemen kritisch zu hinterfragen.

Reinforcement Learning from Human Feedback (RLHF)

Nach dem Grundtraining wird das Modell oft noch durch ein Verfahren namens **Reinforcement Learning from Human Feedback (RLHF)** bzw. bestärkendes Lernen verfeinert.

Dabei bewerten Menschen die Antworten des Modells, zum Beispiel danach,

- wie hilfreich,
- verständlich
- oder angemessen eine Antwort ist.

Das System nutzt Rückmeldungen, um sein Verhalten anzupassen und Vorhersagen zu verbessern. Dadurch entstehen Antworten, die **sprachlich flüssiger und konsistenter** wirken. Alle modernen LLMs wie ChatGPT oder Claude haben diese Form von menschlicher Feinjustierung durchlaufen, bevor sie auf den Markt gekommen sind. Wir merken also, ganz ohne den Menschen geht es nicht.

3. Grenzen von LLMs

So leistungsfähig Large Language Models auch sind, besteht das Risiko, dass ihre Texte **sprachlich überzeugend und intelligent wirken**, aber **inhaltlich nicht korrekt** sind. LLMs kennen keine absolute Wahrheit; sie erzeugen Inhalte ausschließlich auf Basis von **statistischen Wahrscheinlichkeiten**, die aus Trainingsdaten und Bewertungen abgeleitet werden, und stoßen dabei an folgende **Grenzen und Herausforderungen**:

1. Halluzinationen

LLMs können falsche oder frei erfundene Informationen liefern, die **plausibel klingen**, aber **nicht stimmen**. Das passiert, weil sie keine Fakten prüfen, sondern nur wahrscheinlich klingende Texte erzeugen.

2. Fehlendes Verständnis

LLMs „verstehen“ Inhalte nicht im menschlichen Sinn. Sie wissen nicht, was Wörter *bedeuten*, sondern nur, wie sie typischerweise zusammen vorkommen.

3. Veraltetes Wissen

Wenn ein Modell nicht regelmäßig aktualisiert wird, kennt es keine **aktuellen Ereignisse** oder **neuen Daten** nach dem Zeitpunkt seines Trainings.

4. Bias (Voreingenommenheit)

Da LLMs auf menschlichen Texten trainiert werden, übernehmen sie auch **gesellschaftliche Vorurteile** oder **einseitige Darstellungen**, die in den Daten vorkommen.

5. Datenschutz und Urheberrecht

In Trainingsdaten können geschützte Inhalte enthalten sein, was rechtliche und ethische Fragen aufwirft. Gerade Bild generierende Systeme sehen sich häufig mit Vorwürfen von Urheberrechtsverletzung konfrontiert.

4. LLMs im Vergleich

Es gibt heute mehrere große **Large Language Models**, die von verschiedenen Unternehmen entwickelt wurden. Sie basieren alle auf ähnlichen Prinzipien, unterscheiden sich aber beispielsweise in **Größe, Trainingsdaten, Zugänglichkeit, Fähigkeiten und Zielrichtung**.

Beispiele bekannter LLMs

Modell & Anbieter	Besonderheit / Fokus	Einsatz & Nutzen	Lizenz / Offenheit
GPT-4 / GPT-4o von OpenAI	Sehr leistungsfähig, vielseitig (Text, Code, Analyse, Konversation)	Schreiben, Programmieren, Wissensarbeit, Chatbots	Proprietär (Cloud-basiert)
Gemini von Google DeepMind	Multimodal (Text, Bild, Code, Video), eng mit Google-Ecosystem verknüpft	Multimodale Anwendungen, Such- und Wissensintegration	Proprietär
Claude von Anthropic	Fokus auf Sicherheit, Ethik, transparente KI-Antworten	Sichere, erklärbare KI-Nutzung in sensiblen Bereichen	Proprietär
Llama von Meta AI	Offen zugänglich, stark für Forschung & Fine-Tuning	Eigene Anpassungen, Forschung, interne Nutzung	Teilweise offen (Open-Weight, Lizenzbeschränkungen)
Mistral / Mixtral von Mistral AI (EU)	Europäischer Fokus auf Effizienz, Datenschutz, Open-Source-Ansatz	On-Premises-Lösungen, datenschutzsensible Anwendungen	Offen (Apache 2.0 / Open-Weight)

Offene Modelle (Open Source)

Offene oder „open-weight“ Modelle (z.B. **Llama**, **Mistral**, **Zephyr**) gewinnen stark an Bedeutung. Dabei handelt es sich nicht unbedingt um echte Open-Source-Modelle, denn dafür müssten auch die Trainingsdaten und der gesamte Trainingsprozess offenliegen. Open-Weight-Modelle machen lediglich die Modellgewichte öffentlich, werden aber häufig trotzdem als Open Source bezeichnet. Diese Modelle ermöglichen:

- **Datenhoheit & Datenschutz** (lokaler Betrieb, keine Cloudpflicht)
- **Anpassbarkeit** (Fine-Tuning, eigene Trainingsdaten)
- **Kostenkontrolle & Unabhängigkeit** von US-Plattformen

Aber: eigener Betrieb erfordert **technisches Know-how, Rechenressourcen und Wartung.**

Checkliste: Welches LLM passt zu meinem Projekt?

1. Trainingsdaten: Welche Art von Daten wurde verwendet?

- Offene Internetdaten → breite Allgemeinbildung, viele Sprachstile
- Lizenzierter / kuratierte Daten → verlässlicher, präziser, kontrollierter Inhalt

Beispiel: GPT-4 trainiert auf einer Mischung aus Webdaten, Büchern und Artikeln → gut für gezielte generelle Textgenerierung.

2. Zugänglichkeit: Wie leicht lässt sich das Modell nutzen?

- Kommerziell (z.B. GPT, Claude) → einfach via API nutzbar, Support vorhanden
- Open Source (z.B. Llama, Mistral) → volle Kontrolle, Anpassung möglich, keine Lizenzkosten

Beispiel: Llama 3 kann lokal eingesetzt werden → ideal für Projekte mit Datenschutzanforderungen.

3. Fähigkeiten / Modality: Welche Art von Daten soll verarbeitet werden?

- Text → alle klassischen LLMs
- Multimodal (Text, Bild, Audio) → Gemini, GPT-4 multimodal

Beispiel: Für ein Projekt, das Bildbeschreibungen generieren soll → GPT-4 multimodal oder Gemini.

4. Ziele / Schwerpunkt: Was soll das Modell erreichen?

- Breiter Einsatz / kreative Texte → GPT, Claude
- Effizienz, Datenschutz, leichte Integration → Mistral, Llama

Beispiel: Ein datenschutzfreundlicher interner Chatbot → Mistral oder Llama sind besser geeignet als kommerzielle APIs.

5. Weitere Kriterien (optional)

- **Kosten:** Open-Source oft günstiger, kommerzielle Modell oft nutzungsbasiert
- **Community / Support:** Größere Modelle wie GPT haben mehr Dokumentation und Nutzerfeedback
- **Anpassbarkeit:** Open-Source Modelle können feinjustiert oder in eigene Pipelines integriert werden

5. Fazit

1

Grundprinzip: LLMs lernen, das **nächste Wort vorherzusagen**, indem sie auf riesigen Textmengen Muster erkennen.

2

Tokens: Sie arbeiten mit **Tokens**, den kleinsten Einheiten der Sprache, was ihnen ermöglicht, flexibel und detailliert Texte zu erzeugen.

3

Kreativität und Variation: Durch geschickte Auswahlmechanismen entstehen **abwechslungsreiche und kreative Texte**, selbst bei identischen Eingaben.

4

Feinabstimmung mit RLHF: Reinforcement Learning from Human Feedback verbessert die Antworten durch menschliches Feedback und macht sie nützlicher, verständlicher und sicherer.

5

Grenzen: LLMs **verstehen Inhalte nicht wirklich**, können **Halluzinationen erzeugen** und sind anfällig für Bias. Sie ersetzen menschliches Urteilsvermögen nicht, sondern unterstützen es.

6

Praxisbezug: LLMs sind mächtige Werkzeuge für Textgenerierung, Analyse, Übersetzung und kreative Aufgaben - ihr Potential entfaltet sich besonders, wenn **menschliche Kontrolle und kritische Prüfung** einbezogen werden.

Kurz gesagt: LLMs sind beeindruckende Sprachwerkzeuge, aber **keine denkenden Wesen**. Sie kombinieren **mathematische Mustererkennung** mit menschlicher Anleitung, um Texte zu erzeugen, die sinnvoll, nützlich und kreativ wirken.

09 Quellen

Kursübersicht > [KI-Technologien verstehen](#)

Literaturverzeichnis

- Doshi-Velez, F., & Kim, B. (2017). *Towards a rigorous science of interpretable machine learning*. <https://arxiv.org/abs/1702.08608>
- eGov-Campus. (2021). *KI in öffentlichen verwaltungen*.
https://learn.egov-campus.org/courses/kiverwaltung_uzl_2021-1/overview
- Molnar, C. (2025). *Interpretable machine learning: a guide for making black box models explainable* (Third edition). Christoph Molnar.
- Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). „Why should i trust you?“ Explaining the predictions of any classifier. *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 1135–1144.
- Roscher, R., Bohn, B., Duarte, M. F., & Garcke, J. (2020). Explainable machine learning for scientific insights and discoveries. *IEEE access: practical innovations, open solutions*, 8, 42200–42216.

- Schrills, T. P. P. (2025). *Integrating humans and artificial intelligence in diagnostic tasks: Automation-related user experience & interaction in explainable AI / Integration von Mensch und Künstlicher Intelligenz bei diagnostischen Aufgaben: Automatisierungsbezogene User Experience & Interaktion in erklärbarer KI* [Doctoral dissertation, Universität zu Lübeck]. https://epub.uni-luebeck.de/handle/zhb_hl/3417
- Shapley, L. S. & others. (1953). *A value for n-person games*.
- University of Helsinki & MinnaLearn. (2018). *Elements of AI*.
<https://www.elementsofai.com/>