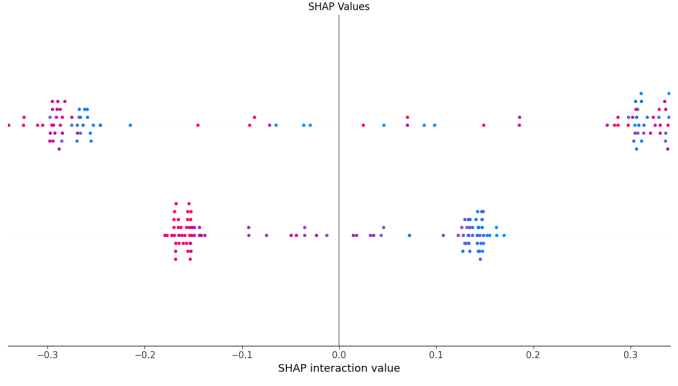


Explainable Artificial Intelligence

Felix Neubürger

2025

Fachhochschule Südwestfalen, Ingenieurs- & Wirtschaftswissenschaften

Abfrage Erwartungen und Vorwissen

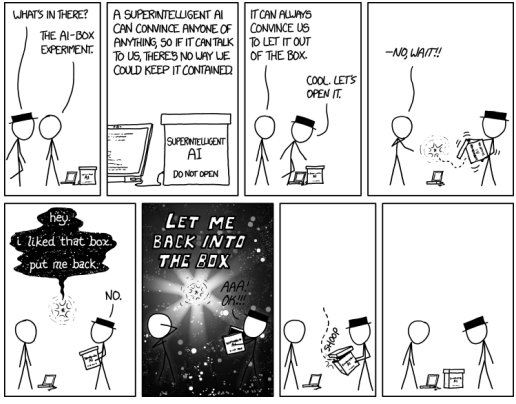
- Was verstehen Sie unter Explainable AI?
- Haben Sie bereits Erfahrungen mit maschinellem Lernen oder KI?
- Welche Erwartungen haben Sie an diese Vorlesung?
- Welche Anwendungsbereiche von KI interessieren Sie besonders?
- Gibt es spezifische Fragen, die Sie in dieser Vorlesung beantwortet haben möchten?

Inhalte der Vorlesung

- Begriffsklärungen
- Erkenntnistheoretischer Exkurs
- Methoden der Explainable AI
- Quantitative Methoden
- Anwendung der gelernten Methoden in einem Beispiel

Ziele der Vorlesung - Welche Fragen sollen beantwortet werden?

- Wofür Explainable AI?
- Was bedeutet Explainable AI?
- Interpretable AI?
- Trustworthy AI?
- Wie funktioniert das mathematisch?
- Wie schaffe ich Transparenz für Stakeholder?



[https://xkcd.com/1450/]

Format der Vorlesung - Wie sollen diese Fragen beantwortet werden?

- Theroretischer Teil mit Folien
- Selbststudium mt einem Lehrbuch[a]
- Praktischer Teil in Gruppen an einem Projekt
- Gruppengröße 2 oder 3 Personen
- Einzelarbeit möglich wenn eigenes Thema vorhanden
- Abgabe der Ausarbeitung einen Tag vor der Veranstaltung in der Blockwoche
- Vorstellung der Projektergebnisse in der Blockwoche
- Gewichtung der Bewertung Projektausarbeitung (50%) und Vortrag (50%)

[a] https://christophm.github.io/interpretable-ml-book/

Künstliche Intelligenz (KI)

- Teilgebiet der Informatik
- Automatisierung intelligenten Verhaltens
- Maschinelles Lernen als Unterbereich

Maschinelles Lernen (ML)

- Unterbereich der KI
- Algorithmen lernen aus Daten
- Treffen von Vorhersagen oder Entscheidungen
- Deep Learning basiert auf künstlichen neuronalen Netzen

Explainable Artificial Intelligence (XAI)

- Ansätze zur Verständlichkeit von KI-Entscheidungen
- Wichtig für Vertrauen und Transparenz
- Beispiel: Erklärungen in der medizinischen Diagnostik

Definitionen und Unterschiede

Interpretierbarkeit

- Fähigkeit, die internen Mechanismen eines Modells zu verstehen.
- Ermöglicht direkte Einsicht in die Funktionsweise des Modells.
- Beispiel: Lineare Regression, Entscheidungsbäume.

Erklärbarkeit

- Fähigkeit, die Entscheidungen oder Vorhersagen eines Modells verständlich zu machen.
- Oft durch zusätzliche Methoden bei komplexen Modellen erreicht.
- Beispiel: Neuronale Netze mit Post-hoc-Erklärungen.

Erkenntnistheoretische Aspekte

- **Wissenserwerb:** Wie tragen Interpretierbarkeit und Erklärbarkeit zum Verständnis von KI-Entscheidungen bei?
- **Vertrauen:** Inwiefern beeinflusst die Nachvollziehbarkeit von Modellen das Vertrauen der Nutzer?
- **Transparenz vs. Komplexität:** Balance zwischen detaillierter Einsicht und praktischer Anwendbarkeit.
- **Ethische Verantwortung:** Bedeutung von Erklärbarkeit für ethisch vertretbare KI-Systeme.

Bedeutung der Interpretierbarkeit im Maschinellen Lernen

- **Definition:**
  Interpretierbarkeit bezeichnet das Maß, in dem ein Mensch die Ursache einer Entscheidung eines Modells nachvollziehen kann.

- **Warum ist Interpretierbarkeit wichtig?**
  - **Vertrauensbildung:**
    Nutzer vertrauen eher Modellen, deren Entscheidungswege sie verstehen.
  - **Fehleranalyse:**
    Verständnis für Modellentscheidungen erleichtert das Erkennen und Beheben von Fehlern.
  - **Einhaltung gesetzlicher Vorgaben:**
    In sensiblen Bereichen wie Medizin oder Finanzen sind nachvollziehbare Entscheidungen oft gesetzlich vorgeschrieben.

Herausforderungen und Begriffsabgrenzungen

- **Herausforderungen:**
  - Fehlende einheitliche Definition von Interpretierbarkeit erschwert Kommunikation und Forschung.
  - Kompromiss zwischen Modellkomplexität und Interpretierbarkeit oft notwendig.
- **Abgrenzung zu verwandten Begriffen:**
  - **Erklärbarkeit (Explainability):**
    Fähigkeit, interne Mechanismen eines Modells verständlich zu machen.
  - **Transparenz:**
    Ausmaß, in dem die Funktionsweise eines Modells offenliegt.
  - **Vertrauen:**
    Maß, in dem Nutzer darauf vertrauen, dass ein Modell korrekte und faire Entscheidungen trifft.

EU-Regulierung und Erklärbare Künstliche Intelligenz

Die Europäische Union hat den Artificial Intelligence Act verabschiedet,[1] der am 1. August 2024 in Kraft trat.[2]

EU AI Act: Überblick

- **Ziel:** Einführung eines risikobasierten Klassifizierungssystems für KI-Anwendungen.
- **Risikokategorien:**
  - **Unzulässiges Risiko:** Verbotene KI-Anwendungen.
  - **Hohes Risiko:** Strenge Anforderungen an Transparenz, Sicherheit und Compliance.
  - **Geringes oder minimales Risiko:** Weniger strenge oder keine spezifischen Anforderungen.

Erklärbarkeit als zentrale Anforderung

- **Transparenzpflichten:** Anbieter müssen Informationen bereitstellen, die es ermöglichen, die Funktionsweise von KI-Systemen zu verstehen.
- **Vertrauenswürdigkeit:** Erklärbare KI fördert das Vertrauen der Nutzer und erleichtert die Akzeptanz von KI-Technologien.

[1] Regulation (EU) 2024/1689 des Europäischen Parlaments und des Rates vom 13. Juni 2024 über harmonisierte Vorschriften für künstliche Intelligenz. Verfügbar unter: https://eur-lex.europa.eu/eli/reg/2024/1689/oj

[2] Pressemitteilung der Europäischen Kommission: "AI Act tritt in Kraft". Verfügbar unter: https://ec.europa.eu/commission/presscorner/detail/de/ip_24_1234

Herausforderungen des EU AI Acts für Unternehmen

- **Komplexität der Regulierung:**
  Der AI Act verfolgt einen risikobasierten Ansatz, bei dem KI-Systeme in verschiedene Risikoklassen eingeteilt werden. Unternehmen müssen ihre KI-Anwendungen entsprechend einstufen und die jeweiligen Anforderungen erfüllen.[3]
- **Standardisierung und technische Umsetzung:**
  Die Entwicklung harmonisierter Standards für Hochrisiko-KI-Systeme ist komplex und zeitaufwendig. Verzögerungen können zu Unsicherheiten bei der Implementierung führen und Innovationen hemmen.[4]
- **Vermeidung von Innovationshemmnissen:**
  Es besteht die Sorge, dass strenge Regulierungen Innovationen im Bereich der Künstlichen Intelligenz behindern könnten. Unternehmen müssen Wege finden, um sowohl den gesetzlichen Anforderungen zu entsprechen als auch ihre Innovationsfähigkeit zu bewahren.[5]
- **Wettbewerbsfähigkeit im internationalen Kontext:**
  Unternehmen in Regionen mit weniger strengen Vorschriften könnten schneller Innovationen umsetzen und dadurch Wettbewerbsvorteile erlangen. Europäische Firmen stehen vor der Herausforderung, trotz strengerer Regulierungen konkurrenzfähig zu bleiben.[6]

[3] https://www.it-schulungen.com/wir-ueber-uns/wissensblog/welche-anforderungen-stellt-der-eu-ai-act.html
[4] https://www.connect-professional.de/security/der-ai-act-chancen-nutzen-risiken-managen.332959.html
[5] https://www.dps-bs.de/blog/der-ai-act-weichenstellung-fuer-kuenstliche-intelligenz-in-europa/
[6] https://de.linkedin.com/pulse/der-eu-ai-act-chancen-und-herausforderungen-f%C3%BCr-andreas-quandt-ljxne

Methoden der Explainable AI (XAI)

Global interpretierbare Modelle:

- Lineare Regression
- Entscheidungsbäume
- Regelbasierte Modelle

Post-hoc Erklärungen:

- Lokale Methoden (z.B. LIME, SHAP)
- Visualisierungen (z.B. Feature Importance, PDPs)
- Gegenbeispiele (Counterfactual Explanations)

Surrogatmodelle:

- Vereinfachte Modelle, die komplexe Modelle approximieren

Globale Methoden

Feature Importance:

- Bewertung der Bedeutung einzelner Merkmale

Permutation Feature Importance:

- Bewertung durch Permutation der Merkmale

Partial Dependence Plots (PDP):

- Einfluss eines Merkmals auf die Vorhersage

Global Surrogates:

- Erklärbares Modell, das ein komplexes Modell nachahmt

Lokale Methoden

LIME:

- Lokale lineare Approximationen des Modells

SHAP:

- Spieltheorie-basierte Quantifizierung von Merkmalbeiträgen

Counterfactual Explanations:

- Minimale Änderungen für eine andere Vorhersage

Visualisierungen

Feature Importance:

- Balkendiagramme zur Darstellung der Merkmalsbedeutung

Partial Dependence Plots (PDP):

- Einfluss eines Merkmals auf die Vorhersage

Individual Conditional Expectation (ICE):

- Individuelle Effekte von Merkmalen für einzelne Datenpunkte

Lineare Modelle

Lineare Regression:

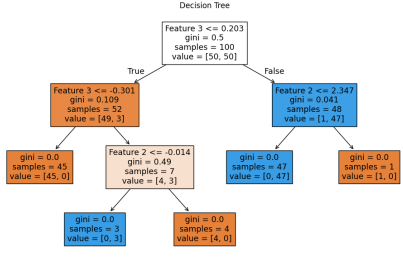
- Modell: $Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p + \epsilon$
- β_0: Achsenabschnitt
- β_i: Regressionskoeffizienten
- ϵ: Fehlerterm

Generalisierte Additive Modelle (GAMs):

- Modell: $Y = \beta_0 + f_1(X_1) + \cdots + f_p(X_p) + \epsilon$
- f_i: Glatte Funktionen für nichtlineare Beziehungen

Entscheidungsbäume

- Rekursive Partitionierung des Merkmalsraums
- Jeder Knoten: Entscheidung basierend auf Merkmal und Schwellenwert
- Ziel: Maximierung der Homogenität in den Blättern



Modellagnostische Methoden

Permutation Feature Importance:

- Permutation eines Merkmals
- Messung des Anstiegs des Vorhersagefehlers
- Signifikanter Anstieg = hohe Bedeutung

Partielle Abhängigkeitsdiagramme (PDPs):

- Zeigen durchschnittliche Wirkung eines Merkmals
- Berechnung: $\hat{f}_{X_S}(x_S) = \mathbb{E}_{X_C}[\hat{f}(x_S, X_C)]$

Akkumulierte lokale Effekte (ALE):

- Messen durchschnittliche Änderung der Vorhersage
- Berechnung: $ALE_j(x) = \int_{x_{min}}^{x} \mathbb{E}\left[\frac{\partial \hat{f}(x)}{\partial x_j} \mid x_j = z\right] dz$

Feature Permutation Importance: Einführung

- Ziel: Bewertung der Bedeutung eines Merkmals durch Permutation.
- Grundidee: Permutation eines Merkmals zerstört dessen Beziehung zur Zielvariable.
- Vorgehen:
  - Permutiere die Werte eines Merkmals.
  - Berechne den Anstieg des Vorhersagefehlers.
  - Ein signifikanter Anstieg deutet auf hohe Bedeutung hin.

Feature Permutation Importance: Mathematische Berechnung

- Gegeben:
  - Modell f
  - Testdatensatz $D = \{(x_i, y_i)\}_{i=1}^{n}$
  - Fehlerfunktion $L(y, \hat{y})$
- Berechnung:
  1. Berechne den ursprünglichen Fehler:

$$E_{\text{orig}} = \frac{1}{n} \sum_{i=1}^{n} L(y_i, f(x_i))$$

  2. Permutiere die Werte des Merkmals j: x_j^{perm}.
  3. Berechne den Fehler nach Permutation:

$$E_{\text{perm}} = \frac{1}{n} \sum_{i=1}^{n} L(y_i, f(x_i^{\text{perm}}))$$

  4. Feature Importance:

$$I_j = E_{\text{perm}} - E_{\text{orig}}$$

Partial Dependence Plots (PDP): Einführung

- Ziel: Darstellung des Einflusses eines Merkmals auf die Vorhersage.
- Grundidee: Durchschnittliche Vorhersage über alle anderen Merkmale hinweg.
- Anwendung:
  - Globale Analyse von Modellen.
  - Visualisierung nichtlinearer Beziehungen.

Partial Dependence Plots (PDP): Mathematische Berechnung

- Gegeben:
  - Modell f
  - Merkmalsmenge $X = \{X_S, X_C\}$, wobei X_S das zu analysierende Merkmal ist.
- Berechnung:

$$\hat{f}_{X_S}(x_S) = \mathbb{E}_{X_C}[f(x_S, X_C)]$$

- Diskrete Approximation:

$$\hat{f}_{X_S}(x_S) \approx \frac{1}{n} \sum_{i=1}^{n} f(x_S, x_{C,i})$$

- Visualisierung: Plot von $\hat{f}_{X_S}(x_S)$ gegen x_S.

Accumulated Local Effects (ALE): Einführung

- Ziel: Messung des durchschnittlichen Einflusses eines Merkmals auf die Vorhersage.
- Unterschied zu PDP:
  - ALE berücksichtigt die Abhängigkeiten zwischen Merkmalen.
  - ALE ist lokaler und robuster gegenüber Korrelationen.

Accumulated Local Effects (ALE): Mathematische Berechnung

- Gegeben:
  - Modell f
  - Merkmalsbereich $[x_{\min}, x_{\max}]$
- Berechnung:
  1. Berechne die lokale Wirkung:
  $$\Delta f(x_j, z) = \frac{\partial f(x)}{\partial x_j}\Big|_{x_j = z}$$
  2. Integriere über den Bereich:
  $$ALE_j(x) = \int_{x_{\min}}^{x} \mathbb{E}\left[\Delta f(x_j, z) \mid x_j = z\right] dz$$
  3. Subtrahiere den Mittelwert, um zentrierte Effekte zu erhalten.

SHAP-Werte

- Basieren auf kooperativer Spieltheorie
- Beitrag jedes Merkmals zur Vorhersage:
- $\phi_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(|N|-|S|-1)!}{|N|!} [f(S \cup \{i\}) - f(S)]$

SHAP-Werte: Einführung

- **Grundlage:**
  - Basieren auf der Spieltheorie, insbesondere den Shapley-Werten.
  - Shapley-Werte messen den durchschnittlichen marginalen Beitrag eines Spielers (Merkmals) zu einem kooperativen Spiel (Modellvorhersage).
- **Eigenschaften der Shapley-Werte:**
  - **Effizienz:** Die Summe aller Beiträge entspricht der Gesamtvorhersage.
  - **Symmetrie:** Gleiche Merkmale erhalten gleiche Beiträge.
  - **Dummy-Eigenschaft:** Merkmale ohne Einfluss haben einen Beitrag von 0.
  - **Additivität:** Beiträge aus mehreren Modellen können kombiniert werden.
- **Anwendung:**
  - Erklärbarkeit von Modellvorhersagen.
  - Identifikation der wichtigsten Merkmale.

SHAP-Werte: Berechnung (Teil 1)

- **Formel:**

$$\phi_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(|N| - |S| - 1)!}{|N|!} [f(S \cup \{i\}) - f(S)]$$

- **Erklärung der Symbole:**
  - ϕ_i: SHAP-Wert für das Merkmal i, der den Beitrag dieses Merkmals zur Modellvorhersage quantifiziert.
  - N: Gesamte Menge aller Merkmale im Modell.
  - S: Teilmenge der Merkmale aus N, die i nicht enthält ($S \subseteq N \setminus \{i\}$).

SHAP-Werte: Berechnung (Teil 2)

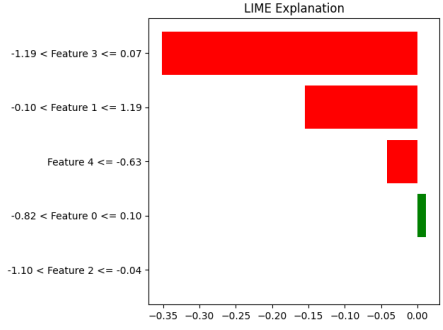
- **Erklärung der Symbole (Fortsetzung):**
  - $f(S)$: Modellvorhersage, wenn nur die Merkmale in S bekannt sind (andere Merkmale werden ignoriert).
  - $f(S \cup \{i\})$: Modellvorhersage, wenn die Merkmale in S sowie das Merkmal i bekannt sind.
  - $\frac{|S|!(|N|-|S|-1)!}{|N|!}$: Gewichtungsfaktor, der die Anzahl der möglichen Reihenfolgen berücksichtigt, in denen die Merkmale in S und i kombiniert werden können.
- **Interpretation:**
  - Der SHAP-Wert ϕ_i misst den durchschnittlichen marginalen Beitrag des Merkmals i zur Modellvorhersage über alle möglichen Teilmengen S.
  - Ein positiver ϕ_i bedeutet, dass das Merkmal i die Vorhersage erhöht, während ein negativer ϕ_i darauf hinweist, dass es die Vorhersage verringert.

SHAP-Werte: Vorteile

- **Klarheit und Fairness:**
  - SHAP-Werte bieten eine konsistente Methode zur Quantifizierung der Bedeutung von Merkmalen.
  - Sie erfüllen wichtige Eigenschaften wie Effizienz und Symmetrie, was zu fairen und nachvollziehbaren Ergebnissen führt.
- **Modellagnostik:**
  - SHAP-Werte können für beliebige Modelle verwendet werden, unabhängig von deren Komplexität.
  - Dies ermöglicht eine einheitliche Analyse über verschiedene Modelltypen hinweg.
- **Globale und lokale Interpretierbarkeit:**
  - Globale Analysen zeigen die allgemeine Bedeutung von Merkmalen.
  - Lokale Analysen erklären spezifische Vorhersagen, was Vertrauen und Transparenz fördert.
- **Spieltheoretische Grundlage:**
  - Die mathematische Basis der Shapley-Werte garantiert eine robuste und theoretisch fundierte Methode.
  - Dies stärkt die Akzeptanz in wissenschaftlichen und regulatorischen Kontexten.
- **Visualisierungsmöglichkeiten:**
  - SHAP-Werte können leicht visualisiert werden, z. B. durch Summary Plots oder Force Plots.
  - Dies erleichtert die Kommunikation der Ergebnisse an Stakeholder.

Lokale Surrogatmodelle: LIME

- Einfaches Modell (z.B. lineare Regression) lokal anpassen
- Künstliche Datenpunkte in der Nähe generieren
- Gewichtung basierend auf Ähnlichkeit zur Instanz



Lokale Surrogatmodelle: Einführung in LIME

- **Ziel:** Erklärbarkeit komplexer Modelle durch einfache, lokale Approximationen.
- **Grundidee:**
  - Ein komplexes Modell wird in der Umgebung eines spezifischen Datenpunkts durch ein einfaches Modell (z.B. lineare Regression) approximiert.
  - Die Approximation ist nur lokal gültig und erklärt die Vorhersage für diesen Datenpunkt.
- **Vorteile:**
  - Modellagnostisch: Funktioniert unabhängig vom zugrunde liegenden Modell.
  - Flexibel: Unterstützt verschiedene Arten von Daten (tabellarisch, Text, Bilder).

LIME: Beispiel für tabellarische Daten

- **Beispiel:** Ein Kreditbewertungsmodell sagt voraus, ob ein Kunde kreditwürdig ist.
- **Datenpunkt:** Ein Kunde mit folgenden Merkmalen:
  - Einkommen: 50.000 €
  - Alter: 35 Jahre
  - Schulden: 10.000 €
- **Vorhersage des Modells:** Kreditwürdig (Wahrscheinlichkeit: 85%).
- **Frage:** Warum hat das Modell diese Vorhersage getroffen?

LIME: Mathematisches Vorgehen (Schritt 1)

Schritt 1: Generierung von Nachbardatenpunkten

- Erzeuge künstliche Datenpunkte in der Umgebung des zu erklärenden Punktes.
- Beispiel: Variiere Einkommen, Alter und Schulden leicht, um ähnliche Datenpunkte zu erzeugen.
- Notation:

$$Z = \{(x'_1, f(x'_1)), (x'_2, f(x'_2)), \ldots, (x'_n, f(x'_n))\}$$

wobei x'_i ein Nachbardatenpunkt und $f(x'_i)$ die Vorhersage des komplexen Modells ist.

LIME: Mathematisches Vorgehen (Schritt 2)

Schritt 2: Gewichtung der Nachbardatenpunkte

- Ziel: Lokale Modelle sollen die Umgebung des Originalpunkts möglichst genau beschreiben.
- Nahe Datenpunkte enthalten relevantere Informationen für die lokale Struktur des Modells als weiter entfernte Punkte.
- Berechne Gewichte basierend auf der Ähnlichkeit der Nachbardatenpunkte zum Originalpunkt.
- Verwende eine Distanzfunktion $d(x, x')$ und eine Kernel-Funktion $\pi(x, x')$:

$$\pi(x, x') = \exp\left(-\frac{d(x, x')^2}{\sigma^2}\right)$$

- Beispiel: Für den Punkt **(50.000, 35, 10.000)** haben Punkte mit ähnlichem Einkommen, Alter und Schulden höhere Gewichte, da sie die lokalen Eigenschaften des Modells besser repräsentieren.

LIME: Mathematisches Vorgehen (Schritt 3)

Schritt 3: Training des lokalen Modells

- Trainiere ein einfaches Modell (z.B. lineare Regression) auf den gewichteten Nachbardatenpunkten.
- Ziel: Minimierung der gewichteten Fehlerfunktion:

$$\text{Loss}(g, \pi) = \sum_{i=1}^{n} \pi(x, x_i') \cdot \left(f(x_i') - g(x_i')\right)^2$$

  wobei g das lokale Modell ist.
- Ergebnis: Ein lineares Modell, das die Vorhersage des komplexen Modells lokal approximiert.

LIME: Mathematisches Vorgehen (Schritt 4)

Schritt 4: Interpretation der Ergebnisse

- Die Koeffizienten des lokalen Modells g geben die Bedeutung der Merkmale an.
- Beispiel: Für den Punkt $(50.000, 35, 10.000)$ könnte das lokale Modell ergeben:

$$g(x) = 0.3 \cdot \text{Einkommen} - 0.2 \cdot \text{Schulden} + 0.1 \cdot \text{Alter}$$

- Interpretation:
  - Einkommen hat den größten positiven Einfluss auf die Vorhersage.
  - Schulden haben einen negativen Einfluss.
  - Alter hat einen geringeren positiven Einfluss.

Zusammenfassung: LIME

- LIME erklärt komplexe Modelle durch einfache, lokale Approximationen.
- Schritte:
  1. Generiere Nachbardatenpunkte.
  2. Berechne Gewichte basierend auf Ähnlichkeit.
  3. Trainiere ein einfaches Modell auf den gewichteten Daten.
  4. Interpretiere die Koeffizienten des lokalen Modells.
- Vorteile:
  - Modellagnostisch und flexibel.
  - Liefert intuitive Erklärungen für spezifische Vorhersagen.
- Einschränkungen:
  - Erklärungen sind nur lokal gültig.
  - Wahl der Parameter (z.B. σ) beeinflusst die Ergebnisse.

Zusammenfassung: Modellagnostische Erklärungsmethoden

Methode	Vorteile	Nachteile
Permutation Feature Importance	Einfache Implementierung, Modellagnostisch	Kann durch Merkmalskorrelationen verzerrt werden
Partial Dependence Plots (PDP)	Globale Analyse, Visualisierung nichtlinearer Beziehungen	Ignoriert Merkmalsabhängigkeiten
Accumulated Local Effects (ALE)	Robust gegenüber Merkmalskorrelationen, Lokale Analyse	Schwieriger zu interpretieren als PDP
SHAP-Werte	Globale und lokale Interpretationen, Theoretisch fundiert	Hoher Rechenaufwand bei vielen Merkmalen
LIME	Flexibel, Modellagnostisch, Lokale Erklärungen	Erklärungen nur lokal gültig, Parameterwahl beeinflusst Ergebnisse

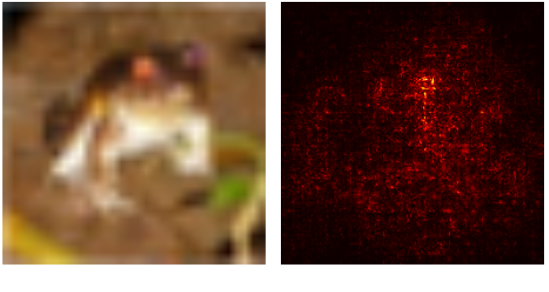
Tabelle: Zusammenfassung der modellagnostischen Erklärungsmethoden mit Vor- und Nachteilen

Modellabhängige Interpretationsmethoden: Einführung

- **Definition:**
  - Modellabhängige Methoden nutzen die spezifische Struktur und Eigenschaften eines Modells, um Erklärungen zu generieren.
  - Im Gegensatz zu modellagnostischen Methoden sind sie auf bestimmte Modelltypen zugeschnitten.
- **Vorteile:**
  - Höhere Präzision und Effizienz durch Nutzung von Modellwissen.
  - Bessere Anpassung an die spezifischen Eigenschaften des Modells.
- **Beispiele:**
  - Gradientenbasierte Methoden (z.B. Grad-CAM, Integrated Gradients).
  - Layer-wise Relevance Propagation (LRP).
  - Feature Visualization für neuronale Netze.
- **Anwendungsbereich:**
  - Besonders relevant für komplexe Modelle wie neuronale Netze, da diese oft schwer interpretierbar sind.

Methoden zur Interpretation neuronaler Netzwerke

- **Feature Visualization:**
  - Visualisierung der Merkmale, auf die Neuronen reagieren.
  - Ermöglicht Einblicke in die von Neuronen erkannten Muster.
- **Saliency Maps:**
  - Identifikation von Eingabebereichen, die den größten Einfluss auf die Ausgabe haben.
  - Darstellung der Bedeutung einzelner Pixel oder Merkmale.
- **Layer-wise Relevance Propagation (LRP):**
  - Rückverfolgung der Entscheidung des Netzwerks auf die Eingabedaten.
  - Zuweisung von Relevanzwerten zu einzelnen Eingabeelementen.



Feature Visualization: Algorithmus

- **Ziel:** Visualisierung der Merkmale, auf die ein Neuron oder eine Schicht eines neuronalen Netzwerks reagiert.
- **Vorgehen:**
  1. Wähle ein Neuron oder eine Schicht aus, die visualisiert werden soll.
  2. Optimiere ein Eingabebild, sodass die Aktivierung des Neurons maximiert wird.
  3. Verwende Regularisierungstechniken, um visuell interpretierbare Ergebnisse zu erhalten.
- **Warum Regularisierung?**
  - Ohne Regularisierung können die optimierten Eingabebilder verrauscht oder schwer interpretierbar sein.
  - Regularisierung hilft, visuell verständliche und interpretierbare Muster zu erzeugen.
- **Anwendung:**
  - Verständnis der von einem Modell erkannten Muster.
  - Debugging und Verbesserung von Modellen.

Feature Visualization: Mathematische Berechnung

- **Gegeben:**
  - Neuronale Aktivierung $A(x)$ für ein Eingabebild x.
  - Ziel: Maximierung von $A(x)$.
- **Optimierungsproblem:**

$$x^* = \arg\max_x A(x) - \lambda R(x)$$

- **Erklärung der Terme:**
  - $A(x)$: Aktivierung des Neurons für das Eingabebild x.
  - $R(x)$: Regularisierungsterm (z.B. Total Variation, L2-Norm).
  - λ: Gewichtung der Regularisierung.
- **Lösung:**
  - Gradient-basierte Optimierung (z.B. Gradient Descent).
  - Aktualisierung des Eingabebilds:

$$x \leftarrow x + \eta \frac{\partial}{\partial x} (A(x) - \lambda R(x))$$

Feature Visualization: Nutzen für die Modellinterpretation

- **Einblicke in die Funktionsweise:**
  - Zeigt, welche Muster oder Merkmale ein Neuron oder eine Schicht erkennt.
  - Hilft zu verstehen, wie das Modell Eingaben verarbeitet.
- **Debugging von Modellen:**
  - Identifiziert Neuronen, die auf irrelevante oder unerwartete Muster reagieren.
  - Unterstützt bei der Verbesserung der Modellarchitektur.
- **Erklärung von Entscheidungen:**
  - Visualisiert, welche Merkmale zu einer bestimmten Vorhersage beitragen.
  - Erhöht das Vertrauen in die Modellentscheidungen.
- **Anwendungsbeispiele:**
  - Bildklassifikation: Welche Bildbereiche sind relevant?
  - Textverarbeitung: Welche Wörter oder Phrasen sind entscheidend?

Saliency Maps: Algorithmus

- **Ziel:** Identifikation der Eingabebereiche, die den größten Einfluss auf die Modellvorhersage haben.
- **Vorgehen:**
  1. Berechne den Gradienten der Modellvorhersage $f(x)$ bezüglich der Eingabe x.
  2. Erstelle ein Bild der absoluten Gradientenwerte.
  3. Visualisiere das Bild, um die wichtigen Bereiche hervorzuheben.
- **Anwendung:**
  - Erklärbarkeit von Bildklassifikationsmodellen.
  - Debugging von Modellen.

Saliency Maps: Mathematische Berechnung

- **Gegeben:**
  - Modellvorhersage $f(x)$ für eine Eingabe x.
- **Berechnung:**

$$S(x) = \left| \frac{\partial f(x)}{\partial x} \right|$$

- **Erklärung der Terme:**
  - $\frac{\partial f(x)}{\partial x}$: Gradient der Vorhersage $f(x)$ bezüglich der Eingabe x.
  - $S(x)$: Saliency Map, die die Bedeutung jedes Pixels oder Merkmals zeigt.
- **Beispiel:**
  - Für ein Bild x mit Pixelwerten berechne den Gradienten $\frac{\partial f(x)}{\partial x}$.
  - Erstelle eine Heatmap basierend auf den absoluten Werten der Gradienten.

Layer-wise Relevance Propagation (LRP): Algorithmus

- **Ziel:** Rückverfolgung der Modellvorhersage auf die Eingabedaten, um die Relevanz jedes Eingabeelements zu bestimmen.
- **Vorgehen:**
  1. Beginne mit der Modellvorhersage $f(x)$.
  2. Verteile die Relevanz schichtweise rückwärts durch das Netzwerk.
  3. Nutze Relevanzverteilungsregeln, um die Relevanz auf die Eingabedaten zu projizieren.
- **Anwendung:**
  - Erklärbarkeit von neuronalen Netzwerken.
  - Identifikation wichtiger Eingabemerkmale.

Layer-wise Relevance Propagation (LRP): Mathematische Berechnung

- **Gegeben:**
  - Neuronale Aktivierungen z_{ij} zwischen Neuronen i und j.
  - Relevanz R_j eines Neurons j in der nächsten Schicht.
- **Startwert der Relevanz:**
  - Die Relevanz R_j in der Ausgangsschicht entspricht der Modellvorhersage $f(x)$:

$$R_j = f(x)$$

- **Relevanzverteilung:**

$$R_i = \sum_j \frac{z_{ij}}{\sum_k z_{kj}} R_j$$

- **Erklärung der Terme:**
  - z_{ij}: Beitrag des Neurons i zum Neuron j.
  - $\sum_k z_{kj}$: Gesamte Eingabe zum Neuron j.
  - R_i: Relevanz des Neurons i.
- **Beispiel:**
  - Für ein neuronales Netzwerk mit Eingabe x und Vorhersage $f(x)$ wird die Relevanz von $f(x)$ schichtweise auf die Eingabe x zurückgeführt.

Layer-wise Relevance Propagation (LRP): Interpretation einzelner Schichten

- **Ziel:** Analyse der Relevanzverteilung in jeder Schicht des neuronalen Netzwerks.
- **Vorgehen:**
  - Rückverfolgung der Relevanz von der Ausgangsschicht bis zur Eingabeschicht.
  - Untersuchung der Relevanzverteilung in jeder Zwischenschicht.
- **Nutzen:**
  - Identifikation der Schichten, die am meisten zur Vorhersage beitragen.
  - Verständnis der Informationsverarbeitung im Netzwerk.

LRP: Interpretation der Zwischenschichten

- **Relevanz in Zwischenschichten:**
  - Jede Schicht erhält Relevanzwerte basierend auf ihrer Aktivierung und ihrem Beitrag zur nächsten Schicht.
  - Relevanzverteilung zeigt, welche Neuronen in einer Schicht besonders wichtig sind.
- **Beispiel:**
  - In einer Convolutional Layer können Relevanzwerte zeigen, welche Filter die wichtigsten Merkmale extrahieren.
  - In einer Fully Connected Layer können sie die Bedeutung einzelner Neuronen quantifizieren.
- **Visualisierung:**
  - Heatmaps oder Diagramme können verwendet werden, um die Relevanz in jeder Schicht darzustellen.

LRP: Vorteile der Schichtweisen Analyse

- **Einblicke in die Modellstruktur:**
  - Verstehen, wie Informationen durch das Netzwerk fließen.
  - Identifikation von Schichten, die für spezifische Aufgaben entscheidend sind.
- **Debugging:**
  - Erkennen von Schichten, die irrelevante oder fehlerhafte Informationen verarbeiten.
  - Verbesserung der Netzwerkarchitektur basierend auf den Relevanzanalysen.
- **Erklärbarkeit:**
  - Transparenz für Stakeholder durch detaillierte Erklärungen der Modellentscheidungen.
  - Erhöhtes Vertrauen in die Vorhersagen des Modells.

Grad-CAM: Einführung

- **Ziel:** Visualisierung der relevanten Bildbereiche, die zu einer Modellvorhersage beitragen.
- **Grundidee:**
  - Grad-CAM nutzt die Gradienten der Zielklasse, um die Bedeutung der Feature-Maps einer Convolutional Layer zu bestimmen.
  - Die resultierende Heatmap zeigt, welche Bildbereiche für die Vorhersage am wichtigsten sind.
- **Anwendungsbereiche:**
  - Bildklassifikation: Identifikation relevanter Bildbereiche.
  - Medizinische Bildanalyse: Lokalisierung von Anomalien.
  - Debugging von CNNs: Überprüfung, ob das Modell auf sinnvolle Merkmale achtet.

Grad-CAM: Schritt 1 - Gradientenberechnung

- **Ziel:** Berechnung der Gradienten der Zielklasse y^c bezüglich der Feature-Maps A^k einer Convolutional Layer.
- **Formel:**

$$\frac{\partial y^c}{\partial A_{ij}^k}$$

- **Erklärung:**
  - y^c: Modellvorhersage für die Zielklasse c.
  - A^k: Feature-Map k der ausgewählten Convolutional Layer.
  - $\frac{\partial y^c}{\partial A_{ij}^k}$: Gradient der Zielklasse y^c bezüglich der Aktivierung an Position (i, j) in der Feature-Map A^k.
- **Nutzen:** Die Gradienten zeigen, wie stark die Aktivierungen in A^k die Zielklasse beeinflussen.

Grad-CAM: Schritt 2 - Gewichtung der Feature-Maps

- **Ziel:** Aggregation der Gradienten über die räumlichen Dimensionen, um die Bedeutung jeder Feature-Map zu bestimmen.
- **Formel:**

$$\alpha_k^c = \frac{1}{Z} \sum_i \sum_j \frac{\partial y^c}{\partial A_{ij}^k}$$

- **Erklärung:**
  - α_k^c: Gewicht für die Feature-Map A^k in Bezug auf die Zielklasse c.
  - Z: Anzahl der räumlichen Positionen in der Feature-Map A^k.
  - $\sum_i \sum_j$: Summe über alle räumlichen Positionen (i, j).
- **Nutzen:** Die Gewichte α_k^c geben an, wie wichtig jede Feature-Map für die Zielklasse ist.

Grad-CAM: Schritt 3 - Erzeugung der Heatmap

- **Ziel:** Erstellung der Grad-CAM Heatmap durch gewichtete Kombination der Feature-Maps.
- **Formel:**

$$L^c_{\text{Grad-CAM}} = \text{ReLU}\left(\sum_k \alpha_k^c A^k\right)$$

- **Erklärung:**
  - α_k^c: Gewicht der Feature-Map A^k für die Zielklasse c.
  - $\sum_k$: Summe über alle Feature-Maps.
  - ReLU: Aktivierungsfunktion, um negative Werte zu entfernen.
- **Nutzen:** Die resultierende Heatmap $L^c_{\text{Grad-CAM}}$ zeigt die relevanten Bildbereiche für die Zielklasse.

Grad-CAM: Zusammenfassung und Vorteile

- **Zusammenfassung:**
  1. Berechne die Gradienten der Zielklasse y^c bezüglich der Feature-Maps A^k.
  2. Aggregiere die Gradienten, um die Gewichte α_k^c zu bestimmen.
  3. Erstelle die Grad-CAM Heatmap durch gewichtete Kombination der Feature-Maps.
- **Vorteile:**
  - Intuitive Visualisierung der relevanten Bildbereiche.
  - Modellagnostisch für Convolutional Neural Networks (CNNs).
  - Unterstützt Debugging und Verbesserung von Modellen.
- **Anwendungsbeispiele:**
  - Bildklassifikation: Identifikation der wichtigsten Bildbereiche.
  - Medizinische Bildanalyse: Lokalisierung von Anomalien.

Grad-CAM: Herausforderungen

- **Herausforderungen:**
  - Bei Modellen wie RCNNs, die nicht durchgängig differenzierbar sind, können Gradientenberechnungen problematisch sein.
  - Nicht-differenzierbare Operationen (z. B. ROI-Pooling) führen zu ungenauen oder fehlenden Gradienten.
  - Mögliche Lösung: Approximationstechniken oder modifizierte Varianten von Grad-CAM.

Integrated Gradients: Einführung

- **Ziel:** Quantifizierung des Beitrags jedes Eingabemerkmals zur Modellvorhersage.
- **Grundidee:**
  - Integriere die Gradienten der Modellvorhersage entlang eines geraden Pfads von einer Baseline-Eingabe x' zur aktuellen Eingabe x.
- **Formel:**

$$\mathrm{IG}_i(x) = (x_i - x_i') \int_{\alpha=0}^{1} \frac{\partial f(x' + \alpha(x - x'))}{\partial x_i} d\alpha$$

- **Eigenschaften:**
  - **Pfadinvarianz:** Die Summe der Beiträge entspricht der Differenz der Modellvorhersagen zwischen x und x'.
  - **Baseline:** Die Wahl der Baseline (z.B. Nullvektor) beeinflusst die Ergebnisse.

Integrated Gradients: Detaillierte Berechnung

- **Gegeben:**
  - Eingabe x und Baseline x' (z. B. Nullvektor oder Durchschnittswerte).
  - Modellvorhersage $f(x)$.
- **Schritte:**
  1. **Pfaddefinition:** Definiere einen geraden Pfad von x' nach x:
  $$x(\alpha) = x' + \alpha(x - x'), \quad \alpha \in [0, 1]$$
  2. **Gradientenberechnung:** Berechne den Gradienten der Modellvorhersage entlang des Pfads:
  $$\frac{\partial f(x(\alpha))}{\partial x_i}$$
  3. **Integration:** Integriere die Gradienten entlang des Pfads:
  $$\text{IG}_i(x) = (x_i - x_i') \int_{\alpha=0}^{1} \frac{\partial f(x(\alpha))}{\partial x_i} d\alpha$$
- **Numerische Approximation:**
  - Diskretisiere den Pfad in m Schritte: $\alpha_1, \alpha_2, \ldots, \alpha_m$.
  - Approximiere das Integral durch eine Summe:
  $$\text{IG}_i(x) \approx (x_i - x_i') \cdot \frac{1}{m} \sum_{k=1}^{m} \frac{\partial f(x(\alpha_k))}{\partial x_i}$$

Integrated Gradients: Anwendungszwecke

- **Anwendungsbereiche:**
  - **Bildklassifikation:** Identifikation der Pixel, die am meisten zur Vorhersage beitragen.
  - **Textverarbeitung:** Bewertung der Bedeutung einzelner Tokens oder Wörter.
  - **Tabellarische Daten:** Analyse der Merkmalsbeiträge für spezifische Vorhersagen.
- **Vorteile:**
  - Modellagnostisch: Funktioniert für beliebige differenzierbare Modelle.
  - Robust gegenüber kleinen Änderungen in den Eingabedaten.
- **Beispiele:**
  - Medizinische Diagnostik: Identifikation relevanter Merkmale in Patientendaten.
  - Finanzwesen: Erklärung von Kreditentscheidungen.

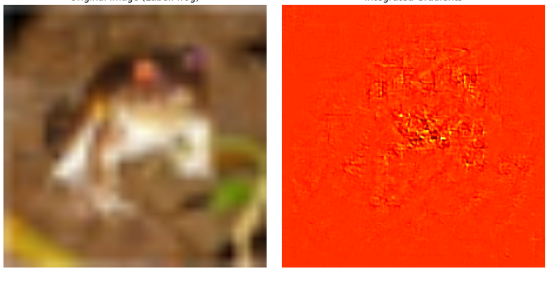
Zusammenfassung: Methoden der Explainable AI

Methode	Vorteile	Nachteile	Anwendungsfälle
Grad-CAM	Intuitive Visualisierung relevanter Bildbereiche	Funktioniert nur für CNNs, Probleme bei nicht-differenzierbaren Modellen	Bildklassifikation, Medizinische Bildanalyse
Integrated Gradients	Robust, Modellagnostisch	Wahl der Baseline beeinflusst Ergebnisse	Bilder, Text, Tabellarische Daten
Layer-wise Relevance Propagation (LRP)	Schichtweise Analyse, Rückverfolgbarkeit	Abhängig von Modellarchitektur	Neuronale Netzwerke

Tabelle: Zusammenfassung der XAI-Methoden mit Vor- und Nachteilen sowie Anwendungsfällen

Werkzeuge und Bibliotheken für XAI

- **Captum:**
  - PyTorch-Bibliothek für Interpretierbarkeitsmethoden.
  - Unterstützt Techniken wie integrierte Gradienten und DeepLIFT.
- **ELI5:**
  - Bibliothek zur Erklärung von ML-Modellen und Vorhersagen.
  - Unterstützt verschiedene Modelle wie Sklearn, XGBoost und Keras.
- **SHAP-Bibliothek:**
  - Implementierung der SHAP-Werte für verschiedene Modelltypen.
  - Ermöglicht detaillierte Analysen der Merkmalsbeiträge.



Erklärung für Sentiment-Analyse mit LLM

- **Ziel:** Generierung und Speicherung einer Erklärung für eine Sentiment-Analyse-Aufgabe.
- **Vorgehen:**
  - Verwendung eines vortrainierten Sprachmodells (LLM) aus der Hugging Face Transformers-Bibliothek.
  - Durchführung der Sentiment-Analyse auf einem Beispieltext.
  - Visualisierung der Erklärung (z.B. Token-Wichtigkeit) als Balkendiagramm.
- **Ergebnis:**
  - Die Bedeutung einzelner Tokens wird grafisch dargestellt, um die Entscheidungsfindung des Modells zu verdeutlichen.

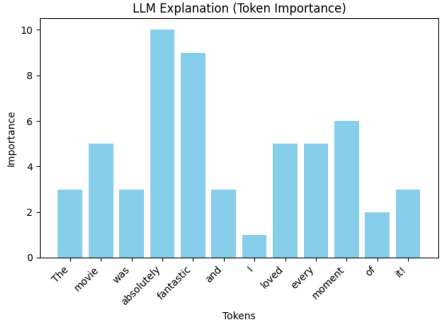


Abbildung: Visualisierung der Token-Wichtigkeit für Sentiment-Analyse

Interpretation von LLMs (Large Language Models)

- **Herausforderungen:**
  - Hohe Komplexität und Anzahl der Parameter erschweren die Nachvollziehbarkeit.
  - Entscheidungen basieren auf nichtlinearen Beziehungen zwischen Tokens.
- **Ansätze zur Interpretation:**
  - **Attention Visualisierung:**
    - Darstellung der Aufmerksamkeit (Attention Scores) zwischen Tokens.
    - Tool: BertViz.
  - **Feature Attribution:**
    - Identifikation der wichtigsten Tokens für eine Vorhersage.
    - Tools: Captum, SHAP.
  - **Neuronale Aktivierungen:**
    - Analyse der Aktivierungsmuster einzelner Neuronen.
    - Tool: Neuroscope.

Interpretation von LLMs: Attention Visualisierung

- **Ziel:** Darstellung der Aufmerksamkeit (Attention Scores) zwischen Tokens.
- **Grundidee:**
  - Transformer-Modelle wie BERT und GPT nutzen Attention-Mechanismen, um die Beziehungen zwischen Tokens zu modellieren.
  - Die Attention-Matrix zeigt, wie stark ein Token auf andere Tokens achtet.
- **Beispiel:**
  - Satz: "The cat sat on the mat."
  - Visualisierung: Zeigt, dass "cat" stark mit "sat" und "mat" verbunden ist.
- **Werkzeuge:**
  - BertViz: Interaktive Visualisierung der Attention-Matrizen.

Attention Visualisierung: Algorithmus

- **Schritt 1: Berechnung der Attention-Matrix**
  - Gegeben: Eingabe-Tokens $x_1, x_2, \ldots, x_n$.
  - Berechnung der Attention-Werte:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

  - Q, K, V: Query-, Key- und Value-Matrizen.
  - d_k: Dimension der Key-Vektoren.
- **Schritt 2: Extraktion der Attention Scores**
  - Die Attention-Matrix enthält die Scores für jedes Token-Paar.
  - Beispiel: Attention von ”cat” auf ”sat” beträgt 0.8.
- **Schritt 3: Visualisierung**
  - Erstelle eine Heatmap, um die Scores darzustellen.
  - Höhere Werte werden durch intensivere Farben hervorgehoben.

Interpretation von LLMs: Feature Attribution

- **Ziel:** Identifikation der wichtigsten Tokens für eine Vorhersage.
- **Ansatz:**
  - Quantifiziere den Beitrag jedes Tokens zur Modellvorhersage.
  - Nutze Methoden wie Integrated Gradients oder SHAP.
- **Beispiel:**
  - Satz: "The movie was absolutely fantastic."
  - Ergebnis: "fantastic" hat den höchsten Beitrag zur positiven Sentiment-Vorhersage.
- **Werkzeuge:**
  - Captum: Unterstützt Integrated Gradients.
  - SHAP: Berechnet Shapley-Werte für Tokens.

Feature Attribution: Algorithmus (Integrated Gradients)

- **Schritt 1: Definition der Baseline**
  - Wähle eine Baseline-Eingabe x' (z. B. leere Tokens).
- **Schritt 2: Berechnung der Gradienten**
  - Definiere einen Pfad von der Baseline x' zur Eingabe x:

$$x(\alpha) = x' + \alpha(x - x'), \quad \alpha \in [0, 1]$$

  - Berechne die Gradienten entlang des Pfads:

$$\frac{\partial f(x(\alpha))}{\partial x_i}$$

- **Schritt 3: Integration der Gradienten**
  - Integriere die Gradienten entlang des Pfads:

$$\mathrm{IG}_i(x) = (x_i - x_i') \int_{\alpha=0}^{1} \frac{\partial f(x(\alpha))}{\partial x_i} d\alpha$$

- **Ergebnis:** Die Werte $\mathrm{IG}_i(x)$ quantifizieren den Beitrag jedes Tokens.

Interpretation von LLMs: Neuronale Aktivierungen

- **Ziel:** Analyse der Aktivierungsmuster einzelner Neuronen.
- **Ansatz:**
  - Untersuche, wie stark ein Neuron auf verschiedene Eingaben reagiert.
  - Identifiziere spezialisierte Neuronen (z. B. für Grammatik oder Semantik).
- **Beispiel:**
  - Neuron X reagiert stark auf Adjektive wie ”beautiful” oder ”ugly”.
- **Werkzeuge:**
  - Neuroscope: Visualisiert neuronale Aktivierungen.

Neuronale Aktivierungen: Algorithmus

- **Schritt 1: Auswahl eines Neurons**
  - Wähle ein spezifisches Neuron in einer Schicht des Modells.
- **Schritt 2: Berechnung der Aktivierungen**
  - Für eine Eingabe x berechne die Aktivierung des Neurons:

$$a(x) = \text{ReLU}(W \cdot x + b)$$

  - W: Gewichtsmatrix, b: Bias, ReLU: Aktivierungsfunktion.
- **Schritt 3: Analyse der Aktivierungen**
  - Untersuche die Aktivierungen für verschiedene Eingaben.
  - Identifiziere Muster oder Eingabetypen, die starke Aktivierungen auslösen.
- **Ergebnis:** Verstehe die Rolle des Neurons im Modell.

Bibliotheken zur Interpretation von LLMs

- **BertViz:**
  - Visualisierung der Attention-Matrizen in Transformer-Modellen.
  - Unterstützt Modelle wie BERT, GPT-2.
  - `https://github.com/jessevig/bertviz`
- **Transformers Interpret:**
  - Feature-Attribution-Methoden für Hugging Face Modelle.
  - Unterstützt LIME, Integrated Gradients.
  - `https://github.com/cdpierse/transformers-interpret`
- **Captum:**
  - PyTorch-Bibliothek für Interpretierbarkeit.
  - Unterstützt Integrated Gradients, Layer Conductance.
  - `https://captum.ai/`
- **SHAP:**
  - Shapley-Werte für Feature Attribution.
  - Unterstützt Transformer-Modelle.
  - `https://github.com/slundberg/shap`

Quantitative Methoden der XAI

- **Ziel:** Bewertung der Qualität und Zuverlässigkeit von Erklärungen, die von XAI-Methoden generiert werden.
- **Quantus Python-Bibliothek:** `https://github.com/understandable-machine-intelligence-lab/Quantus`
  - Implementiert über 35 Metriken zur Evaluation von Erklärungen.
  - Unterstützt die Analyse von Erklärungen hinsichtlich:
    - **Plausibilität:** Wie gut stimmen die generierten Erklärungen mit menschlichem Verständnis überein?
    - **Robustheit:** Wie stabil sind die Erklärungen bei kleinen Änderungen der Eingabe?
    - **Fidelity:** Wie gut repräsentieren die Erklärungen das zugrunde liegende Modell?
  - Beispiele für Metriken zur Bewertung von Erklärungen:
    - **Faithfulness:** Misst, wie stark die Erklärungen mit den Modellvorhersagen korrelieren.
    - **Complexity:** Bewertet die Verständlichkeit und Einfachheit der Erklärungen.
- **Vorteile:**
  - Einheitliche Evaluationsmethoden für die Qualität von Erklärungen verschiedener XAI-Techniken.
  - Ermöglicht systematische und reproduzierbare Analysen der Erklärungen.

Quantitative XAI Metriken: Faithfulness (Teil 1)

- **Definition:** Faithfulness misst, wie gut die generierten Erklärungen die tatsächlichen Modellvorhersagen repräsentieren. Sie quantifiziert, ob die in der Erklärung hervorgehobenen Merkmale tatsächlich die Modellentscheidung beeinflussen.
- **Grundidee:** Die Bedeutung eines Merkmals wird durch dessen Entfernung aus der Eingabe überprüft. Wenn die Modellvorhersage sich stark ändert, zeigt dies, dass das entfernte Merkmal eine wichtige Rolle spielt. Die Metrik bewertet somit die Treue (Faithfulness) der Erklärung in Bezug auf das Verhalten des Modells.
- **Mathematische Formel:**

$$\text{Faithfulness} = \frac{1}{n} \sum_{i=1}^{n} \left| f(x) - f(x \ \ x_i) \right|$$

- **Erklärung der Terme:**
  - $E(x)$: Erklärung für die vollständige Eingabe x.
  - $E(x \ \ x_i)$: Erklärung, nachdem das Merkmal x_i entfernt wurde.
  - n: Anzahl der Merkmale in der Eingabe.

Quantitative XAI Metriken: Faithfulness (Teil 2)

- **Interpretation:**
  - Eine hohe Änderung der Erklärung $\|E(x) - E(x\ x_i)\|$ deutet darauf hin, dass das entfernte Merkmal x_i eine zentrale Rolle in der generierten Erklärung spielt.
  - Wenn die Änderung der Erklärung gering ist, könnte dies darauf hinweisen, dass das Merkmal in der Erklärung überbewertet wurde oder keinen signifikanten Einfluss auf die generierte Erklärung hat.
- **Bedeutung für XAI:**
  - Faithfulness ist eine zentrale Metrik, um die Qualität von Erklärungen zu bewerten.
  - Sie stellt sicher, dass die Erklärungen nicht nur plausibel erscheinen, sondern auch tatsächlich das Verhalten des Modells widerspiegeln.
  - Dies ist besonders wichtig, um Vertrauen in die Erklärungen und das Modell zu schaffen.

Quantitative XAI Metriken: Robustness

- **Definition:** Bewertet die Stabilität der Erklärungen bei kleinen Änderungen der Eingabe.
- **Grundidee:** Vergleiche Erklärungen für ähnliche Eingaben.
- **Mathematische Formel:**

$$\text{Robustness} = \frac{1}{n} \sum_{i=1}^{n} \| E(x) - E(x + \delta) \|_2$$

- **Erklärung:**
  - $E(x)$: Erklärung für die Eingabe x.
  - δ: Kleine Störung der Eingabe.
  - $\| \cdot \|_2$: Euklidische Distanz.
- **Interpretation:** Geringe Änderungen in der Erklärung zeigen eine robuste Methode.

Quantitative XAI Metriken: Plausibility

- **Definition:** Misst, wie gut die Erklärungen mit menschlichem Verständnis übereinstimmen.
- **Grundidee:** Vergleiche Erklärungen mit annotierten Daten.
- **Mathematische Formel:**

$$\text{Plausibility} = \frac{1}{n} \sum_{i=1}^{n} \text{Sim}(E(x), A(x))$$

- **Erklärung:**
  - $E(x)$: Erklärung für die Eingabe x.
  - $A(x)$: Menschliche Annotation für x.
  - $\text{Sim}(\cdot, \cdot)$: Ähnlichkeitsmaß (z. B. Cosinus-Ähnlichkeit).
- **Interpretation:** Höhere Werte zeigen eine bessere Übereinstimmung mit menschlichem Verständnis.

Quantitative XAI Metriken: Complexity

- **Definition:** Bewertet die Verständlichkeit der Erklärungen.
- **Grundidee:** Kürzere und einfachere Erklärungen sind besser verständlich.
- **Mathematische Formel:**

$$\text{Complexity} = \text{Length}(E(x))$$

- **Erklärung:**
  - $E(x)$: Erklärung für die Eingabe x.
  - $\text{Length}(\cdot)$: Länge oder Anzahl der Elemente in der Erklärung.
- **Interpretation:** Kürzere Erklärungen sind leichter zu interpretieren.

Zusammenfassung: Quantitative XAI Metriken

Metrik	Beschreibung	Ziel
Faithfulness	Misst, wie gut die Erklärung die Modellvorhersage repräsentiert	Sicherstellung der Treue der Erklärung
Robustness	Bewertet die Stabilität der Erklärung bei kleinen Eingabeänderungen	Überprüfung der Stabilität
Plausibility	Misst die Übereinstimmung der Erklärung mit menschlichem Verständnis	Validierung der Verständlichkeit
Complexity	Bewertet die Länge und Einfachheit der Erklärung	Förderung der Interpretierbarkeit

Tabelle: Zusammenfassung der wichtigsten Metriken zur Bewertung von XAI-Erklärungen

Zusammenfassung der Vorlesung

- **Wofür Explainable AI?**
  - Fördert Vertrauen und Transparenz in KI-Systemen.
  - Erleichtert die Fehleranalyse und Einhaltung gesetzlicher Vorgaben.
- **Was bedeutet Explainable AI?**
  - Ansätze zur Verständlichkeit von KI-Entscheidungen.
  - Unterschied zwischen Interpretierbarkeit (direkte Einsicht) und Erklärbarkeit (Post-hoc-Methoden).
- **Interpretable AI?**
  - Globale Modelle wie lineare Regression und Entscheidungsbäume.
  - Post-hoc-Methoden wie SHAP, LIME und Visualisierungen.
- **Trustworthy AI?**
  - Erklärbarkeit als zentrale Anforderung im EU AI Act.
  - Förderung von Akzeptanz und ethischer Verantwortung.
- **Wie funktioniert das mathematisch?**
  - SHAP-Werte basieren auf Spieltheorie.
  - Methoden wie Grad-CAM und Integrated Gradients nutzen Gradientenberechnungen.
- **Wie schaffe ich Transparenz für Stakeholder?**
  - Nutzung von Visualisierungen (z. B. Saliency Maps, PDPs).
  - Einsatz von XAI-Toolkits wie SHAP, Captum und Quantus.

Wichtige Erkenntnisse und Methoden

- **Globale Methoden:**
  - Feature Importance, Partial Dependence Plots (PDP), Global Surrogates.
- **Lokale Methoden:**
  - LIME, SHAP, Counterfactual Explanations.
- **Visualisierungen:**
  - Saliency Maps, Grad-CAM, Feature Visualization.
- **Quantitative Bewertung:**
  - Faithfulness, Robustness, Plausibility, Complexity.
- **Werkzeuge:**
  - SHAP, Captum, Quantus, AI Explainability 360.

Abschließende Gedanken

- Explainable AI ist entscheidend für die Akzeptanz und den verantwortungsvollen Einsatz von KI.
- Die Wahl der richtigen Methoden hängt vom Anwendungsfall und den Stakeholdern ab.
- Kombination aus mathematischen Grundlagen, Visualisierungen und Werkzeugen ermöglicht umfassende Erklärungen.
- Zukünftige Entwicklungen: Verbesserung der Robustheit und Plausibilität von Erklärungen.

XAI Standardwerke

- **Interpretable ML Book** (Molnar)
  - `https://christophm.github.io/interpretable-ml-book/`
  - Umfassender Leitfaden zu SHAP, LIME, etc.
- **Explainable AI** (Springer 2019)
  - DOI: 10.1007/978-3-030-28954-6
  - Forschungsbeiträge zu Deep Learning Interpretability

XAI Toolkits & Frameworks

- **Quantus**
  - `https://github.com/understandable-machine-intelligence-lab/Quantus`
  - 35+ Metriken für XAI-Evaluation
  - JMLR Paper: Quantus Paper
- **SHAP**
  - `https://github.com/slundberg/shap`
  - Shapley Values für Feature Importance
- **AI Explainability 360** (IBM)
  - `https://aix360.mybluemix.net/`
  - Enthält ProtoDash, CEM

XAI Forschungsarbeiten

- **”Explainable AI: A Review”** (Gilpin 2018)
  - `https://arxiv.org/abs/1801.00631`
  - Systematische Klassifikation
- **LIME Paper** (Ribeiro 2016)
  - `https://arxiv.org/abs/1602.04938`
  - Lokale Erklärbarkeit
- **”Sanity Checks Revisited”** (Hedström 2023)
  - Verbesserte MPRT-Metriken

References I