# Erklärung

## Autor und Herausgeber

Alexander Jung*, Department of Computer Science, Aalto University, Espoo, Finland (ORCID 0000-0001-7538-0990)
Editors: Konstantina Olioumtsevits (ORCID 0000-0003-2697-8659), Ekkehard Schnoor (ORCID 0000-0001-8252-1499), Aalto University, Espoo, Finland
*Corresponding author: alex.jung@aalto.fi

## Zusammenfassung

Eine Erklärung ist ein Artefakt, das dem Benutzer einer machine learning (ML)-Methode zusammen mit einer Vorhersage präsentiert wird, um die Transparenz zu erhöhen, wie diese Vorhersage zustande kam. Erklärungen können eine textuelle Beschreibung, ein Merkmalswichtigkeit-Wert, eine Approximation einer Hypothese-Karte oder eine Heatmap sein. Eine Erklärung von lokalem oder globalem Umfang kann als einfache Hypothese gelernt werden, die die Vorhersagen der gelernten Hypothese auf einem Probing-Set von Merkmalsvektoren reproduziert. Einerseits muss eine Erklärung treu sein: Sie muss die Berechnung widerspiegeln, durch die die gelernte Hypothese zu der Vorhersage gelangt. Andererseits muss sie verständlich sein: Ihr Benutzer muss allein daraus antizipieren können, wie sich die Vorhersage ändert, wenn sich die Merkmale ändern.

## Definition

Ein Wetterdienst prognostiziert Frost für die kommende Nacht in Krems an der Donau (Österreich) und fügt hinzu, welche der jüngsten Messungen am stärksten ins Gewicht fielen. Diese Ergänzung ist eine Erklärung: ein Artefakt, das zusammen mit der von einer machine learning (ML)-Methode gelieferten Vorhersage bereitgestellt wird, um deren Transparenz für den menschlichen Benutzer zu erhöhen.

Erklärungen können verschiedene Formen annehmen. Zum Beispiel können sie aus menschenlesbarem Text oder quantitativen Indikatoren bestehen, wie Merkmalswichtigkeit-Werten (Molnar, 2025; Ribeiro et al., 2016). Abb. 1 veranschaulicht zwei Arten von Erklärungen, beide für dieselbe gelernte Hypothese $\widehat{h}(x)$ eines einzelnen Merkmal $x$. Die erste Erklärung approximiert $\widehat{h}(x)$ um einen spezifischen Merkmal-Wert $x'$ durch eine lineare Funktion $g(x)$. Das Gewicht von $g(x)$, seine Steigung, dient als Merkmalswichtigkeit-Wert. Der zweite Erklärungstyp ist eine kleine Menge von Vorhersagen $\widehat{h}(x^{(1)}), \widehat{h}(x^{(2)}), \widehat{h}(x^{(3)})$ bei ausgewählten Merkmal-Werten, die möglicherweise vom Benutzer gewählt wurden.

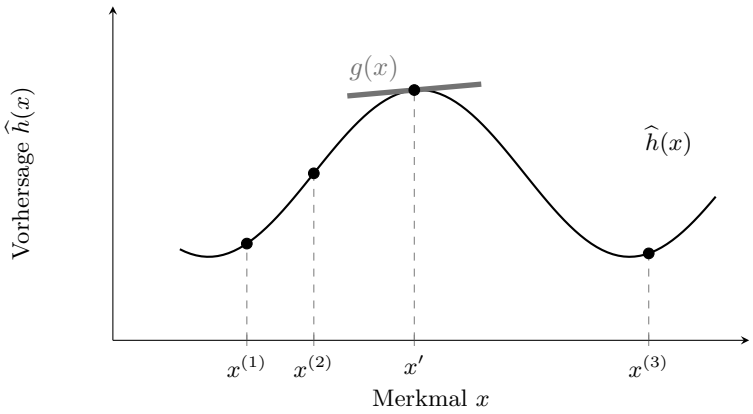



Abbildung 1: Eine gelernte Hypothese $\widehat{h}(x)$ eines einzelnen Merkmal $x$ (durchgezogene Kurve) mit zwei verschiedenen Erklärungen. Das dicke graue Segment ist eine lineare Approximation $g(x)$ von $\widehat{h}(x)$ am Punkt $x'$, markiert durch den gefüllten Punkt auf der Kurve. Die drei weiteren gefüllten Punkte sind die Vorhersagen $\widehat{h}(x^{(1)})$, $\widehat{h}(x^{(2)})$, $\widehat{h}(x^{(3)})$ bei den Merkmal-Werten $x^{(1)}$, $x^{(2)}$, $x^{(3)}$, die auf der horizontalen Achse markiert sind. Eine gestrichelte Linie fällt von jedem der vier gefüllten Punkte zu seinem Merkmal-Wert auf der horizontalen Achse

Eine weitere Form der Erklärung ist eine Heatmap, auch Intensitätskarte oder, für Bilder, Saliency Map genannt (Montavon et al., 2018; Zeiler and Fergus, 2014). Eine solche Karte weist jeder Region eines Bildes einen Wert zu, wie stark diese Region zur Vorhersage beigetragen hat, und zeigt diese Werte durch Farbintensitäten an, die über das Bild selbst gelegt werden (Selvaraju et al., 2017). Für ein Convolutional Neural Network (CNN) können die Werte aus seinen Aktivierungen berechnet werden, was eine class activation map (CAM) tut; Abb. 2 zeigt eine.

Neben ihrer Form unterscheiden sich Erklärungen auch in ihrem Umfang bezüglich des Merkmalsraum. Die lokale lineare Approximation von local interpretable model-agnostic explanations (LIME) liefert eine lokale Erklärung für eine gelernte Hypothese $\widehat{h}$ um einen spezifischen Merkmalsvektor $\mathbf{x} \in \mathcal{X}$. Ein weiteres Beispiel für eine lokale Erklärung ist ein counterfactual für einen Klassifikator $\widehat{h}$. Für einen gegebenen Merkmalsvektor $\mathbf{x}$ besagt ein Counterfactual eine (kleine) Modifikation $\mathbf{x} \mapsto \mathbf{x}'$, sodass sich die Vorhersage ändert, $\widehat{h}(\mathbf{x}) \neq \widehat{h}(\mathbf{x}')$ (Wachter et al., 2018). Der Gerichtshof der Europäischen Union hat solche Informationen als eine angemessene Erklärung einer automatisierten Entscheidung bezeichnet: das Ausmaß, in dem eine Variation der personal data zu einem anderen Ergebnis geführt hätte (Court of Justice of the European Union, 2025).

Im Gegensatz zu einer lokalen Erklärung approximiert eine globale Erklärung von $\widehat{h}$ diese über den gesamten Merkmalsraum. Eine solche Approximation kann durch das Training eines Modell, wie eines Entscheidungsbaum, erhalten wer-

den, um die Vorhersagen $\widehat{h}(\mathbf{x}^{(r)})$ für Merkmalsvektoren $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)}$ zu reproduzieren, die den gesamten Merkmalsraum gut abdecken (Molnar, 2025). Algorithmus 1 beschreibt diese Konstruktion: Die Merkmalsvektoren bilden ein Probing-Set, jedes mit der Vorhersage von $\widehat{h}$ gelabelt, und die Erklärung $g$ wird über empirische Risikominimierung (ERM) über einen Hypothesenraum $\mathcal{H}'$ von Kandidatenerklärungen gelernt. Der Umfang wird allein durch das Probing-Set festgelegt. Derselbe Algorithmus liefert die lokale Erklärung von LIME, wenn das Probing-Set aus Merkmalsvektoren besteht, die um das zu erklärende $\mathbf{x}$ gezogen werden (Ribeiro et al., 2016, Sect. 3.2); die lineare Funktion $g$ aus Abb. 1 ist ihre Ausgabe für ein Probing-Set nahe $x'$ und $\mathcal{H}'$ das lineares Modell.

**Algorithm 1** Erklärung durch ein Surrogat-Modell, nach Ribeiro et al. (2016), Sect. 3.2

**Input:** learned Hypothese $\widehat{h}$ to be explained; probing set of Merkmalsvektoren $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)}$; Hypothesenraum $\mathcal{H}'$ of understandable Hypothesen; Verlustfunktion $L(\cdot, \cdot)$

1: **for** $r = 1, \ldots, m$ **do**
2: $\quad y^{(r)} := \widehat{h}(\mathbf{x}^{(r)})$ ▷ the Vorhersage to be explained serves as Label
3: **end for**
4: $g \in \arg\min_{h \in \mathcal{H}'} \frac{1}{m} \sum_{r=1}^{m} L\left(\left(\mathbf{x}^{(r)}, y^{(r)}\right), h\right)$ ▷ ERM

**Output:** the explanation $g$ and its Trainingsfehler on the probing set

Erklärungen von (den Vorhersagen, die aus) ML-Methoden erhalten werden, sollten (i) verständlich und (ii) treu sein (Lipton, 2018; Ribeiro et al., 2016, Sect. 3). Verständlich bedeutet, dass der Benutzer mit der Erklärung selbst arbeiten kann: Allein daraus kann er antizipieren, wie sich die Vorhersage ändert, wenn sich die Merkmale ändern. In Algorithmus 1 fällt diese Anforderung auf die Wahl von $\mathcal{H}'$, die aus Hypothesen bestehen muss, die der Benutzer verstehen kann, wie lineare Abbildungen oder flache Entscheidungsbäume. Treu bedeutet, dass die Erklärung die Berechnung widerspiegelt, die die gelernte Hypothese durchführt. Für die Ausgabe $g$ von Algorithmus 1 misst der Trainingsfehler auf dem Probing-Set, wie treu $g$ dort $\widehat{h}$ reproduziert, und nirgendwo sonst. Eine Intensitätskarte über den Pixeln eines Bildes ist treu, wenn das Manipulieren einiger der von ihr hervorgehobenen Pixel die Vorhersage ändert, während dieselbe Manipulation von Pixeln, die sie dunkel lässt, dies nicht tut (Samek et al., 2017, Sect. III-A, p. 2666).

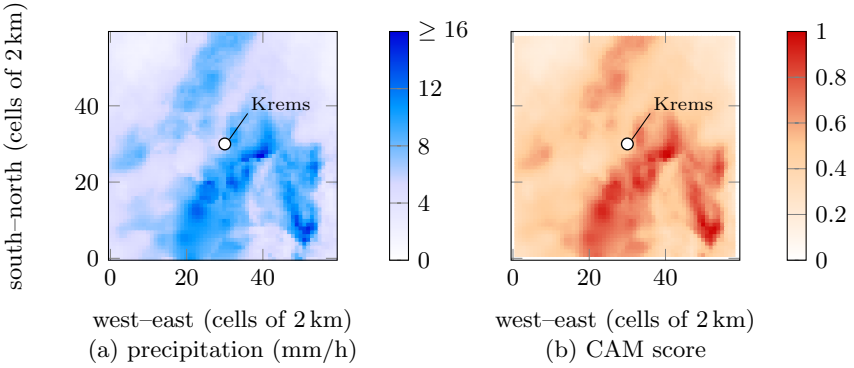
Verständlichkeit und Treue können widersprüchliche Anforderungen sein. Eine untreue Erklärung kann verständlich sein, während sie keine Informationen über die Berechnung enthält. Bei einigen Saliency-Methoden ist die aus einem künstliches neuronales Netz (KNN) mit zufälligen Gewichte berechnete Karte nahezu dieselbe wie die aus dem trainierten KNN berechnete Karte (Adebayo et al., 2018, Sect. 3). Eine solche Karte zeichnet die Kanten des Eingabebildes nach und enthält nichts über die Vorhersage, doch sie sieht wie eine plausible Erklärung aus. Eine Erklärung, die überall mit der gelernten Hypothese übereinstimmte, wäre wieder diese Hypothese, was für einen Benutzer

unverständlich sein könnte, sodass eine einfachere Erklärung an einer Stelle die Treue aufgeben muss (Ribeiro et al., 2016, Sect. 3.2).

Abb. 2 veranschaulicht den verständlichen Teil einer Erklärung für eine Vorhersage, die aus Wetterradar erstellt wurde. Die Merkmale sind der stündliche Niederschlag über einem 120 km-Kasten um Krems an der Donau, angegeben auf einem Gitter von Zellen von 2 km mal 2 km, den Pixeln des Radarbildes. Die Vorhersage ist, ob es in Krems zwei Stunden nach der gezeigten Stunde regnen wird. Ein auf diesen Bildern trainiertes CNN wird durch eine CAM (Selvaraju et al., 2017) erklärt, die in Panel (b) gezeigt wird. Eine zweite Erklärung, in Worten, ist unter den Panels abgedruckt, und ihre Konstruktion wird nach dem Treuetest beschrieben. Diese beiden Erklärungen könnten von einem nichtfachkundigen Benutzer einer Wetter-App als verständlich angesehen werden. Ihre Treue muss jedoch durch einen Störungstest überprüft werden.

Für die in Abb. 2(a) gezeigte Stunde weist das CNN dem Regen eine Wahrscheinlichkeit nahe eins zu, und die CAM in Panel (b) bewertet die Niederschlagsbänder südlich und südöstlich der Stadt am höchsten, anstatt den bereits über der Stadt liegenden Regen. Die Treue wird durch Störung des Radarbildes getestet. Das Setzen des Niederschlags auf Null in den 360 Zellen mit den höchsten CAM-Werten senkt das logit des Regens, die Zahl, die die Sigmoidfunktion in die vorhergesagte Wahrscheinlichkeit umwandelt, um 1.57, während das Nullsetzen der 360 Zellen mit den niedrigsten Werten es nur um 0.26 senkt. Dieselbe Reihenfolge gilt für 63 der 72 Stunden des Testdatensatz, die letzten 30 Prozent der aufgezeichneten Stunden, die nicht zum Trainieren des CNN verwendet wurden.

Abb. 2 zeigt auch eine Erklärung in Textform, den Satz, der unter den Panels abgedruckt ist. Eine feste Regel wandelt die CAM von Panel (b) in diesen Satz um: Sie sammelt die 360 Zellen, die die Karte am höchsten bewertet, dieselben Zellen, die der obige Störungstest auf Null setzt, und berechnet daraus zwei Größen. Der erste ist der Radius um Krems, der neun von zehn dieser Zellen enthält, auf ein Vielfaches von 5 km aufgerundet. Das letzte Zehntel befindet sich in den hintersten Ecken des Kastens und wird weggelassen, da es den Radius über den gesamten Kasten ausdehnen würde. Die zweite ist die Kompassrichtung, in der diese Zellen liegen, vom Zentrum von Krems aus gesehen: ihre durchschnittliche Position, jede Zelle gewichtet mit ihrem CAM-Wert, gerundet auf einen von 16 Kompasssektoren. Da die Regel fest ist, ist der Satz genau so treu wie die Karte, die er zusammenfasst, deren Treue der obige Störungstest überprüft. Der Satz kann auch mit dem Radarbild in Panel (a) verglichen werden.




Textual explanation: *"Rain is likely at Krems in two hours since precipitation has been recorded within a radius of 60 km, mainly to the south-south-east of the town."*

Abbildung 2: Eine Erklärung für eine Vorhersage, die am 15. September 2024 um 08:00 UTC aus Wetterradar erstellt wurde. (a) Der beobachtete Niederschlag in mm/h über einem 120 km-Kasten um Krems an der Donau (Österreich), markiert durch den Kreis, auf einem Gitter von 2 km-Zellen; die Farbskala reicht von weiß (trocken) bis dunkelblau (16 mm/h und darüber). Aus diesem Bild prognostiziert ein CNN Regen in Krems zwei Stunden später. (b) Die CAM, die jede Zelle danach bewertet, wie stark sie zur Vorhersage beigetragen hat; die Farbskala reicht von weiß (0) bis dunkelrot (1). Die höchsten Werte liegen auf den Bändern südlich und südöstlich der Stadt, anstatt auf dem Regen über der Stadt. Unten: eine textuelle Erklärung, generiert aus der CAM, die den Radius und die Richtung der Zellen benennt, die sie am höchsten bewertet. Daten generiert von `pythondemos/explanation.py`

Siehe auch: ML, Vorhersage, Merkmal, Datenpunkt, Klassifikation, Erklärbarkeit, erklärbare künstliche Intelligenz (XAI), interpretierbares maschinelles Lernen, LIME, SHapley Additive exPlanations (SHAP), counterfactual, CAM, Merkmalswichtigkeit, ERM.

## Querverweise

## Literatur

1. Adebayo et al. (2018). Sanity Checks for Saliency Maps. In: Adv. Neural Inf. Process. Syst. (NeurIPS), pp. 9505–9515.

2. Court of Justice of the European Union (2025). Judgment of 27 February 2025, CK v Dun & Bradstreet Austria GmbH and Magistrat der Stadt Wien, C-203/22, ECLI:EU:C:2025:117. Court of Justice of the

European Union, First Chamber. URL: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:62022CJ0203.

3. Selvaraju et al. (2017). Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. In: 2017 IEEE Int. Conf. Comput. Vis., pp. 618–626.

4. Lipton (2018). The Mythos of Model Interpretability: In machine learning, the concept of interpretability is both important and slippery. Queue, 16(3), pp. 31–57.

5. Molnar (2025). Interpretable Machine Learning: A Guide for Making Black Box Models Explainable. 3rd ed., Ebook.

6. Montavon et al. (2018). Methods for interpreting and understanding deep neural networks. Digital Signal Processing, 73, pp. 1–15.

7. Ribeiro et al. (2016). "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In: Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, pp. 1135–1144.

8. Samek et al. (2017). Evaluating the Visualization of What a Deep Neural Network Has Learned. IEEE Trans. Neural Netw. Learn. Syst., 28(11), pp. 2660–2673.

9. Wachter et al. (2018). Counterfactual Explanations Without Opening the Black Box: Automated Decisions and the GDPR. Harvard Journal of Law & Technology, 31(2), pp. 841–887.

10. Zeiler and Fergus (2014). Visualizing and Understanding Convolutional Networks. In: Proc. European Conf. Comput. Vis. (ECCV), pp. 818–833.