Merkmalswichtigkeit

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

Werte für die Merkmalswichtigkeit messen, wie stark eine gelernte Hypothese von einzelnen Merkmale abhängt. Solche Werte lassen sich aus den Gewichte einer linearen Hypothese gewinnen, die entweder aus den Labels eines Trainingsdatensatzs gelernt wurde, wobei sie die Wichtigkeit eines Merkmals für das Label messen, oder aus den Vorhersagen einer gegebenen Hypothese, wobei sie deren Abhängigkeit von dem Merkmal messen. Die Gewichte eines lineares Modells beschreiben die Änderung der Vorhersage pro Einheit eines Merkmals. Eine andere Konstruktion nutzt die Änderung des durchschnittlichen Verlustes, wenn die Werte eines Merkmals gestört werden, indem sie entweder durch unabhängige Ziehungen aus seiner eigenen Wahrscheinlichkeitsverteilung ersetzt oder über die Datenpunkte eines Datensatzes hinweg permutiert werden. Werte für die Merkmalswichtigkeit messen in erster Linie die Abhängigkeit eines trainierten Modells von einem Merkmal. Diese Werte spiegeln möglicherweise nicht die Relevanz des Merkmals für die Lernaufgabe selbst wider. Insbesondere kann ein trainiertes Modell ein relevantes Merkmal ignorieren, was zu einem niedrigen Wichtigkeitswert führt.

Definition

Von den täglichen Beobachtungen, welche die Wetterstation Krems an der Donau (Österreich) aufzeichnet, werden hier sechs verwendet: das Maximum und das Minimum der Lufttemperatur, der mittlere Luftdruck, die mittlere relative Luftfeuchtigkeit, die Sonnenscheindauer und der Niederschlag. Diese Rohbeobachtungen werden in einer maschinelles Lernen (ML)-Anwendung genutzt, in der Datenpunkte tägliche Wetterbedingungen repräsentieren. Das Label y eines solchen Datenpunkts ist die Tageshöchsttemperatur, die aus zwölf Merkmale $x_1, \ldots, x_{12}$ vorhergesagt werden soll. Diese Merkmale sind die sechs Beobachtungen des Vortages und die sechs Beobachtungen von vor zwei Tagen.

Abb. 1 zeigt beobachtete Tageshöchsttemperaturen zusammen mit ihren Vorhersagen, die aus einer mittels lineare Regression gelernten Hypothese $\widehat{h}$ gewonnen wurden. Jede Vorhersage (oder Prognose) $\widehat{h}(\mathbf{x})$ für einen Datenpunkt wird aus dessen Merkmalsvektor $\mathbf{x} = \big(x_1, \ldots, x_{12}\big)^\top$ berechnet. Ein Wert für die Wichtigkeit eines Merkmals misst, wie stark diese Vorhersage von einem bestimmten Merkmal des Datenpunkts abhängt. Er ordnet jedem Merkmal eines

Datenpunkts eine Zahl zu: Eine große Zahl kennzeichnet ein Merkmal, das die Vorhersage stark beeinflusst, während eine Zahl nahe null ein Merkmal kennzeichnet, das sie kaum beeinflusst. Von welchen der zwölf Merkmale die Vorhersagen aus Abb. 1 abhängen, wird durch die nachstehenden Wichtigkeitswerte gemessen.

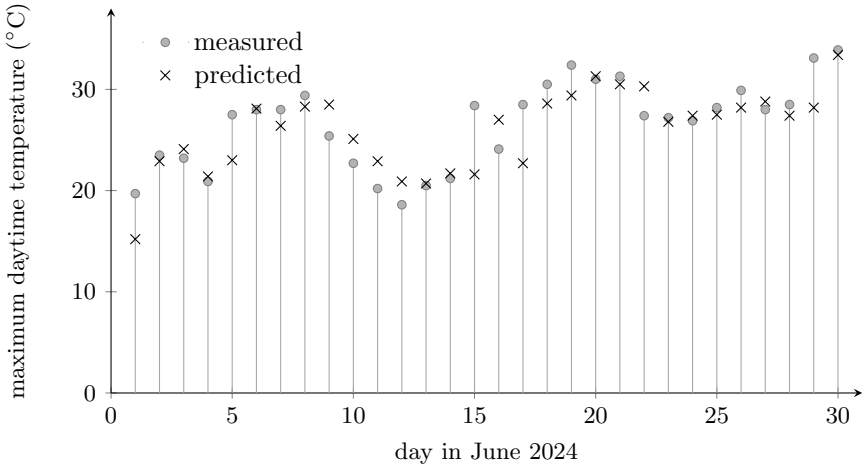


Abbildung 1: Das gemessene Label y, die Tageshöchsttemperatur (graue Linien mit kreisförmigen Markierungen), an der Wetterstation Krems an der Donau (Österreich) für Juni 2024 und die entsprechenden Vorhersagen $\widehat{h}(\mathbf{x})$ (schwarze Kreuze) einer Hypothese, die mittels lineare Regression aus zwölf Merkmale gelernt wurde, welche Beobachtungen während der zwei vorangegangenen Tage sind. Daten erzeugt mit `pythondemos/featureimportance.py`

ML-Methoden berechnen Werte für die Wichtigkeit von Merkmale für unterschiedliche Zwecke. Als Erklärung in der erklärbare künstliche Intelligenz (XAI) beschreibt ein Wert, wie stark eine gelernte Hypothese von einzelnen Merkmale eines Datenpunkts abhängt (Molnar, 2025). Bei der Modellauswahl können Wichtigkeitswerte die Auswahl einer Teilmenge der Merkmale der Datenpunkte leiten, woraufhin unter Verwendung dieser Teilmenge eine neue Hypothese gelernt wird (Guyon and Elisseeff, 2003, Abschn. 2).

Betrachtet sei ein lineares Modell mit der gelernten Hypothese

$$\widehat{h}(\mathbf{x}) = \sum_{j=1}^{12} \widehat{w}_j x_j.$$

Eine naheliegende Wahl für die Werte der Wichtigkeit eines Merkmals sind dessen Gewichte $\widehat{w}_j$. Eine Erhöhung des j-ten Merkmals eines Datenpunkts um eine Einheit ändert die Vorhersage um $\widehat{w}_j$. Das Vorzeichen von $\widehat{w}_j$ bestimmt, ob die Vorhersage mit dem Merkmal steigt oder fällt, und der Betrag $|\widehat{w}_j|$, um

wie viel. Ein Merkmal x_j mit einem großen Betrag $|\widehat{w}_j|$ kann dann als wichtig angesehen werden.

Eine Herausforderung bei der Verwendung der durch lineare Regression gelieferten Gewichte $\widehat{w}_j$ als Werte für die Wichtigkeit von Merkmale ist ihre Abhängigkeit von der Skala der Werte des Merkmals. Ersetzt man beispielsweise das Merkmal x_1 durch $10^3 x_1$, so lernt die lineare Regression ein Gewicht $\widehat{w}_1$, das um den Faktor 10^3 kleiner ist, während jede Vorhersage gleich bleibt. Durch Umskalieren eines Merkmals oder Ändern seiner Maßeinheit (z. B. Millimeter statt Meter) kann seine Wichtigkeit daher beliebig groß oder klein gemacht werden. Diese Abhängigkeit wird vermieden, indem die lineare Regression auf standardisierte Merkmale angewendet wird (siehe Merkmalsstandardisierung): Jede Spalte der Merkmalsmatrix wird um ihren Stichprobenmittel zentriert und durch ihre Standardabweichung geteilt, sodass $\widehat{w}_j$ zur Änderung der Vorhersage pro Standardabweichung des j-ten Merkmals wird.

Algorithmus 1 fasst die Konstruktion von Werten für die Wichtigkeit eines Merkmals durch Anwendung der lineare Regression auf die standardisierten Werte der Merkmale eines gegebenen Datensatzes mit der Merkmalsmatrix $\mathbf{X} = \left(\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)}\right)^\top \in \mathbb{R}^{m \times d}$ zusammen. Seine zweite Eingabe, der Labelvektor $\mathbf{y}$, kann auf zwei Arten gewählt werden, und diese Wahl entscheidet darüber, was die Werte bedeuten.

Die erste Wahl für die Einträge der Eingabe $\mathbf{y}$ in Algorithmus 1 sind die Labels der Datenpunkte, $\mathbf{y} = \left(y^{(1)}, \ldots, y^{(m)}\right)^\top$. Bei dieser Wahl misst der von Algorithmus 1 gelieferte Wert I_j, wie stark eine aus dem Datensatz gelernte lineare Hypothese das j-te Merkmal nutzt, um das Label vorherzusagen. Somit misst der Wert I_j die Wichtigkeit des Merkmals x_j für das Label selbst, also für die Lernaufgabe, und kann daher die Auswahl einer Teilmenge von Merkmale während der Modellauswahl leiten (Guyon and Elisseeff, 2003, Abschn. 2).

Die zweite Wahl für die Einträge der Eingabe $\mathbf{y}$ in Algorithmus 1 sind die Vorhersagen einer gegebenen gelernten Hypothese (oder eines trainierten Modells) $\widehat{h}$, $\mathbf{y} = \left(\widehat{h}(\mathbf{x}^{(1)}), \ldots, \widehat{h}(\mathbf{x}^{(m)})\right)^\top$. Die Gewichte sind diejenigen einer linearen Approximation der (möglicherweise nichtlinearen) Hypothese $\widehat{h}$. Bei dieser Wahl messen die von Algorithmus 1 gelieferten Werte I_j, wie stark $\widehat{h}$ vom j-ten Merkmal abhängt. Hier dienen die Werte als eine Erklärung für die von $\widehat{h}$ gelieferten Vorhersagen (Ribeiro et al., 2016, Abschn. 3.2).

Für eine lineare $\widehat{h}$ stimmen die beiden Möglichkeiten überein, wann immer $\widehat{h}$ selbst mittels lineare Regression auf demselben Datensatz gelernt wurde, da die lineare Regression bei Anwendung auf ihre eigenen Vorhersagen, die bereits eine lineare Funktion der Merkmale sind, ihre Gewichte zurückgibt. Im Gegensatz dazu unterscheiden sich die beiden Möglichkeiten für eine nichtlineare $\widehat{h}$, wie ein einfaches Experiment mit einem Random Forest unten zeigt.

Algorithm 1 Merkmalswichtigkeit aus der lineare Regression auf standardisierten Merkmale

Input: Merkmalsmatrix $\mathbf{X} = \left(\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)}\right)^\top \in \mathbb{R}^{m \times d}$ with entries $x_j^{(r)}$; Labelvektor $\mathbf{y}$, whose entries are either the Labels of the Datenpunkte, $\mathbf{y} = \left(y^{(1)}, \ldots, y^{(m)}\right)^\top$, or the Vorhersagen $\mathbf{y} = \left(\widehat{h}(\mathbf{x}^{(1)}), \ldots, \widehat{h}(\mathbf{x}^{(m)})\right)^\top$ of a learned Hypothese $\widehat{h}$ to be analyzed

1: **for** $j = 1, \ldots, d$ **do**

2: $\quad \widehat{\mu}_j := \frac{1}{m} \sum_{r=1}^{m} x_j^{(r)}, \quad \widehat{\sigma}_j := \sqrt{\frac{1}{m} \sum_{r=1}^{m} \left(x_j^{(r)} - \widehat{\mu}_j\right)^2}$

3: $\quad$ replace every entry $x_j^{(r)}$ of column j by $\left(x_j^{(r)} - \widehat{\mu}_j\right)/\widehat{\sigma}_j$ ▷ standardize

4: **end for**

5: $\bar{y} := \frac{1}{m} \mathbf{1}^\top \mathbf{y}$; replace $\mathbf{y}$ by $\mathbf{y} - \bar{y}\mathbf{1}$, with $\mathbf{1} \in \mathbb{R}^m$ the vector of ones ▷ center the entries of $\mathbf{y}$

6: $\widehat{\mathbf{w}} := \arg\min_{\mathbf{w} \in \mathbb{R}^d} \frac{1}{m} \|\mathbf{y} - \mathbf{X}\mathbf{w}\|^2$ ▷ lineare Regression

Output: importance scores $I_j := |\widehat{w}_j|$ for $j = 1, \ldots, d$: the importance of the j-th Merkmal for the Label or for the Vorhersagen of $\widehat{h}$, according to the input

Eine andere Konstruktion eines Werts für die Wichtigkeit eines Merkmals lernt überhaupt keine Hypothese. Ihre Eingabe ist eine Hypothese $\widehat{h}$, die bereits durch eine ML-Methode gelernt wurde, und ihre Ausgabe bewertet, wie stark $\widehat{h}$ von jedem Merkmal abhängt. Diese Konstruktion, die für Random Forest unter der Bezeichnung Variablenwichtigkeit (engl. *variable importance*) eingeführt wurde (Breiman, 2001, Abschn. 10), misst die Änderung des durchschnittlichen Verlustes, wenn die Werte eines einzelnen Merkmals durch unabhängige Ziehungen aus der eigenen Wahrscheinlichkeitsverteilung dieses Merkmals ersetzt werden. Bei einem Random Forest wird jeder Entscheidungsbaum anhand der Datenpunkte des Trainings bewertet, die nicht in seinem eigenen, durch Resampling gewonnenen Trainingsdatensatz enthalten waren, und der Wert ist der relative Anstieg der Fehlklassifikationsrate. Algorithmus 2 fasst die Methode in einer Form zusammen, die auf jede gelernte Hypothese anwendbar ist. Fisher et al. (2019), Abschn. 3 führen diese Form unter der Bezeichnung Modellabhängigkeit (engl. *model reliance*) für eine feste Hypothese mit der Permutationsschätzung aus ihrem Abschn. 3.1 ein und zeigen (Fisher et al., 2019, Thm. 5), dass der Wert auf dem Trainingsdatensatz selbst geschätzt werden kann.

Die Eingabe von Algorithmus 2 ist ein Datensatz mit einer Merkmalsmatrix $\mathbf{X} \in \mathbb{R}^{m \times d}$ und einem Labelvektor $\mathbf{y} \in \mathbb{R}^m$ sowie eine gelernte Hypothese (oder ein trainiertes Modell) $\widehat{h}$. Dieser Datensatz wird als Bewertungs-Datensatz bezeichnet. Er sollte durch dieselbe Wahrscheinlichkeitsverteilung erzeugt werden wie das zum Lernen von $\widehat{h}$ verwendete Trainingsdatensatz: Idealerweise sind die Datenpunkte von beiden Realisierungen von unabhängig und identisch verteilt (u.i.v.) Zufallsvariablen (ZV).
Für das j-te Merkmal liefert Algorithmus 2 einen Wert I_j, der den durchschnittlichen Verlust von $\widehat{h}$ auf dem Datensatz $\left(\mathbf{X}, \mathbf{y}\right)$ mit seinem durchschnittlichen

Verlust auf einem gestörten Datensatz vergleicht. Dieser Datensatz behält $\mathbf{y}$ bei, hat aber eine andere Merkmalsmatrix,

$$\widetilde{\mathbf{X}}^{(j)} = \left(\widetilde{\mathbf{x}}^{(1)}, \ldots, \widetilde{\mathbf{x}}^{(m)}\right)^\top.$$

Die gestörte Merkmalsmatrix entsteht durch Permutieren der Einträge der j-ten Spalte von $\mathbf{X}$ über ihre Zeilen hinweg, wobei jede andere Spalte unverändert bleibt (Abb. 2). Die Permutationswichtigkeit des j-ten Merkmals ist der Anstieg des durchschnittlichen Verlusts, den dieser Austausch verursacht,

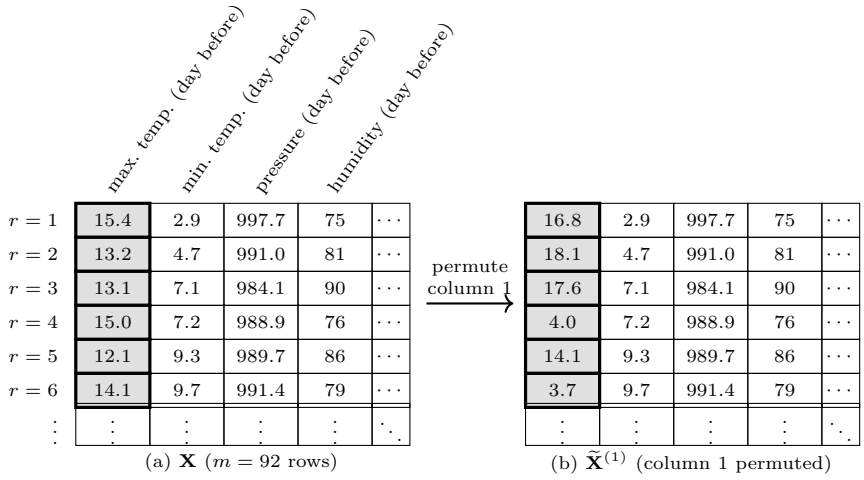
$$I_j := \frac{1}{m}\sum_{r=1}^{m} L\left(\left(\widetilde{\mathbf{x}}^{(r)}, y^{(r)}\right), \widehat{h}\right) - \frac{1}{m}\sum_{r=1}^{m} L\left(\left(\mathbf{x}^{(r)}, y^{(r)}\right), \widehat{h}\right),$$

wobei $y^{(r)}$ das Label des Datenpunkts r ist. Die Permutation ist zufällig, weshalb Algorithmus 2 I_j über B unabhängige Permutationen der j-ten Spalte von $\mathbf{X}$ mittelt.

Algorithm 2 Permutationswichtigkeit (Variablenwichtigkeit), ähnlich zu Fisher et al. (2019), Abschn. 3.1

Input: scoring Datensatz with Merkmalsmatrix $\mathbf{X} = \left(\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)}\right)^\top$ and Labelvektor $\mathbf{y} = \left(y^{(1)}, \ldots, y^{(m)}\right)^\top$; learned Hypothese $\widehat{h}$; Verlust $L(\cdot,\cdot)$; number B of permutations

$\widehat{L} := \frac{1}{m}\sum_{r=1}^{m} L\left(\left(\mathbf{x}^{(r)}, y^{(r)}\right), \widehat{h}\right)$ ▷ average Verlust on the Datensatz
for $j = 1, \ldots, d$ **do**
  for $b = 1, \ldots, B$ **do**
    $\widetilde{\mathbf{X}}^{(j)} :=$ $\mathbf{X}$ with the entries of column j permuted uniformly at random across the rows; denote its rows by $\left(\widetilde{\mathbf{x}}^{(r)}\right)^\top$
    $\widetilde{L}^{(b)} := \frac{1}{m}\sum_{r=1}^{m} L\left(\left(\widetilde{\mathbf{x}}^{(r)}, y^{(r)}\right), \widehat{h}\right)$
  end for
  $I_j := \frac{1}{B}\sum_{b=1}^{B} \widetilde{L}^{(b)} - \widehat{L}$
end for
Output: importance scores $I_1, \ldots, I_d$ for each Merkmal $x_1, \ldots, x_d$



(a) $\mathbf{X}$ ($m = 92$ rows) (b) $\widetilde{\mathbf{X}}^{(1)}$ (column 1 permuted)

Abbildung 2: (a) Die Merkmalsmatrix $\mathbf{X}$ eines Datensatzes, der Algorithmus 2 übergeben wird. Dieser Datensatz stammt aus täglichen Wetterbeobachtungen in Krems an der Donau. Jede Zeile enthält einen Datenpunkt, der einen bestimmten Tag repräsentiert. Die schattierte Spalte enthält die Höchsttemperatur des Vortages. Die erste Zeile ist der Datenpunkt, der den 1. Oktober 2024 repräsentiert, dessen erstes Merkmal, 15.4, die Höchsttemperatur vom 30. September ist. (b) Die gestörte Merkmalsmatrix $\widetilde{\mathbf{X}}^{(1)}$, die durch Permutieren der Einträge dieser Spalte über die 92 Zeilen hinweg entsteht, während jede andere Spalte unverändert bleibt. Die Permutationswichtigkeit des Merkmals ist der Anstieg des durchschnittlichen Verlusts, wenn die Vorhersagen aus den Zeilen von $\widetilde{\mathbf{X}}^{(1)}$ anstelle der Zeilen von $\mathbf{X}$ berechnet werden; die gezeigte Permutation ist die erste gezogene. Daten erzeugt mit `pythondemos/featureimportance.py`

Abb. 3 zeigt die von Algorithmus 2 mit dem quadratischer Verlust und $B = 20$ berechneten Werte I_j für eine lineare Hypothese, die aus den Wetter-Datenpunkte von Krems an der Donau gelernt wurde. Der Bewertungs-Datensatz besteht aus den 92 Datenpunkte, deren Label die Höchsttemperatur eines Tages vom 1. Oktober bis zum 31. Dezember 2024 ist, sodass ihre Merkmale die Beobachtungen vom 29. September bis zum 30. Dezember sind. Die lineare Hypothese aus Abb. 1 wurde auf dem Trainingsdatensatz der 272 Datenpunkte gelernt, deren Label-Tage vom 3. Januar bis zum 30. September reichen. Die Höchsttemperatur des Vortages erhält den größten Wert, während die Tiefsttemperatur des Vortages einen Wert nahe null erhält. Diese Tiefsttemperatur ist nicht nutzlos: ihre Korrelation mit dem Label auf dem Bewertungs-Datensatz beträgt 0.69. Der Wert I_j spiegelt in erster Linie wider, wie stark sich $\widehat{h}$ auf das Merkmal x_j stützt, was sich von dessen Relevanz für das Label eines Datenpunkts unterscheiden kann.

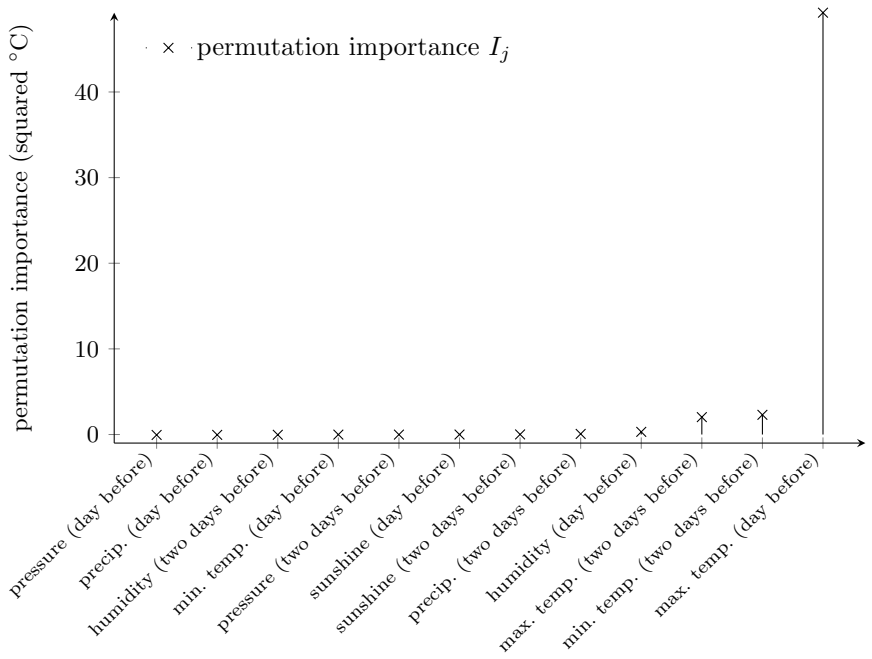


Abbildung 3: Permutationswichtigkeitswerte (Algorithmus 2, $B = 20$) für die zwölf Merkmale der Datenpunkte, die für die lineare Hypothese $\widehat{h}$ aus Abb. 1 verwendet wurden, in aufsteigender Reihenfolge von I_j (schwarze Kreuze). Algorithmus 2 verwendete einen Bewertungs-Datensatz, der aus 92 Datenpunkte besteht, welche die Tage vom 1. Oktober bis zum 31. Dezember 2024 repräsentieren. Im Gegensatz dazu wurde die lineare Hypothese aus Datenpunkte gelernt, welche die Tage vom 3. Januar bis zum 30. September repräsentieren. Achsenbeschriftungen: Höchst- und Tiefstlufttemperatur, Niederschlag sowie Vortag und zwei Tage zuvor zählen von dem Tag zurück, dessen Höchsttemperatur das Label ist. Daten erzeugt mit `pythondemos/featureimportance.py`

Beide obigen Algorithmen greifen auf $\widehat{h}$ nur über deren Vorhersagen zu und sind daher auf jede gelernte Hypothese anwendbar. Sie unterscheiden sich darin, was sie bewerten: Algorithmus 1 bewertet eine lineare Approximation von $\widehat{h}$, während Algorithmus 2 $\widehat{h}$ selbst bewertet, sodass seine Werte auch dann aussagekräftig sind, wenn keine lineare Approximation genau ist. Beide bewerten über einen gesamten Datensatz hinweg, wie stark sich $\widehat{h}$ auf ein Merkmal stützt, im Gegensatz zu einer lokalen Erklärung wie lokale interpretierbare modellunabhängige Erklärungen (LIME), die $\widehat{h}$ um einen einzelnen Datenpunkt herum approximiert.

Abb. 4 zeigt die von Algorithmus 2 gelieferten Werte für einen Random Forest, der auf demselben Trainingsdatensatz gelernt und auf demselben

Bewertungs-Datensatz ausgewertet wurde wie die lineare Hypothese aus Abb. 3. Beide erreichen nahezu denselben durchschnittlichen quadratischer Verlust auf dem Bewertungs-Datensatz, 8.2 in quadrierten Grad Celsius, stützen sich aber unterschiedlich auf die Merkmale. Algorithmus 1 zeigt denselben Unterschied, wenn seine Eingabe von den Labelsn zu den Vorhersagen wechselt: gefüttert mit den Labelsn des Trainingsdatensatzs, bewertet er die Höchsttemperatur des Vortages mit 8.44 und die Tiefsttemperatur des Vortages mit 0.09, während er sie, gefüttert mit den Vorhersagen des Random Forests auf demselben Trainingsdatensatz, mit 6.16 und 0.40 bewertet. Der Random Forest stützt sich somit auf die Tiefsttemperatur des Vortages, ein Merkmal, das die aus den Labelsn gelernte lineare Hypothese ignoriert.

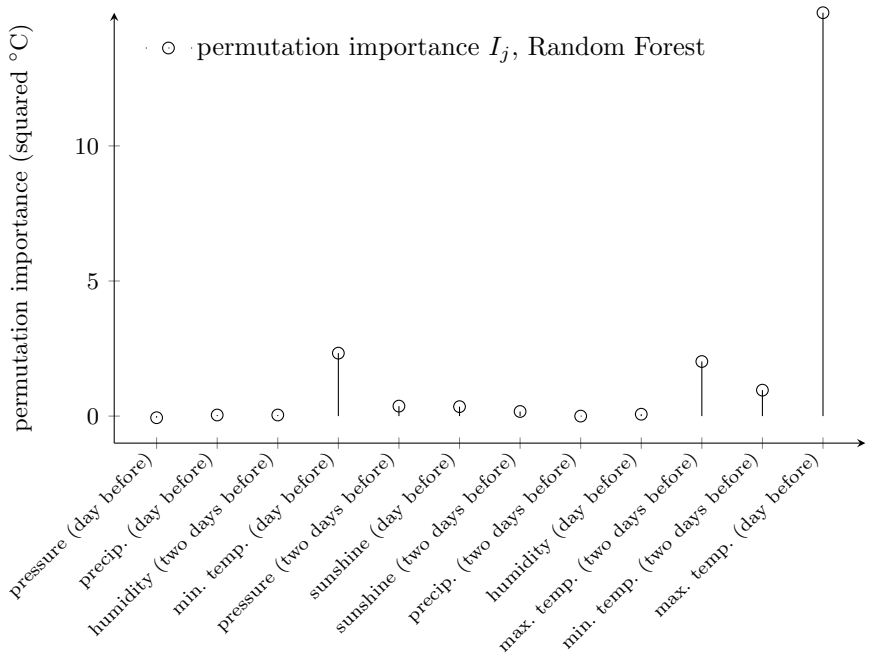


Abbildung 4: Wichtigkeitswerte I_j aus Algorithmus 2 ($B = 20$; schwarze Kreise) für den Wetter-Datensatz aus Abb. 1, für einen Random Forest, der auf dem Trainingsdatensatz mit Label-Tagen vom 3. Januar bis zum 30. September 2024 gelernt und auf dem Bewertungs-Datensatz aus Abb. 3 mit Label-Tagen vom 1. Oktober bis zum 31. Dezember bewertet wurde, in der Merkmalsreihenfolge jener Abbildung, sodass beide Spalte für Spalte verglichen werden können. Achsenbeschriftungen wie in Abb. 3. Daten erzeugt mit `pythondemos/featureimportance.py`

Synonyme: Merkmalsrelevanz, Variablenwichtigkeit.
Siehe auch: Merkmal, Merkmalsmatrix, lineares Modell, Verlust, Random

Forest, LIME, Erklärung, Kovarianz.

Querverweise

Literatur

1. Breiman (2001). Random Forests. Machine Learning, 45(1), pp. 5–32.
2. Fisher et al. (2019). All Models are Wrong, but Many are Useful: Learning a Variable's Importance by Studying an Entire Class of Prediction Models Simultaneously. Journal of Machine Learning Research, 20(177), pp. 1–81.
3. Molnar (2025). Interpretable Machine Learning: A Guide for Making Black Box Models Explainable. 3rd ed., Ebook.
4. 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.
5. Guyon and Elisseeff (2003). An introduction to variable and feature selection. Journal of machine learning research, 3(Mar), pp. 1157–1182.