Themenblock 10 Robust/Hybrid/Robust AI 10_05Ergebnis_SHAP

Kurzeinführung in SHAP

Erarbeitet von
Marc Feger M.Sc.

Lernziele

- Du verstehst die Grundprinzipien des SHAP-Frameworks und seine Anwendung auf die Interpretation von KI-Modellen
- Du kennst die Schritte zur Berechnung von SHAP-Werten und ihre Anwendung in einem praktischen Beispiel der Sentiment Analyse
- Du erkennst die Wichtigkeit von Transparenz in KI-Entscheidungen und die Rolle von SHAP bei der Schaffung von Vertrauen in KI-Systeme

Inhalt

Einstieg

Quelle [1- 5]

Herzlich willkommen zu diesem Video.

Eine Herangehensweise im Umgang in der Interpretierbarkeit von KI-Methoden ist das sogenannte SHAP-Framework.

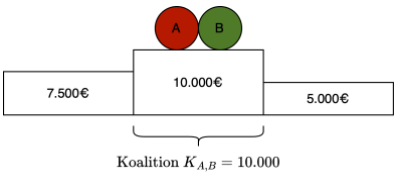
SHAP steht dabei für SHapley Additive exPlanations, wobei das Framework für eine Sammlung an unterschiedlichen Methoden-Familien steht. SHAP ist eine Methode und bietet tiefere Einblicke in das, „Warum" hinter den Vorhersagen von KI-Modellen, indem sie den Beitrag jedes Merkmals zur endgültigen Entscheidung quantifiziert.
Die Formel mag anfangs umfangreich erscheinen, ich werde dir aber zeigen, dass sie im Prinzip sehr intuitiv ist:
In diesem Video führe ich Dich in die Technik hinter den SHAP-Werten ein.

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

Ich erläutere Dir, wie diese Techniken funktionieren, warum sie wichtig sind und wie sie eingesetzt werden können, um die KI-Entscheidungsfindung nachzuvollziehen.

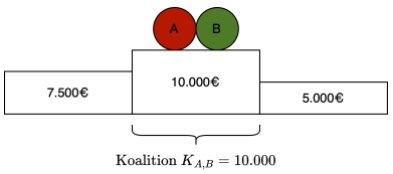
Einführung in SHAP-Werte

Stell Dir vor, Du möchtest den Einfluss jedes einzelnen Features (also jeder einzelnen Eigenschaft) Deines Datensatzes auf die Entscheidungen, die Dein KI-Modell trifft, nicht nur erkennen, sondern auch messen. Die Idee hinter den SHAP-Werten stammt aus der Spieltheorie, genauer gesagt den Shapley-Werten. Diese bieten eine faire Methode, um zu bestimmen, wie der „Gewinn" – oder in unserem Fall der Beitrag zum Endergebnis – unter den Spielern, also den Features aufgeteilt wird, wenn sie alleinstehen oder zusammenarbeiten.



Nehmen wir dazu an, dass eine Koalition aus zwei Spielern an einem Spiel teilnehmen und dabei 10.000 Euro gewinnen. Beide Spieler fragen sich, wie der Gewinn fair aufgeteilt werden kann, da Spieler A und Spieler B unterschiedliche Erfahrung mit dem Spiel haben.

Weiterhin gilt, dass eine 50/50-Aufteilung der individuellen Leistung der Spieler nicht gerecht wird und daher der marginale Beitrag jedes Spielers bestimmt werden muss. Der marginale Beitrag ist der Anstieg des Koalitions-Wertes, wenn ein bestimmter Spieler einer Koalition beitritt.



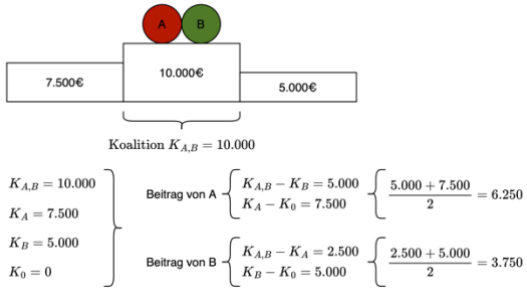
$K_{A,B} = 10.000$

$K_A = 7.500$

$K_B = 5.000$

$K_0 = 0$

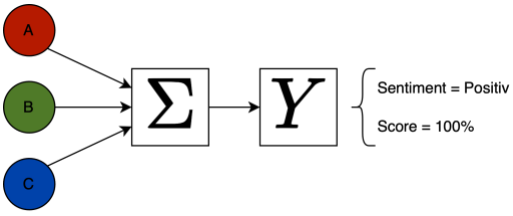
Nehmen wir weiterhin an, dass wir das Spiel beliebig mit individuellen Spieler-Koalitionen durchführen können und die Ergebnisse messen können. Es zeigt sich, dass A und B eine Gewinner-Kombination ist, aber A allein auch besser als B abschneidet. Wenn niemand antritt, dann gibt es auch keinen Gewinn. Um zu quantifizieren, wie hoch A‘s Beitrag tatsächlich ausfällt, betrachten wir den Durchschnitt der jeweiligen Anstiege im Gesamtergebnis, wenn A am Spiel beteiligt wird. Alternativ können wir dieselbe Berechnung für B durchführen und sehen, dass A tatsächlich mehr zum Spiel-Ergebnis beiträgt als B:



In der Summe ergibt sich wieder der Gesamtgewinn, wobei die einzelnen Ergebnisse auch direkt SHAP-Werte sind. SHAP-Werte sind also eine faire Art, den Gewinn eines Spiels unter den Spielern zu verteilen. Dabei gibt ein einzelner SHAP-Wert den erwarteten marginalen Beitrag, also den gewichteten durchschnittlichen Beitrag eines Spielers, an.

Einführung in SHAP-Werte

Kommen wir nun zur Interpretation von KI-Modellen, wobei wir Beobachtungen bezüglich der Eingabe und der Ausgabe machen können. Ein solches Szenario ist nichts anderes als ein Spiel, in dem wir die Regeln nicht kennen, aber wissen, welche Spieler, also Features und Ergebnisse vorliegen. Nehmen wir an, dass wir drei Features (A, B, C) eines Klassifikators und dessen Ausgabe kennen. Die Frage lautet also, wie stark der Einfluss eines bestimmten Features auf das finale Ergebnis des Klassifikators ist.



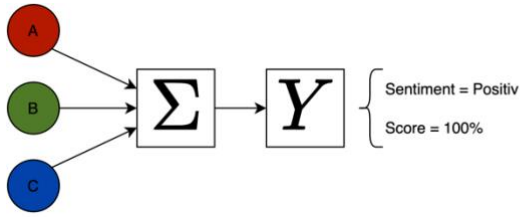
$K_{A,B,C} = 100\%$

$K_{A,B} = 75\%$
$K_{A,C} = 75\%$
$K_{B,C} = 50\%$

$K_A = 50\%$
$K_B = 50\%$
$K_C = 0\%$
$K_0 = 0\%$

Dabei können wir alle möglichen Feature-Koalitionen aufstellen und beobachten, wie sich der Ausgabe-Score verändert. Wieder zeigt sich, dass einige Kombinationen bzw. individuelle Features besser als andere abschneiden.
Zu Einfachheit wollen wir den Einfluss des Features A berechnen. Daher betrachten wir alle Koalitionen, in denen A vorkommt.

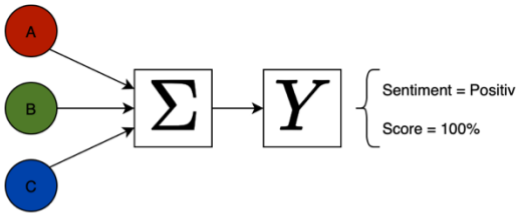
Auch hier zeigt sich, dass A’s Anwesenheit unterschiedlich starke Auswirkungen auf die Leistung der Koalitionen hat. Anders als in unserem initialen Beispiel müssen aber die jeweiligen marginalen Beiträge „fair” bewertet werden, da einige wahrscheinlicher sind als andere.



$$K_{A,B,C} - K_{B,C} = 50\%$$
$$K_{A,B} - K_{B} = 25\%$$
$$K_{A,C} - K_{C} = 75\%$$
$$K_{A} - K_{0} = 50\%$$

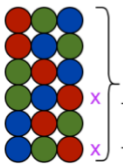
$$\frac{1}{3} \times 50\% + \frac{1}{6} \times 25\% + \frac{1}{6} \times 75\% + \frac{1}{3} \times 50\% = 50\%$$

Wir sehen also, dass A einen zu erwartenden marginalen Beitrag von 50 % zum Gesamtergebnis hat und damit eines der stärkeren Features für den Klassifikator sein muss. Aber wie kommen die einzelnen Gewichtungen zustande?



$$K_{A,B,C} - K_{B,C} = 50\% \quad \Big\} \frac{1}{3} \times 50\% + \dots$$

$$K_{A,B,C} - K_{B,C} = 50\% \Rightarrow P(K_{A,B,C} - K_{B,C}) = \frac{1}{3}$$



$$3! = 6$$
$$2! \times 1! = 2 \quad \Big\} \quad \frac{2}{6} = \frac{1}{3}$$

Um zu verstehen, wie die jeweiligen Gewichtungen zustande kommen, betrachten wir das Folgende. Relevant zur Bestimmung der Gewichtung mit 1/3 für den ersten Vergleich ist die Betrachtung, wie viele Beitrittskombinationen es insgesamt gibt und inwiefern A dabei einer bestehenden Koalition von B und C beitritt. Insgesamt haben wir 6 Möglichkeiten, in denen die Features einer Koalition beitreten können. In zwei dieser Fälle tritt A einer bestehenden Koalition aus B und C zu. Daher ergibt sich eine Wahrscheinlichkeit von 2/6 oder 1/3 für den Beitritt von A zu einer bestehenden Koalition aus B und C. Analog kann man die restlichen Wahrscheinlichkeiten berechnen.

Zusammenfassung

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

Gewichtung | Marginaler Beitrag

Insgesamt ergibt sich der SHAP-Wert eines Features aus der Summe der marginalen Beiträge dieses Features über alle möglichen Koalitionen von Features, denen es beitreten kann. Um ein umfassendes Verständnis der SHAP-Werte (Shapley Additive Explanation) zu erlangen, empfiehlt sich die Verwendung des Python-Frameworks SHAP. Dieses Framework ist nicht nur auf die Analyse von Textdaten beschränkt, sondern bietet auch Unterstützung für eine Vielzahl von Modelltypen, einschließlich Bäumen, linearen Modellen und Transformer-Modellen. Es ermöglicht die Anwendung auf unterschiedlichste Datenarten, darunter sowohl Text als auch Bilder.

Ein konkretes Beispiel im Framework ist die Erläuterung von Sentiment-Scores basierend auf dem IMDb-Datensatz unter Verwendung von Transformer-Modellen. Darüber hinaus umfasst das SHAP-Framework diverse Implementierungen und Anwendungsbeispiele, die die Flexibilität und Breite der unterstützten Modelle und Datenformate demonstrieren. Es bietet somit tiefgreifende Einblicke und Erklärungen für die Entscheidungsfindung künstlicher Intelligenz über ein breites Spektrum von Anwendungsfällen hinweg.

Quellen

Quelle [1] Molnar (2023). Interpretable Machine Learning: A Guide for Making Black Box Models Explainable. https://christophm.github.io/interpretable-ml-book/shapley.html

Quelle [2] Lundberg and Lee (2017). A Unified Approach to Interpreting Model Predictions. https://proceedings.neurips.cc/paper_files/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf

Quelle [3] Lundberg (2018). SHAP Python-Framework. https://shap.readthedocs.io/en/latest/

Quelle [4] A Data Odyssey (2023). The mathematics behind Shapley Values. https://www.youtube.com/watch?v=UJeu29wq7d0

Quelle [5] Shapley (1951), The value of an n-player game. https://www.rand.org/content/dam/rand/pubs/research_memoranda/2008/RM670.pdf

Weiterführendes Material

Lundberg (2018). SHAP Python-Framework. https://shap-lrjball.readthedocs.io/en/latest/index.html

Disclaimer