h_da
HOCHSCHULE DARMSTADT
UNIVERSITY OF APPLIED SCIENCES
fbmn
FACHBEREICH MATHEMATIK
UND NATURWISSENSCHAFTEN

h_da
HOCHSCHULE DARMSTADT
UNIVERSITY OF APPLIED SCIENCES
fbi
FACHBEREICH INFORMATIK

Hochschule Darmstadt

Fachbereiche Mathematik und Naturwissenschaften & Informatik

Erklärbarkeit maschineller Lernverfahren

Feature Engineering mit Black-Box-Modellen

Abschlussarbeit zur Erlangung des akademischen Grades

Master of Science (M. Sc.)
im Studiengang Data Science

vorgelegt von

Christophe Krech

Referent	:	Prof. Dr. Markus Döhring
Korreferent	:	Prof. Dr. Horst Zisgen

Ausgabedatum	:	1. April 2019
Abgabedatum	:	16. September 2019

ERKLÄRUNG

Ich versichere hiermit, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die im Literaturverzeichnis angegebenen Quellen benutzt habe.

Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder noch nicht veröffentlichten Quellen entnommen sind, sind als solche kenntlich gemacht.

Die Zeichnungen oder Abbildungen in dieser Arbeit sind von mir selbst erstellt worden oder mit einem entsprechenden Quellennachweis versehen.

Diese Arbeit ist in gleicher oder ähnlicher Form noch bei keiner anderen Prüfungsbehörde eingereicht worden.

Darmstadt, 16. September 2019

Christophe Krech

ABSTRACT

Especially in regulated industries, linear models continue to enjoy great popularity. They have been tried and tested in practice and the regulatory authorities are familiar with their interpretation. Despite years of experience with these models, the developers are often not able to achieve the performance of modern machine learning methods like treebased ensembles or neural networks.
The introduction of the General Data Protection Regulation (GDPR) enhances the need for explainable and trustworthy machine learning methods. The aim is to develop a model with the highest possible prediction quality, that can provide the highest possible level of explainability. However, both requirements are diametrically opposed - at least for many problems. In the meantime, numerous methods have been developed to post-hoc explain the decisions of a black box model. These methods allow deep insights into the behaviour of the models, but cannot be compared with intrinsically explainable methods like logistic regression in terms of their interpretability.
In this thesis two approaches are presented to combine the better predictive power of the modern, highly complex models with the explainable character of a logistic regression. These approaches implement a performance-oriented feature engineering for white-box models using the explanations of a more powerful black box approach. For this purpose, two different methods are developed, which can be applied independently or together. Depending on the methodology and the problem, these either increase the performance of the explainable model or its explainability. In some cases even both is possible.
The first methodology makes use of the structural differences between white-box and black-box methods with regard to the functional form of the resulting models. This indicates that transformations of the feature space are derived which maintain the explanatability of the model. The second method only considers observations if the black box model comes to a different prediction than the white box model. By applying both methods, the performance of a logistic regression can be increased by 1.5 percentage points by SHAP explanations of a gradient boosted machine on an example data set. This reduces the difference between the ROC-AUC of the black box model and the logistic regression by 68 percent.

Keywords: *Explainable Machine Learning, Feature Engineering, Gradient Boosting Machines, Shapley Additive Explanations, Logistic Regression, Black Box Models*

ZUSAMMENFASSUNG

Insbesondere im regulierten Umfeld erfreuen sich lineare Modelle nach wie vor einer großen Beliebtheit. Sie sind im Einsatz erprobt und auch die Aufsichtsbehörden mit deren Interpretation vertraut. Trotz jahrelanger Erfahrung im Umgang mit diesen Modellen, sind die Entwickler häufig nicht imstande, die Performance moderner maschineller Lernverfahren aus den Bereichen der Ensemble Modelle oder Neuronalen Netze zu erzielen.
Durch die Einführung der Datenschutzgrundverordnung (DSGVO) ist die Notwendigkeit für erklärbare und nachvollzierbare maschinelle Lernverfahren erneut gestiegen. Ziel muss es sein, ein möglichst erklärbares Modell mit einer möglichst hohen Vorhersagequalität zu entwickeln. Beide Anforderungen stehen sich - zumindest bei vielen Problemen - allerdings diametral gegenüber. Mittlerweile sind zahlreiche Methoden entstanden, die Entscheidungen eines Black-Box-Modells nachträglich zu erklären. Diese Methoden ermöglichen tiefe Einblicke in das Verhalten der Modelle, sind aber in Bezug auf ihre Interpretierbarkeit nicht mit intrinsisch erklärbaren Verfahren, wie einer logistischen Regression, vergleichbar.
In dieser Arbeit werden zwei Ansätze präsentiert, die bessere Trennschärfe der modernen, hochgradig komplexen Verfahren, mit dem erklärbaren Charakter einer logistischen Regression zu kombinieren. Diese Ansätze führen ein performance-orientiertes Feature Engineering für erklärbare Modelle durch und greifen dabei auf die Erklärungen eines leistungsfähigeren Black-Box-Ansatzes zurück. Dazu werden zwei verschiedene Methodiken entwickelt, die unabhängig voneinander, aber auch gemeinsam angewendet werden können. Je nach von Methodik und Problem, erhöht sich durch diese entweder die Performance des erklärbaren Modells, oder dessen Erklärbarkeit. In einigen Fällen ist sogar beides möglich.
Die erste Methodik bedient sich der strukturellen Unterschiede zwischen White- und Black-Box-Verfahren in Bezug auf die funktionale Form der daraus resultierenden Modelle. Daraus werden Transformationen des Merkmalsraums abgeleitet, die stets die Erklärbarkeit des Modells im Auge behalten. Die zweite Methodik betrachtet hingegen ausschließlich Beobachtungen, für die das Black-Box-Modell zu einer anderen Einschätzung wie das White-Box-Modell gelangt. Durch die Anwendung der beiden Verfahren kann die Performance einer logistischen Regression durch SHAP-Erklärungen einer Gradient Boosted Machine auf dem Beispieldatensatz um 1.5 Prozentpunkte erhöht werden. Die Differenz zwischen der ROC-AUC des Black-Box-Modells und der logistischen Regression sinkt dadurch um 68 Prozent.

Schlagwörter: *Erklärbarkeit, Explainable Machine Learning, Feature Engineering, Gradient Boosting Machines, Shapley Additive Explanations, Logistische Regression, Black-Box-Modelle*

Wenn du es einem Sechsjährigen nicht erklären kannst,
hast du es selbst nicht gut genug verstanden.
Albert Einstein

DANKSAGUNG

An dieser Stelle möchte ich mich bei all denjenigen Personen bedanken, die zum Gelingen dieser Masterarbeit beigetragen haben.

Zuerst gebührt mein Dank meiner Lebensgefährtin Laura Wieland, die mir während des gesamten Masterstudiums immer mit Rat und Tat zur Seite stand und stets Verständnis für meinen hin und wieder übertriebenen Ehrgeiz hatte.

Darüber hinaus gilt mein Dank meiner Familie - insbesondere meinen Eltern und meiner Schwester - die mich in den vergangenen Jahren durch viele positive Worte immer wieder motiviert und unterstützt haben.

Ein besonderer Dank gilt den Betreuern von Arvato Financial Solutions Dr. Karla Schiller und Markus Jabs für die intensiven Diskussionen und Anregungen, die die Masterarbeit bereichert haben. Ebenfalls möchte ich mich bei Patrick Dunau für die vielen kritischen Anregungen und das immer offene Ohr für Fragen meinerseits bedanken.

Allen Kolleginnen und Kollegen, sowie Kommilitoninnen und Kommilitonen, danke ich für die Geduld beim Korrigieren der sprachlichen und grammatikalischen Abarten, die mitunter beim Schreiben entstanden sind und das fachliche Feedback.

Zu Letzt ein besonders großes Dankeschön an Prof. Dr. Markus Döhring für die Betreuung der Masterarbeit. Mit viel Geduld, Interesse und Hilfsbereitschaft, zahlreichen interessanten Debatten und Ideen hat er maßgeblich dazu beigetragen, dass die Masterarbeit in dieser Form vorliegt.

Vielen Dank!

INHALTSVERZEICHNIS

ABBILDUNGSVERZEICHNIS

TABELLENVERZEICHNIS

Teil I

GRUNDLAGEN UND STAND DER FORSCHUNG

1 EINLEITUNG

1.1 MOTIVATION

Künstliche Intelligenz (KI) und maschinelles Lernen (ML) dringen seit einigen Jahren stärker in verschiedene Bereiche unseres Lebens vor. Bereits heute sind die Verfahren imstande, Vorhersagen mit hoher Genauigkeit zu treffen und so ihre Benutzer enorm zu unterstützen, wenn sie ihr menschliches Gegenüber dabei nicht sogar übertreffen. Gerade aus dem Bereich der Online- und Gesellschaftsspiele gibt es mittlerweile zahlreiche Beispiele, in denen intelligente Verfahren die menschliche Leistungsfähigkeit ausstechen z.B. [Onl19] oder [Sil+17]. Aber auch der Bereich der automatisierten Bilderkennung hat in den letzten Jahren große Fortschritte gemacht und schlägt mittlerweile selbst Experten in ihren Fachgebieten (vgl. hierzu [He+15] und [Est+17]).

Es existieren jedoch auch zahlreiche Szenarien, in denen eine genaue Prädiktion nicht alleiniges Kriterium für den Einsatz intelligenter Systeme ist, sondern Vertrauen in die Modelle sowie Transparenz die entscheidenden Argumente sind. In solchen Szenarien ist der Einsatz moderner maschineller Lernverfahren, wie z.B. tiefer neuronaler Netze oder Gradient Boosting Machines, mit Herausforderungen verbunden, sind diese Modelle doch inhärent komplex und gleichen oft einer Black-Box. Selbst für die Entwickler dieser Modelle und Experten ist es mitunter unmöglich, die abstrakten, gelernten Zusammenhänge nachzuvollziehen, geschweige denn in einfachen Worten wieder zu geben. An dieser Stelle müssen die Entscheidungen des Modells durch Erklärungen ergänzt werden. Autopiloten im Flugzeug, Zahlungsmittelsteuerung im Onlinehandel oder Betrugserkennung bei Kreditkartennutzung sind nur einige Beispiele [Dö+18].

Je mehr solcher künstlich intelligenten Black-Box-Systeme in unseren Alltag eindringen, desto stärker treten die Vorbehalte und Sorgen der Bevölkerung gegenüber diesen zu Tage. Spätestens seit dem Triumph der künstlichen Intelligenz AlphaGo über den besten menschlichen Go-Spieler Lee Sedol ist das Thema auch im Mainstream angekommen [Sil+17]. Seitdem berichten die Medien immer mehr über Durchbrüche in der Forschung und die scheinbar unendlichen Möglichkeiten von KI. Zunehmend mehren sich aber auch kritische Stimmen, die vor Gefahren und Risiken des unregulierten Einsatzes von intelligenten Systemen warnen. Diese Ambivalenz sorgt für große Verunsicherung in der Bevölkerung - wie zum Beispiel eine Studie des Instituts für Management- und Wirtschaftsforschung 2018 ergab [IMW18].

Hinzu kommt, dass moderne maschinelle Lernverfahren sehr anspruchsvoll sind. Sie sind mathematisch nur noch schwer zu durchschauen. Durch die hohe Dimensionalität der Probleme und den zunehmenden Grad an Abstraktion sind ihre Entscheidungen kaum mehr nachzuvollziehen. Das macht es dem Laien geradezu unmöglich, Chancen und Gefahren von KI ab- und einzuschätzen.

Die vielen offenen Fragen und Ängste in der Bevölkerung im Umgang mit künstlicher Intelligenz haben auch zu politischen und juristischen Konsequenzen geführt. Zusätzlich zu den bereits seit Jahrzehnten stark regulierten Unternehmen aus der Pharma- und Auskunftsteibranche geraten nun maschinelle Lernverfahren im Allgemeinen ins Auge der Regulierungsbehörden.

Mit der seit dem 25. Mai 2018 geltenden Datenschutz-Grundverordnung (DSGVO) wurde der Datenschutz auf europäischer Ebene vereinheitlicht. Artikel 13 legt fest, dass bei einer automatisierten Entscheidungsfindung die betroffene Person Anspruch auf *aussagekräftige Informationen über die involvierte Logik* hat. Darüber hinaus schreibt Artikel 12 vor, dass diese Informationen *in präziser, transparenter, verständlicher und leicht zugänglicher Form* bereitzustellen sind. Alle, die Machine Learning Modelle produktiv im Einsatz haben, deren Entscheidungen Einzelpersonen direkt betreffen, müssen auch Paragraph 22 berücksichtigen [GF17]. Dort heißt es unter anderem:

> [...] In jedem Fall sollte eine solche Verarbeitung mit angemessenen Garantien verbunden sein, einschließlich der spezifischen Unterrichtung der betroffenen Person und des Anspruchs auf direktes Eingreifen einer Person, auf Darlegung des eigenen Standpunkts, auf **Erläuterung der nach einer entsprechenden Bewertung getroffenen Entscheidung** sowie des Rechts auf Anfechtung der Entscheidung. [...]

Unter Juristen, Datenschützern und auch innerhalb der Machine Learning Community ist eine heiße Diskussion über die Auslegung des Paragraphen entbrannt. Während einige daraus ein *Recht auf Erklärbarkeit* einer jeden Entscheidung eines KI-Systems ableiten, sehen andere darin keine grundsätzliche Einschränkung im Einsatz von maschinellen Lernverfahren. Einige interessante Beiträge zu dieser Diskussion sind in diesem Artikel [Pia18] auf KDnuggets zusammengefasst. Völlig unabhängig von der endgültigen juristischen Beurteilung des Paragraphen, unterstreicht er die Wichtigkeit der Erklärbarkeit von maschinellen Lernverfahren.

Rein technisch betrachtet ist eine Erklärung der Vorhersage nicht nötig um gute Modelle zu trainieren. Die gute Performance des Modells auf unbekannten Daten ist ausreichend. Aber auch abseits von regulatorischen Anforderungen kann es sehr sinnvoll und hilfreich sein, diesen zusätzlichen Schritt in den Workflow einzubinden, bietet er doch eine ganze Reihe an Chancen [Gla18]:

1. **Verbesserung unserer Modelle** Ist bekannt, auf Grundlage welcher Feature das Modell Entscheidungen und Vorhersagen trifft, kann besser überprüft werden, ob die gelernten Regeln des Modells überhaupt sinnvoll sind. Dies verhindert, dass Modelle falsche Schlüsse aus den Daten ziehen. Ein mittlerweile sehr berühmtes Beispiel, wie aufschlussreich Modellerklärungen sein können, liefern Marco Ribeiro und seine Co-Autoren:

(a) Husky classified as wolf (b) Explanation

Abbildung 1.1: Erklärung eines Bildklassifizierers mittels LIME (Quelle: [RSG16b])

   Ziel des Ansatzes war es, Huskeys von Wölfen zu unterscheiden. Das Modell erzielte auf den Testdaten eine gute Performance. Erst durch die Erklärung wurde klar, dass das Modell nicht gelernt hatte, Wölfe von Huskys zu unterscheiden, sondern vielmehr auf Hintergrundinformationen vertraute. Dies lag in in einem Bias in den Trainingsdaten begründet. Das heißt, durch die Erklärungen wurde ein relevantes Problem bei der Auswahl der Merkmale erkannt.

2. **Vertrauen und Akteptanz** Nur wenn die Entscheidung eins Modells nachvollziehbar ist, kann diesem wirklich vertraut werden. Menschen sind rationale Wesen und stets auf der Suche nach kausalen Zusammenhängen. Einem Algorithmus, der nur seine Entscheidung mittelt, wird der Mensch immer mit einem gewissen Maß an Skepsis gegenüberstehen. Das trifft sowohl auf den Data Scientisten zu, der das Modell entwickelt, als auch auf die Menschen, die später das System einsetzen.

3. **Vorurteile und Modellbias verhindern** Die Basis für jedes ML-Modell sind die Daten, die in das Training des Modells einfließen. Daher ist es von besonders großer Bedeutung, in den Daten versteckte Vorurteile zu erkennen. Das Modell kann sonst leicht zu einer selbsterfüllenden Prophezeiung führen und bestehende Ungleichheiten verstärken. Erst durch Kenntnis der für das Modell relevanten Merkmale können eventuell problematische Feature entfernt werden. Cathy O'Neil hat eine ganze Reihe von Beispielen zusammengetragen, die aufzeigen, wie der

unreflektierte Einsatz von maschinellen Lernverfahren Ungleichheiten und Diskriminierung verstärken kann [O'N16]. Beispielsweise betrachtet sie ein Assistenzsystem, das amerikanischen Richtern helfen soll, ein geeignetes Strafmaß für Kriminelle festzulegen. Dieses System beurteilt, wie groß das Risiko einer zukünftigen erneuten Straftat ist, so dass der Richter dies bei seiner Entscheidung berücksichtigen kann. Wie eine große, auf ProPublica veröffentlichte Analyse zeigt, diskriminiert dieses System systematisch schwarze Straftäter/-innen [JAK16].

1.2 ZIEL DER ARBEIT

Gerade im regulativen Umfeld spielen traditionelle, statistische Verfahren nach wie vor eine große Rolle; sie sind zum einen leicht nachvollziehbar und andererseits im Einsatz erprobt. Moderne maschinelle Lernverfahren, wie neuronale Netze, Random Forests oder Gradient Boosting Machines, zeichnen sich durch eine besonders hohe Prädiktionsgüte im Vergleich zu klassischen statistischen Verfahren, wie der linearen oder logistischen Regression, aus. Dieser Gewinn an Performance geht allerdings auf Kosten der Erklärbarkeit der Modelle. Die Entscheidungen des Modells sind kaum bis gar nicht nachvollziehbar – auch für Experten. Daher wird häufig von Black-Box-Modellen und kontrastierend von erklärbaren White-Box-Modellen gesprochen.

White-Box-Ansätze versuchen häufig die Problemstellung mittels eines einfachen, i.d.R. linearen Modells zu lösen. Dies macht die Verfahren auf der einen Seite sehr transparent und erklärbar, trägt aber meist der Komplexität der zu lösenden Aufgabe nicht ausreichend Rechnung. Ein manuelles Integrieren von nichtlinearen Zusammenhängen und Interaktionen in die White-Box-Modelle ist zwar möglich und führt zu einer verbesserten Performance der Modelle, setzt allerdings entweder ein großes Maß an Erfahrung und Domänenwissen voraus, oder läuft auf systematisches Experimentieren via sogenannter Selektionsverfahren (z.B. Forward- bzw. Backwardselection) bzw. die aufwendige Analyse von Residuen- und Korrelationsplots hinaus.

Feature Engineering spielt folglich eine entscheidende Rolle beim Trainieren klassischer, maschineller Lernverfahren. Das soll keineswegs heißen, dass moderne Black-Box-Modelle nicht durch geschicktes Feature Engineering verbessert werden können. Allerdings findet hier beim Training eine implizite Merkmalsselektion in Kombination mit einer Detektion von Nichtlinearitäten und Interaktionen statt, so dass hier ein Teil des Engineerings bereits während des Modelltrainings erfolgt. Generell muss der geschickten Merkmalsauswahl, sinnvollen Transformationen oder Kategorisierungen ein großer Einfluss auf die Leistungsfähigkeit maschineller Lernverfahren zugeschrieben werden. Oder um es mit Pedros Domingos Worten zu sagen [Dom12]:

> *[. . .] some machine learning projects succeed and some fail. What makes the difference? Easily the most important factor is the features used.*

Da dieser Vorverarbeitungsschritt so großen Einfluss auf die Performance der Machine Learning Modelle hat, verbringen Entwickler dieser Modelle zwischen 60 und 80 Prozent ihrer Zeit mit dem Aufbereiten und Transformieren von Daten, wie eine Befragung von CrowdFlower 2016 ergab [Cro16].

Black-Box-Modelle können dem Entwickler einen Teil dieser aufwendigen Arbeit ersparen, sind sie doch imstande, beliebig komplexe, nichtlineare Entscheidungsfunktionen zu berechnen. Sie erkennen eigenständig nichtlineare Zusammenhänge und Interaktionen, priorisieren die Eingabe-Merkmale und können so oft sehr exakte Vorhersagen liefern. Dieser Gewinn an Performance geht allerdings meist mit einem Verlust der Erklärbarkeit der Modellentscheidung einher.

In den letzten Jahren sind zahlreiche Ansätze und Frameworks zur Erklärbarkeit dieser Black-Box-Verfahren entstanden, die in Kapitel 4 genauer untersucht werden. Diese Methoden ermöglichen tiefe Einblicke in das Verhalten der Modelle. Es lassen sich die wichtigsten Merkmale des Modells, deren Einfluss auf die Vorhersage und auch zweidimensionale Interaktionen zwischen Variablen extrahieren. Mittels solcher Erklärungen können detaillierte Erkenntnisse über die gelernten Zusammenhänge des Machine Learning Modells gewonnen werden.

> Ziel dieser Arbeit ist es, zu untersuchen, inwieweit sich die Performanz von White-Box-Modellen mittels Erklärungen von Black-Box-Modellen verbessern lässt, ohne dabei deren intrinsische Erklärbarkeit zu stark einzuschränken. Insbesondere soll untersucht werden, ob sich unter Zuhilfenahme der Erklärungen eines leistungsfähigen Black-Box-Modells das Feature Engineering für White-Box-Modelle vereinfachen lässt. Ziel ist es, dem Entwickler eines erklärbaren, klassischen Verfahrens Empfehlungen für die Auswahl von Merkmalen und deren geschickte Transformation bis hin zur Integration von Wechselwirkungen von Merkmalen zu geben.

1.3 AUFBAU DER ARBEIT

Zunächst werden in Kapitel 2 die für diese Arbeit relevanten Methoden aus den Bereichen Statistik und maschinelle Lernverfahren eingeführt. Dieses Kapitel ist an einigen Stellen bewusst sehr kurz gehalten, da die verschiedenen Verfahren mit all ihren Eigenschaften, Vor- und Nachteilen nicht im Zentrum der Arbeit stehen. Darüber hinaus ist ausreichend einschlägige Literatur vorhanden, die mehr oder weniger mathematisch formale und intuitive Einführungen in die Thematik liefert, auf die an den entsprechenden Stellen verwiesen sei. Die Konzepte zu Boosting Ensemblen - insbesondere

AdaBoost und Gradient Boosting - werden ausführlicher präsentiert.

Im Anschluss (Kapitel 3) folgt eine systematische Einführung in die Taxonomie und Eigenschaften des *Explainable Machine Learnings*, bevor in Kapitel 4 die in dieser Arbeit verwendeten Erklärbarkeitsansätze konkret erläutert werden. Neben der Analyse der mathematischen Grundlagen, stehen dabei auch die Einsatzmöglichkeiten, Stärken und Schwächen der Verfahren im Zentrum. Am Ende eines jeden Erklärbarkeitsansatzes findet sich eine kurze Zusammenfassung des vorgestellten Verfahrens. Diese beinhaltet eine heuristische Motivation des Verfahrens und erläutert im Anschluss Vor- und Nachteile.

Kapitel 5.1 widmet sich dem Feature Engineering mittels Erklärungen von Black-Box-Modellen. Zunächst werden die Annahmen und Einschränkungen erklärbarer maschineller Lernverfahren im Kontrast zu Black-Box-Modellen analysiert. Daraus wird in Kapitel 6 ein Ansatz entwickelt, der die Performance erklärbarer Modelle verbessert, ohne deren Erklärbarkeit zu vernachlässigen.

In Kapitel 7 folgt ein alternativer Ansatz zum Feature Engineering für White-Box-Modelle. Dabei beschränkt sich die Analyse auf jene Beobachtungen eines Datensatzes, für die es zu großen Vorhersagedifferenzen zwischen Black-Box- und White-Box-Verfahren kommt. Anhand der Erklärungen beider Modelle für diese Dateninstanzen können Gründe für die überlegene Performance des komplexen Modells erkannt und daraus Rückschlüsse für das Feature Engineering des intrinsisch erklärbaren Modells gezogen werden.

Im folgenden Kapitel 8 sollen diese verschiedenen Ansätze an einem Datensatz veranschaulicht werden. Dazu wird der Adult-Datensatz verwendet. Dies beinhaltet eine ausführliche Exploration des Datensatzes.

Den Schluss der Arbeit bildet Kapitel 9, das die Ergebnisse und Erkenntnisse der Arbeit nochmals zusammenfassen wird. Dabei werden auch Einschränkungen und Probleme erläutert. Der Ausblick soll zum einen auf weitere Anknüpfungspunkte zur Verbesserung des vorgestellten Ansatzes eingehen, zum anderen aber auch weiterreichende Fragestellungen beleuchten, die im Verlauf der Arbeit entstanden sind.

2

GRUNDLAGEN DES MASCHINELLEN LERNEN

2.1 LOGISTISCHE REGRESSION

Regressionsanalysen zur Modellierung der Verteilung abhängiger diskreter Variablen werden als logistische Regression oder Logit-Modell bezeichnet. Häufig werden dabei lediglich dichotome (binäre) Zielgrößen betrachtet und die Wahrscheinlichkeiten für das Eintreten einer der beiden Zielklassen modelliert. Das Vorgehen entspricht an vielen Stellen dem einer linearen Regression, trägt allerdings der Tatsache Rechnung, dass die zu modellierende Variable nicht kontinuierlich, sondern diskret ist. Die erklärenden Merkmale können dabei ein beliebiges Skalenniveau aufweisen, wobei diskrete Variablen mit mehr als zwei Ausprägungen in binäre Dummy-Variablen zerlegt werden.
Bei der Modellierung der Eintrittswahrscheinlichkeit einer diskreten Zielgröße mittels linearer Regression kommt es zu den folgenden Problemen:

- Das lineare Regressionsmodell gibt auch Werte < 0 und > 1 aus, was für die Modellierung einer Wahrscheinlichkeit unzweckmäßig ist.
- Die Residuenvarianz ist nicht homoskedastisch, d.h. die Varianz (σ_i^2) der beobachteten Größe einer Beobachtung i ist von ihrer Eintrittswahrscheinlichkeit (π_i) abhängig, da die abhängige Variable einer Bernoulliverteilung folgt.

Infrage kommende Funktionen zur Modellierung dieses Problems müssen einen auf das Intervall $[0, 1]$ beschränkten Wertebereich in Kombination mit einem streng monoton steigenden Funktionsverlauf vorweisen. Die Verwendung von Verteilungsfunktionen ist naheliegend, da diese beide Eigenschaften garantieren. Bei der Verwendung der logistischen Verteilungsfunktion

$$F(\eta) = \frac{\exp(\eta)}{1 + \exp(\eta)}$$

ergibt sich ein sogenanntes Logit-Modell. F wird häufig als Link- oder Kopplungsfunktion bezeichnet, da es im Folgenden ein Regressionsmodell mit den vorhergesagten Wahrscheinlichkeiten verknüpft. Ziel des logistischen Regressionsmodells ist es, mittels der logistischen Verteilungsfunktion den Effekt der erklärenden Merkmale x_i^T auf die Wahrscheinlichkeit für $Y_i = 1$ bzw. $Y_i = 0$ zu beschreiben (für $i = 1, ..., n$):

$$\begin{aligned}\pi_i = \mathrm{P}(Y_i = 1 \mid x_i^\top) &= \frac{\exp(\beta_0 + \mathbf{x}_i^\top \boldsymbol{\beta})}{1 + \exp(\beta_0 + \mathbf{x}_i^\top \boldsymbol{\beta})} \\ &= \frac{1}{1 + \exp(-(\beta_0 + \mathbf{x}_i^\top \boldsymbol{\beta}))} = F(\eta_i),\end{aligned}$$

wobei $\boldsymbol{\beta} = (\beta_1, ...\beta_n)^\top$.
Dementsprechend wird die Wahrscheinlichkeit für $Y = 1$ nicht direkt aus den erklärenden Variablen heraus (so wie bei der linearen Regression) modelliert, sondern indirekt über das sogenannte Logit. Das Logit ist die logarithmierte Chance für das Auftreten von $Y = 1$:

$$\eta_i = Logit(Y_i = 1 \mid x_i^\top) = \ln\left(\frac{\pi_i}{1-\pi_i}\right) = \beta_0 + \mathbf{x}_i^\top \boldsymbol{\beta} \tag{2.1}$$

Die Chance $\left(\frac{\pi_i}{1-\pi_i}\right)$ wird auch als Odds bezeichnet. Die Chancen können zwar Werte größer 1 annehmen, sind allerdings nach unten asymptotisch durch 0 beschränkt. Erst durch die Transformation in Logits ergibt sich der benötigte unbeschränkte Wertebereich. Das Logit fungiert an dieser Stelle als Kopplungsfunktion zwischen den Wahrscheinlichkeiten und dem linearen Schätzer. Dadurch wird es erst möglich, die Regressionsgleichung für das Problem in (2.1) zu schätzen. Die geschätzten Logits einer gegebenen Merkmalsmatrix X lassen sich folglich durch die Bestimmung der Regressionskoeffizienten berechnen.

Die Interpretation dieser Koeffizienten ist herausfordernd [HTF09]. Aufgrund des nichtlinearen und indirekten Einflusses der erklärenden Variablen auf die Eintrittswahrscheinlichkeit π_i können die geschätzten Koeffizienten $\boldsymbol{\beta}$, anders als beim linearen Regressionsmodell, nicht als direkte Einflussfaktoren auf die Wahrscheinlichkeit π_i interpretiert werden. Anhand der Vorzeichen der einzelnen $\beta_j \in \boldsymbol{\beta}$ ist lediglich die Wirkungsrichtung eines Merkmals abzulesen: Negative Vorzeichen korrespondieren mit einer Verringerung der Wahrscheinlichkeit für das Eintreten von $Y_i = 1$ bei steigenden Werten der erklärenden Variable und umgekehrt.
Darüber hinaus besteht die Möglichkeit, sog. Effektkoeffizienten durch Exponenzieren zu ermitteln; die Regressionsgleichung bezieht sich dadurch auf die Chancen und lässt Aussagen über die Stärke des Einflusses einer Merkmalserhöhung auf die Chance zu:

$$\frac{P(Y_i = 1 \mid x_j + 1)}{P(Y_i = 0 \mid x_j + 1)} = \frac{P(Y_i = 1)}{P(Y_i = 0)} = \exp(\beta_j)$$

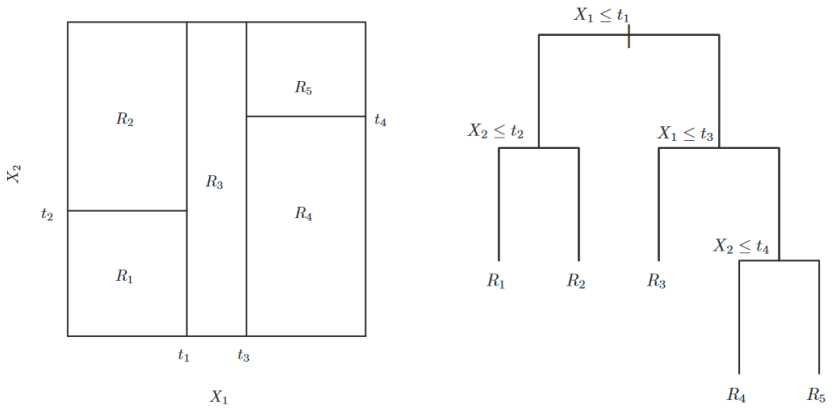
Eine Erhöhung des j-ten Merkmals um eine Einheit führt folglich zu einer Veränderung der Chance um den Faktor $\exp(\beta_j)$. Zusammengefasst können anhand der Regressionskoeffizienten die folgenden Schlussfolgerungen gezogen werden:

Koeffizient $\hat{\beta}$	**Chance** $\left(\frac{\pi_i}{1-\pi_i}\right)$	**Wahrscheinlichkeit** $P(Y = 1)$
$\hat{\beta} > 1$	steigt um $\exp(\hat{\beta})$	steigt
$\hat{\beta} < 1$	fällt um $\exp(\hat{\beta})$	sinkt
$\hat{\beta} = 1$	konstant	konstant

Tabelle 2.1: Interpretation von Regressionskoeffizienten der logistischen Regression

2.2 ENTSCHEIDUNGSBÄUME

Baumbasierte Verfahren sind eine weitere Möglichkeit, Regressions- oder Klassifikationsmodelle zu generieren. Diese partitionieren die Daten mehrfach anhand von Entscheidungsregeln für einzelne Merkmale. Durch die Aufteilung werden verschiedene *Regionen* des Datensatzes erzeugt, wobei jede Instanz zu einer Teilmenge gehört. Die endgültige Region einer Beobachtung wird auch als End- oder Blattknoten und die Zwischenmengen als innere Knoten oder Splitknoten bezeichnet. Das durchschnittliche Ergebnis der Trainingsdaten im Endknoten ergibt die Vorhersage des Baums für diesen Blattknoten. Häufig kommen bei der Berechnung von Entscheidungsbäumen binäre Splits zum Einsatz, es sind aber auch Konstruktionsalgorithmen mit Aufteilung in drei oder mehr Teilregionen möglich. Beispielsweise unterstützt der C4.5 Algorithmus auch nicht-binäre Splits [Sal94]. Im Folgenden sind mit Entscheidungsbäumen immer Algorithmen mit binären Aufteilungen des Datenraums gemeint. In einem zweidimensionalen Merkmalsraum können diese Regionen durch Rechtecke dargestellt werden, wie beispielsweise in Abbildung 2.1 (a).



(a) (b)

Abbildung 2.1: Aufteilung eines 2-D-Datenraums (a) durch den Entscheidungsbaum (b) [Jam+14]

Die Aufteilung erfolgt anhand der Feature und kann auch als Binär-Baum dargestellt werden. Dabei repräsentiert der Wurzel-Knoten stets den vollständigen Datensatz. Die Darstellung als Baum macht Entscheidungsbäume besonders leicht in der Interpretation, da jede Beobachtung auf ihrem Weg durch den Baum verfolgt werden kann. Ein beispielhafter Baum zur Aufteilung (a), findet sich in Abbildung 2.1 (b).

Hierarchische Klassifikationsmodelle, zu denen der Entscheidungsbaum zählt, zerlegen eine Datenmenge iterativ oder rekursiv mit dem Ziel, die Zielklassen im Rahmen des Lernens möglichst gut voneinander zu separieren. Ziel ist das Finden einer eindeutigen Klassenzuordnung für bestimmte

Eigenschaften in den Merkmalen. Die Zerlegung der Daten erfolgt mittels eines sogenannten Unreinheitsmaßes. Es wird der Split gewählt, der zur größten Verringerung der Unreinheit in den Daten führt. Um die Unreinheit eines Knoten bzw. Datenraums zu messen, gibt es verschiedene Maße, die hier nur kurz definiert werden. Eine ausführliche Darstellung findet sich unter anderem bei [Jam+14]. Solche Maße sind zum Beispiel:

- **Entropie** = $-\sum_{i=1}^{n} p_i \cdot \log_i(p_i)$
- **Gini** = $\sum_{i=1}^{n} p_i \cdot (1 - p_i) = 1 - \sum_{i=1}^{n} p_i^2$
- **Klassifikationsfehler** = $1 - max([p_1, ..., p_n])$

Dabei bezeichnet p_i die Wahrscheinlichkeit der i-ten Klasse, die einfach mittels des Anteils der Klasse i in dem betrachteten Datenraum geschätzt werden kann.

Mittels Entscheidungsbäumen ist es möglich einen Großteil der Beobachtungen der Trainingsmenge auswendig zu lernen. Dann bestehen die Blattknoten des Baums nur noch aus sehr wenigen Beobachtungen. Dadurch generalisieren die Bäume allerdings meist nicht mehr ausreichend und die Performance auf unbekannten Daten leidet. Um dieses Overfitting des Baums zu vermeiden, kann zum einen die Anzahl der Splits bzw. die Tiefe des Baumes beschränkt werden, oder sehr tiefe Bäume nachträglich mittels des sogenannten Prunings gestutzt werden. Dabei erfolgt ein erneutes Berechnen der Informationsgewinne eines Splits und das Kürzen von Verzweigungen, sollte deren Informationsgewinn zu gering sein. Oftmals wird hierfür nicht die Entropie oder Gini Unreinheit, sondern der Klassifikationsfehler als Maß für die Unreinheit verwendet. Durch Pruning ist der Baum besser imstande zu verallgemeinern und performt auf unbekannten Daten besser.

Die bisherigen Ausführungen fokussieren sich lediglich auf das Vorgehen bei Klassifikationsproblemen. Für Regressionsprobleme unterscheidet sich dieses allerdings kaum. Die Partitionen des Datenraums ergeben sich durch Suche nach Aufteilungen mit möglichst geringer Residuenquadratsumme. Im Gegensatz zu Klassifikationsbäumen entstehen in den Blättern keine Wahrscheinlichkeiten, sondern konstante Werte, die als Vorhersage für diese Partition der Daten herangezogen werden. Mehr zu Regressionsbäumen u.a. bei [HTF09].

2.3 ENSEMBLE MODELLE

Ein großes Problem verschiedener Klassifikationsmethoden ist ihre Anfälligkeit gegenüber Ausreißern bzw. Auffälligkeiten in den Daten. Ensemble Methoden versuchen dieses Problem zu lösen, indem die Ergebnisse mehrerer Klassifikatoren kombiniert werden. Dies führt oft zu einer Erhöhung der Vorhersagegenauigkeit.

Anstatt eines einzelnen Modells betrachten sie entweder verschiedene Modelltypen auf Basis der gleichen Datengrundlage oder verschiedene Realisationen eines Modelltyps auf unterschiedlichen Teilmengen der Trainingsdaten. Die einzelnen Teilmodelle werden häufig als *schwache Lerner* bzw. *weak learners* bezeichnet. Durch Kombination einzelner Vorhersagen der schwachen Modelle zu einem Ensemble entsteht eine einzige, robustere Vorhersage (z.B. durch Mittelung der einzelnen Vorhersagen). Die grundlegende Idee ist in Abbildung 2.2 dargestellt. Es lassen sich zwei verschiedene Vorgehen beim Trainieren von Ensemble Modellen unterscheiden: Bagging und Boosting Ensemble.

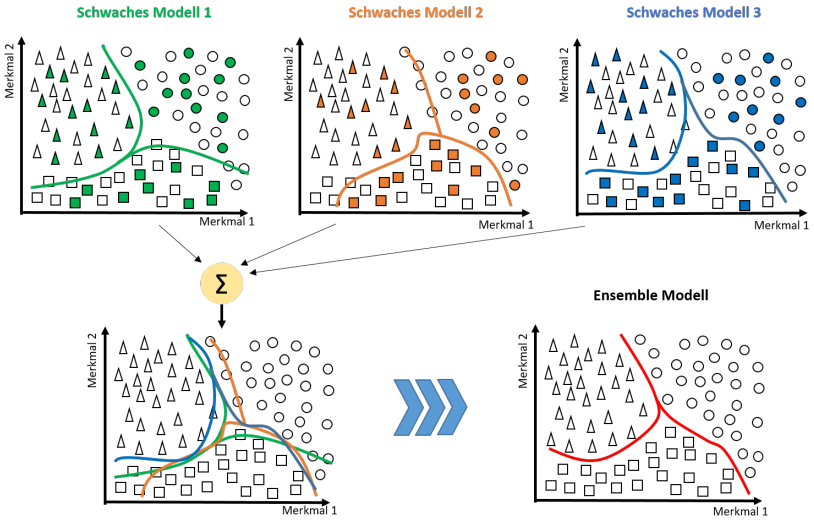


Abbildung 2.2: Darstellung der Funktionsweise von Ensemble Methoden (in Anlehnung an [ZM12])

2.3.1 *Bagging*

Bagging (kurz für Bootstrap Aggregation) und andere Resampling-Techniken finden Anwendung, um die Varianz in den Modellvorhersagen zu reduzieren. Bei Bagging handelt es sich um ein zweistufiges Verfahren: Zunächst werden wiederholt zufällige Stichproben aus einer Basisstichprobe generiert. Die erzeugten Stichproben haben dieselbe Länge wie die Basisstichprobe und werden allgemein Replikationen oder Resamples genannt. Da beim Bootstrapping keine Verteilungsannahmen nötig sind, ist eine Anwendung des Verfahrens auch in Fällen unbekannter theoretischer Verteilungen der Daten möglich [Kul+15].

Jedes Replikat wird zum Aufbau eines neuen Modells verwendet und die Modelle werden zu einem Ensemble zusammengefasst. Die Vorhersage des Ensembles ergibt sich durch Mittelung der Prognose jedes schwachen Lerners. Dies geschieht - abhängig vom Problemtyp - durch Bildung des

arithmetischen Mittels (Regression) oder ein Mehrheitsvotum (Klassifikation). Dadurch verringert sich die Varianz des Ensembles und es entstehen robustere Vorhersagen. Der Bias des Modells ergibt sich gerade als gewichteter Bias der einzelnen schwachen Lerner.

Bagging funktioniert am besten, wenn die einzelnen schwachen Lerner möglichst unkorreliert sind. Aufgrund des Bootstrappings sind die Replikate aber nie vollständig unabhängig, da ein Teil der Beobachtungen stets in mehrere Replikate eingeht. Wenn die verschiedenen schwachen Lerner jeweils eine Varianz von σ^2 haben und die paarweise Korrelation zwischen den Modellen ρ ist, dann hat das Ensemble eine (mittlere) Varianz der Prognose von $\rho \cdot \sigma^2 + \frac{(1-\rho)\cdot\sigma^2}{T}$, wobei T die Anzahl der schwachen Lerner bezeichne. Das heißt, je höher die Korrelation zwischen den Modellen, desto ähnlicher werden die Vorhersagen und der reduzierende Effekt des Ensembles auf die Varianz geht verloren. Besonders bei Merkmalen mit großer prädiktiver Power - einer hohen Korrelation mit der Zielvariablen - entstehen durch Bootstrapping stark korrelierte schwache Lerner, da diese Merkmale in jedem Replikat enthalten sind. Abbildung 2.3 zeigt das Vorgehen exemplarisch, wobei sich die Vorhersage hier als arithmetisches Mittel der einzelnen Lerner ergibt.

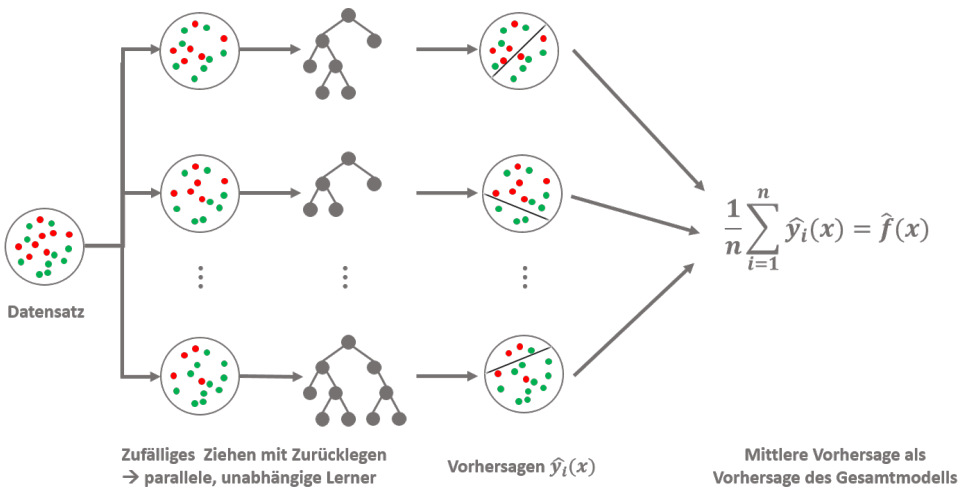


Abbildung 2.3: Vorgehen bei Bagging Ensembles

Random Forests stellen eine sehr populäre Erweiterung des allgemeinen Bagging Ansatzes dar und adressieren dieses Problem. Als schwache Lerner werden dabei nur Entscheidungsbäume herangezogen. Bei diesem Ansatz wird in jedem Knoten eines Baums eine Teilmenge m der p möglichen Variablen aus den Daten zufällig ausgewählt und nur diese Merkmale als möglicher Split herangezogen. Diese zufällige Auswahl von Variablen reduziert die Ähnlichkeit von Bäumen, die aus verschiedenen Bootstrap-Replikationen stammen. Zudem sind zwei Bäume aus einer Replikation sehr wahrscheinlich unterschiedlich [Zho12].

2.3.2 *Adaptives Boosting (AdaBoost)*

Im Gegensatz zum Bagging erfolgt beim Boosting das Training der Lerner nicht parallel, sondern sequentiell anhand des Fehlers des vorangegangenen Lerners. Anstatt die Replikate für das Training durch unabhängiges, zufälliges Ziehen mit Zurücklegen zu erzeugen, werden einzelne Beobachtungen aus dem Originaldatensatz anhand der Vorhersage des vorherigen Lerners gewichtet. Diejenigen Datenpunkte, die den größten Vorhersagefehler verursacht haben, sollen häufiger gezogen werden als Beobachtungen, die zuvor richtig prognostiziert wurden. Zusätzlich ordnet der AdaBoost-Algorithmus jedem resultierenden Modell während des Trainings ein Gewicht zu. Einem Lerner mit gutem Klassifizierungsergebnis auf den Trainingsdaten wird ein höheres Gewicht (α) zugewiesen, als einem Lerner mit geringer Performance. Zur Bewertung eines neuen Lerners muss der Algorithmus dessen Fehler im Auge behalten. Die Vorhersage des Gesamtmodells ergibt sich als gewichtetes Mittel der einzelnen Modelle, wobei Modelle mit einer guten Performance auf den Trainingsdaten einen größeren Einfluss haben. Eine zusammenfassende Darstellung findet sich in Abblidung 2.4.

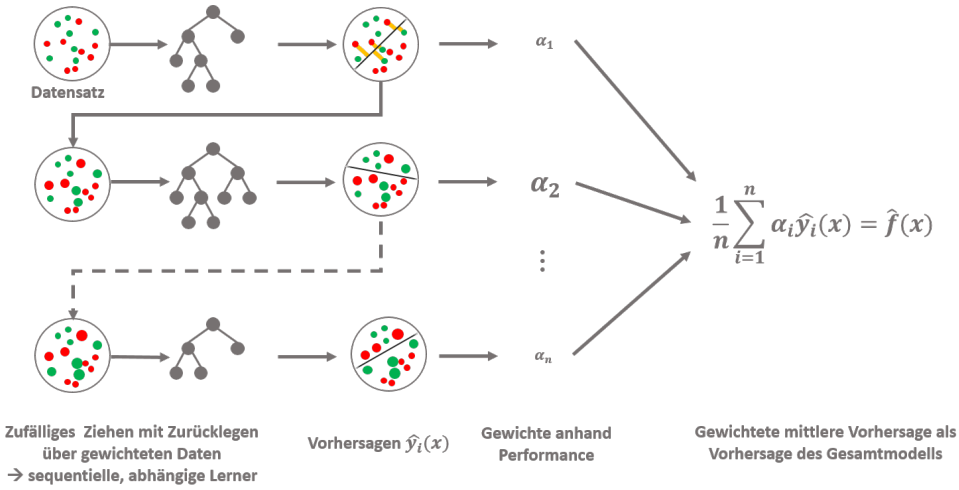


Abbildung 2.4: Vorgehen bei Ada-Boosting Ensembles

Friedman [Fri01] interpretiert dieses Vorgehen als sequentielle Minimierung einer exponentiellen Fehlerfunktion - häufig auch Lossfunktion. Warum diese Interpretation sehr nützlich ist, wird im nächsten Abschnitt zu Gradient Boosting deutlich werden. Bei Hastie et.al. [HTF09] findet sich eine Darstellung der Zusammenhänge zwischen diesem Loss und binären Klassifikationsproblemen. Im Falle einer binären Klassifikation ist die Fehlerfunktion gegeben durch:

$$E = \sum_{n=1}^{N} \exp\left(-t_n \cdot f_m(x_n)\right) \tag{2.2}$$

wobei $f_m(x_n) \in \{-1, 1\}$ ein Klassifizierer ist, der sich aus einer Linearkombination von (schwachen) Lernern der Form $y_i(x)$ zusammensetzt. Also

$$f_m(x) = \frac{1}{2} \sum_{l=1}^{m} \alpha_l \cdot y_l(x)$$

und $t_n \in \{-1, 1\}$ die bivariate Zielgröße des binären Klassifikationsproblems.
Ziel ist die Minimierung der Fehlerfunktion E. Zur Minimierung stehen der Gewichtungsfaktor α_l und die Parameter des *schwachen* Lerners $y_l(x)$ zur Verfügung. Durch das sequentielle Vorgehen ist im m-ten Iterationsschritt davon auszugehen, dass die vorangegangenen Klassifizierer $y_1(x), \ldots, y_{m-1}(x)$ und deren Gewichte $\alpha_1, \ldots, \alpha_{m-1}$ bekannt sind. Das heißt, die Minimierung von E erfolgt bezüglich α_m und y_m. Dann können alle Terme, die im m-ten Iterationsschritt bereits bestimmt sind, zu einem bzgl. der Minimierung konstanten Faktor $w_n^{(m)}$ zusammenfasst werden:

$$\begin{aligned} E &= \sum_{n=1}^{N} \exp\left(-t_n \cdot f_{m-1}(x_n) - \frac{1}{2} t_n \alpha_m y_m(x_n)\right) \\ &= \sum_{n=1}^{N} w_n^{(m)} \cdot \exp\left(-\frac{1}{2} t_n \alpha_m y_m(x_n)\right). \end{aligned}$$

Im Folgenden sei

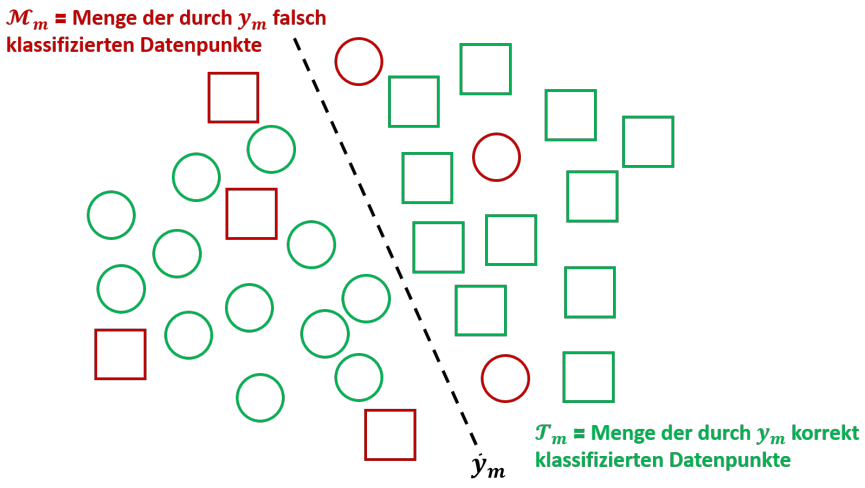


Abbildung 2.5: Beispiel zur Bezeichnung der Mengen $\mathcal{T}_m$ und $\mathcal{M}_m$

- $\mathcal{T}_m$ = Menge der durch $y_m(x)$ korrekt klassifizierten Datenpunkte und
- $\mathcal{M}_m$ = die Menge der durch $y_m(x)$ falsch klassifizierten Datenpunkte

(vgl. Abbildung 2.5) und definiere

$$\mathbb{1}_{\{y_m(x_n) \neq t_n\}} := \begin{cases} 1 & \text{für } y_m(x_n) \neq t_n \qquad n \in \mathcal{M}_m \\ 0 & \text{sonst.} \end{cases}$$

Dann vereinfacht sich der exponentielle Fehler zu

$$E = \sum_{n \in \mathcal{T}_m} w_n^{(m)} e^{-\frac{1}{2}\alpha_m} + \sum_{n \in \mathcal{M}_m} w_n^{(m)} e^{-\frac{1}{2}\alpha_m} \tag{2.3}$$

$$= e^{-\frac{1}{2}\alpha_m} \cdot \left(\sum_{n=1}^{N} w_n^{(m)} - \sum_{n=1}^{N} w_n^{(m)} \cdot \mathbb{1}_{\{y_m(x_n) \neq t_n\}} \right) +$$

$$e^{-\frac{1}{2}\alpha_m} \cdot \left(\sum_{n=1}^{N} w_n^{(m)} \cdot \mathbb{1}_{\{y_m(x_n) \neq t_n\}} \right) \tag{2.4}$$

$$= (e^{\frac{1}{2}\alpha_m} - e^{-\frac{1}{2}\alpha_m}) \cdot \left(\sum_{n=1}^{N} w_n^{(m)} \cdot \mathbb{1}_{\{y_m(x_n) \neq t_n\}} \right) + e^{-\frac{1}{2}\alpha_m} \cdot \sum_{n=1}^{N} w_n^{(m)}. \tag{2.5}$$

Wobei in (2.4) der Zusammenhang, dass richtig klassifizierte Beobachtungen gerade die Differenz zwischen allen Beobachtungen und den falsch klassifizierten Beobachtungen beschreiben, die Berechnung erleichtert [HTF09].
Zur Minimierung von E bzgl. α_m ergibt sich die notwendige Bedingung als

$$\frac{\partial E}{\partial \alpha_m} = \frac{1}{2}(e^{\frac{1}{2}\alpha_m} + e^{-\frac{1}{2}\alpha_m}) \left(\sum_{n=1}^{N} w_n^{(m)} \cdot \mathbb{1}_{\{y_m(x_n) \neq t_n\}} \right) - \frac{1}{2} e^{-\frac{1}{2}\alpha_m} \sum_{n=1}^{N} w_n^{(m)}$$
$$\stackrel{!}{=} 0,$$

was zum folgenden Zusammenhang führt:

$$\epsilon_m := \frac{\sum_{n=1}^{N} w_n^{(m)} \cdot \mathbb{1}_{\{y_m(x_n) \neq t_n\}}}{\sum_{n=1}^{N} w_n^{(m)}} = \frac{e^{-\frac{1}{2}\alpha_m}}{e^{-\frac{1}{2}\alpha_m} + e^{-\frac{1}{2}\alpha_m}} = \frac{1}{1 + e^{\alpha_m}} \tag{2.6}$$

ϵ_m ist sozusagen eine Abschätzung des Vorhersagefehlers des m-ten *schwachen* Lerners, setzt dieses doch gerade die Anzahl der (gewichteten) falsch klassifizierten Beobachtungen ins Verhältnis zur Gesamtanzahl aller (gewichteten) Beobachtungen. In Abhängigkeit von dieser Fehlerabschätzung ergibt sich das Gewicht α_m des m-ten schwachen Klassifizieres über den folgenden Zusammenhang aus Gleichung (2.6):

$$\epsilon_m = \frac{1}{1 + e^{\alpha_m}} \iff \alpha_m = \ln \frac{1 - \epsilon_m}{\epsilon_m}.$$

Die neuen Gewichte für die Beobachtungen w_n^{m+1} in Abhängigkeit vom Vorhersagefehler resultieren aus

$$w_n^{m+1} := w_n^m \cdot \exp\left(\alpha_m \cdot \mathbb{1}_{\{y_m(x_n) \neq t_n\}}\right). \tag{2.7}$$

Prognostiziert der m-te Lerner eine Beobachtung korrekt ($\mathbb{1}_{\{y_m(x_n) \neq t_n\}} = 0$), findet keine Anpassung des Gewichts statt d.h. $w_n^{m+1} = w_n^m$. Anderenfalls garantiert (2.7) eine Vergrößerung des Gewichts. Zu guter Letzt bleibt die Frage, wie die Ergebnisse zu einer gemeinsamen Vorhersage des Modells

kombiniert werden. Dies ist einfach das gewichtete Mittel der einzelnen schwachen Lerner

$$Y_M(x) = sign\left(\sum_{m=1}^{M} \alpha_m \cdot y_m(x)\right).$$

Dadurch ist sichergestellt, dass Modelle mit einer guten Performance auf den Testdaten einen größeren Einfluss auf die Modellvorhersage haben als Modelle mit einer eher schlechten Vorhersagequalität. Eine algorithmische Darstellung (Algorithmus 3) findet sich im Anhang.

Bevor im nächsten Abschnitt ein weiterer - meist sehr performanter - Boosting Algorithmus vorgestellt wird, noch einige Anmerkungen:

Dieses Vorgehen funktioniert für beliebige Klassifikatoren und lässt sich auch leicht auf Regressionsprobleme übertragen. Vergleiche dazu [HTF09]. Anders als bei Bagging verringert das Mitteln mehrerer schwacher Modelle bei Boosting Ensemblen nicht nur die Varianz des Gesamtmodells, sondern auch dessen Bias. Dies liegt im sequentiellen Verringern der Fehlerfunktion (2.2) begründet. Dadurch kommt der Wahl einer geeigneten Anzahl an schwachen Modellen im Ensemble bzw. eines angemessenen Abbruchkriteriums eine große Bedeutung zu, da es sonst auch hier zu Overfitting des Modells auf den Trainingsdaten kommen kann. Dieser Zusammenhang macht AdaBoost auch sehr anfällig für verrauschte Daten.

2.3.3 *Gradient Boosting*

Gradient Boosting gehört - wie der Name schon vermuten lässt - ebenfalls zu den Boosting Ensemble Methoden. Das heißt, auch hier wird beim Training eines jeden Lerners der Fehler des vorangegangen schwachen Modells berücksichtigt. Auch wenn im vorangegangen Abschnitt AdaBoost zunächst über die Anpassung der Gewichte eines jeden Datenpunktes motiviert wurde, ist bereits bei der Definition der exponentiellen Fehlerfunktion (vgl. Gleichung (2.2)) klar geworden, dass dieses Vorgehen auf das sequentielle Minimieren einer geeigneten Fehler-/Loss-Funktion hinausläuft. Diese Idee wird im Folgenden, ausgebaut und es entsteht dadurch ein sehr leistungsfähiges Verfahren. Anstatt die Beobachtung nach jeder Iteration neu zu gewichten, ergeben sich die neuen schwachen Modelle direkt aus den Vorhersagefehlern des vorangegangenen Modells, was zu einer sequentiellen Minimierung der Fehlerfunktion des Ensembles führt. Die grundlegende Idee ist in Abbildung 2.6 dargestellt:

Im Grunde handelt es sich bei Boosting um eine additive Erweiterung einer bestehenden Menge von einfachen Basisfunktionen. Im Falle von AdaBoost fungierten die $y_m(x)$ als Basisfunktionen: Dabei wurde stets das Ziel verfolgt, die zuvor definierte (exponentielle) Fehlerfunktion über den Trai-

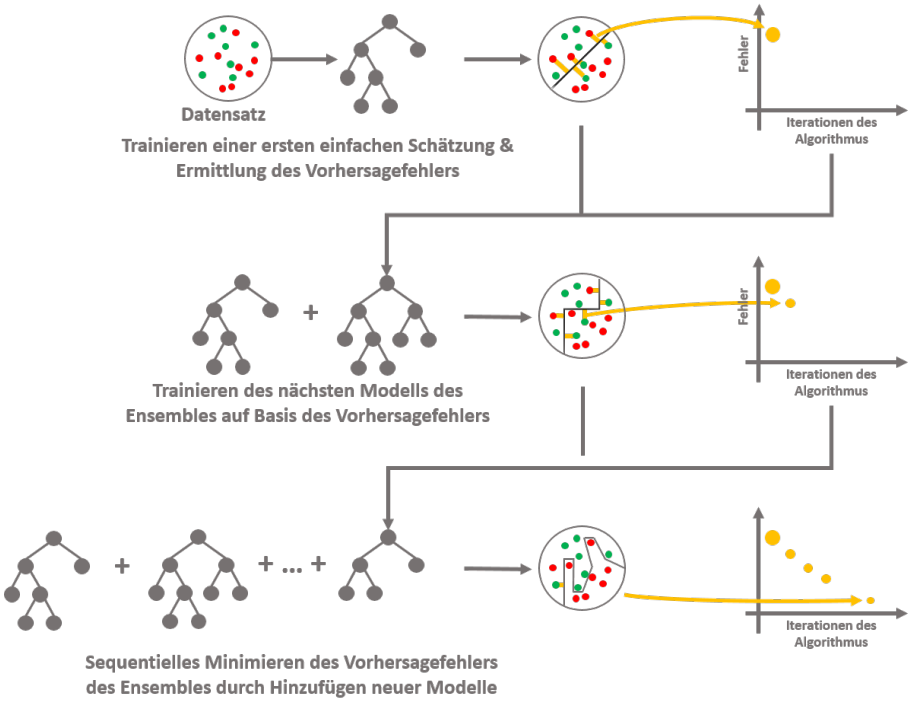


Abbildung 2.6: Vorgehen bei Gradient Boosting Ensembles

ningsdaten zu minimieren. Diese Idee wird nun verallgemeinert. Dazu sei zunächst das folgende Problem beschrieben:

Gegeben seien eine endliche Menge an Beobachtungen $\{y_i, x_i\}$ für $i \in 1, \dots N$, eine Basisfunktion $b(x, \phi)$ mit Parameterset ϕ und eine Lossfunktion $L(y, f(x))$. Gesucht werden ein Parameter ϕ und ein Gewichtungsfaktor w, der die Fehlerfunktion

$$\{w_m, \phi_m\}_1^M = \underset{\{w_m, \phi_m\}_1^M}{\operatorname{argmin}} \sum_{i=1}^{N} L\left(y_i, \sum_{m=1}^{M} w_m b(x_i; \phi_m)\right) \tag{2.8}$$

minimiert. Wahrscheinlichkeitstheoretisch entspricht dies der Minimierung der stichprobenbasierten Schätzung des Erwartungswertes der Loss-Funktion L. Da eine direkte Lösung dieses Optimierungsproblems oft nicht möglich ist, wird ein sequentieller *greedy-Ansatz* verfolgt und die Fehlerfunktion durch Hinzufügen neuer Basisfunktionen zum Ensemble $f_m(x)$ minimiert:

$$\{w_m, \phi_m\} = \underset{w, \phi}{\operatorname{argmin}} \sum_{i=1}^{N} L\left(y_i, f_{m-1}(x) + w \cdot b(x, \phi)\right)$$

Diese (einfache) Vorgehen findet sich untenstehend in algorithmischer Form:

1 Initialisiere $f_0(x) = 0$.
2 **for** $m = 1, \ldots, M$ **do**
3 Trainiere einen Klassifizierer für die Trainingsdaten:

$$(w_m, \phi_m) = \underset{w,\phi}{\operatorname{argmin}} \sum_{i=1}^{N} L\left(y_i, f_{m-1}(x_i) + w \cdot b(x_i, \phi)\right)$$

Setze $f_m(x) = f_{m-1}(x) + w_m \cdot b(x; \phi_m)$
4 **end**

Algorithmus 1 : Additive Vorwärtsminimierung

Aus der Numerik sind eine Reihe von Verfahren bekannt, um Optimierungsprobleme der Art von Gleichung 2.8 zu lösen, u.a. das Gradientenverfahren. Dieses sucht nach einer Lösung der Form

$$f^*(x) = \sum_{m=0}^{M} f_m(x),$$

wobei $f_0(x)$ eine initiale Schätzung und $f_m(x)$ für $m = 1, \ldots M$, inkrementelle Funktionen oder Schritte darstellen, die durch das Optimierungsverfahren bestimmt werden. Im Falle des Gradientenverfahrens sind die Schritte wie folgt definiert:

$$f_m(x) = -\rho_m g_m(x)$$

mit

$$g_m(x) = \left[\frac{\partial L(y, f(x))}{\partial f(x)}\right]_{f(x)=f_{m-1}(x)} \qquad \text{und}$$

$$\rho_m = \underset{\rho}{\operatorname{argmin}} \qquad L(f_{m-1}(x) - \rho g_m(x))$$

g_m ist der Gradient der Loss-Funktion - die Richtung des stärksten Anstiegs der Funktion - und ρ_m bezeichnet die Schrittweite des Verfahrens. Die Minimierung der Loss-Funktion erfolgt anschaulich durch einen Schritt der Größe ρ in Richtung des größten Abstiegs (negativen Anstiegs). Anschließend wird erneut der steilste Abstieg berechnet und dieses Vorgehen wiederholt durchgeführt, bis sich der Fehler numerisch nicht mehr verändert. Mehr zum Gradientenverfahren findet sich unter anderem bei [NWoo].

Nun zum eigentlichen Problem aus Gleichung 2.8 zurück: Gegeben ein Ensemble f_{m-1}, können die $w \cdot b(x_i, \phi)$ als bester *greedy*-Schritt zur stichprobenbasierten Schätzung von $f^*(x)$ betrachtet werden, da die Richtung des

Schritts $b(x_i, \phi)$ durch die gewählte Klasse an Basisfunktionen (z.B. Entscheidungsbäume) eingeschränkt ist. Der stichprobenbasierte negative Gradient

$$-g_m(x_i) = -\left[\frac{\partial L(y, f(x_i))}{\partial f(x_i)}\right]_{f(x_i)=f_{m-1}(x_i)}$$

würde die beste (im Sinne von steilstem Abstieg) Richtung im Datenraum an der Stelle $f_{m-1}(x)$ anzeigen. Allerdings ist dieser Gradient dann nur an den Stellen x_i für $i = 1, \dots N$ in den (Trainings-)Daten definiert und wäre nicht auf andere Beobachtungen (z.B. die Testdaten) generalisierbar. Das würde zu einer schlechten Performance des Ensemble-Modells auf unbekannten Daten führen.
Durch geschickte Wahl der Basisfunktionen $b_m(x; \phi)$ ist eine Generalisierung möglich. Dazu werden die Basisfunktionen betrachtet, die für alle Beobachtungen x_i für $i = 1, \dots N$, am ähnlichsten zum negativen Gradienten $-g_m$ im jeweiligen Iterationsschritt m sind [Fri01]. Es werden folglich die Basisfunktionen gewählt, die über die komplette Verteilung des Datensatzes am stärksten mit dem negativen Gradienten korrelieren. Dazu ist lediglich das deutlich einfachere Optimierungsproblem

$$\phi'_m = \underset{w,\phi}{\operatorname{argmin}} \sum_{i=1}^{N} \left(-g_m(x_i) + w \cdot b(x_i, \phi)\right)^2 \tag{2.9}$$

zu lösen. Der daraus resultierende *Gradient* in Form einer Basisfunktion $b(x, \phi')$ wird anstelle des tatsächlichen Gradienten verwendet. Die Bestimmung der Schrittweite ρ erfolgt analog via

$$\rho_m = \underset{\rho}{\operatorname{argmin}} \sum_{i=1}^{N} L\left(y_i, f_{m-1}(x_i) + \rho \cdot b(x_i, \phi')\right). \tag{2.10}$$

Schlussendlich wird das Ensemble-Modell um

$$f_m(x) = f_{m-1}(x) + \rho_m b(x, \phi')$$

erweitert.

Nochmals kurz zusammengefasst: Anstatt *schwache Lerner* auf den tatsächlichen Beobachtungen zu trainieren, werden negative Gradienten als neue Zielgröße ($\tilde{y}_i = -g_m(x_i)$ für $i = 1, \dots N$) betrachtet - diese wird häufig als Pseudoresiduum bezeichnet [HTF09]. Diese Transformation des Problems ermöglicht es, das sehr anspruchsvolle Optimierungsproblem aus Gleichung 2.8 durch eine Kleinste-Quadrate-Minimierung (2.9), die mittels Algorithmus 1 lösbar ist, gefolgt von der Minimierung eines Parameters (2.10) zu ersetzen. Dieses Vorgehen lässt sich auf alle Loss-Funktionen übertragen, was ein großer Vorteil ist. Ein Algorithmus zur Berechnung findet sich im Anhang (Algorithmus 4??).

Boosting Ensemble werden meist mit Klassifikations- oder Regressionsbäumen als Basisfunktion bzw. schwache Lerner eingesetzt. Dabei spielt die Tiefe und Anzahl der einzelnen Entscheidungsbäume eine große Rolle in Hinblick auf die Performance des Ensembles. Erfahrungen zeigen, dass maximale Tiefen zwischen drei und acht zu guten Ergebnissen führen. Ausführlichere Erläuterungen dazu finden sich bei [HTF09] und [Fri01].

Darüber hinaus kann der Gradient Boosting Algorithmus durch Regularisierung bzw. *Shrinkage* und Subsampling noch weiter verbessert werden. Bei ersterem wird in jedem Gradientenschritt die Schrittweite mit einem Faktor zwischen 0 und 1 multipliziert (häufig als Lernrate bezeichnet). Konvergieren die Vorhersagen langsam in Richtung tatsächlicher Werte, werden Beobachtungen, die bereits *nahe* an der tatsächlichen Zielgröße liegen, zu größeren Blättern zusammengefasst (aufgrund der festen Baumgröße), was zu einem Regularisierungseffekt führt.
Die meisten Gradienten Boosting Algorithmen bieten die Möglichkeit vor jeder Boosting-Iteration ein Zeilen- und Reihen-Sampling durchzuführen. Wie bei Bagging Ensemblen führt dieses Vorgehen zu verschiedenen Bäumen, was in der Regel die Generalisierbarkeit und damit auch die Performance des Ensembles verbessert. Dieses Vorgehen wird oft als *Stochastic Gradient Boosting* bezeichnet [Fri02].

3

SYSTEMATIK DES ERKLÄRBAREN MASCHINELLEN LERNENS

Es existiert (bis jetzt) keine mathematische oder einheitliche Definition von Interpretierbarkeit bzw. Erklärbarkeit. Im Kontext von Machine Learning und künstlicher Intelligenz ergeben sich eine Reihe von Motiven für den Einsatz von interpretierbaren Modellen. Auch wenn Versuche existieren, die verschiedenen Motivationen dahinter greifbar zu machen und eine Definition zu liefern, ist Interpretierbarkeit nach wie vor ein schwammiger Begriff. Eine ausführliche Darstellung der Motive, Herausforderungen und Schwächen im Umgang mit der Interpretierbarkeit von maschinellen Lernverfahren findet sich bei Lipton [Lip18]. Im Folgenden wird die Definition von Miller aus [Mil17] Grundlage der Arbeit sein:

> Interpretierbarkeit ist der Grad, in dem ein Mensch die Ursache einer Entscheidung verstehen kann.

Je größer die Interpretierbarkeit eines Modells, desto einfacher ist es nachzuvollziehen warum bestimmte Entscheidungen oder Vorhersagen getroffen wurden. Ein Modell ist besser erklärbar als ein anderes, wenn seine Entscheidungen für den Menschen leichter zu verstehen sind als die Entscheidungen des anderen Modells. Wie genau die Nachvollziehbarkeit einer Entscheidung, und somit auch die Erklärbarkeit eines Modells zu messen ist, ist nach wie vor eine unbeantwortete Forschungsfrage. Bis heute existiert kein einheitliches Vorgehen. Häufig greifen Autoren auf Experimente mit einer kleinen Gruppe von Probanden zurück, die die Entscheidungen verschiedener Modelle nachvollziehen sollen. Mehr zu den Problemen im Kontext von Evaluation und Motiven finden sich bei [Lip18].

Zunächst werden einige grundlegende Begriffe im Zusammenhang mit Explainable Machine Learning (ExML) eingeführt und einige Anforderungen an bzw. Eigenschaften einer Erklärung dargestellt. Anschließend werden die Erklärbarkeitsansätze, welche im weiteren Verlauf der Arbeit Relevanz besitzen, im Detail vorgestellt. In dieser Arbeit werden die Begriffe interpretierbar und erklärbar austauschbar verwendet, allerdings immer zwischen der **Erklärbarkeit bzw. Interpretierbarkeit** des Modells und der **Erklärung** zur Vorhersage zu einer konkreten Beobachtung unterschieden.

3.1 TAXONOMIE DES ERKLÄRBAREN MASCHINELLEN LERNENS

Methoden zur Erklärbarkeit von maschinellen Lernverfahren sind ein aktuelles Forschungsfeld und nahezu täglich erscheinen Paper, die sich mit dieser Thematik befassen. Mittlerweile hat sich eine Vielzahl von Ansätzen

etabliert, so dass es mitunter schwierig ist, einen Überblick zu behalten. Ein Großteil der Verfahren lässt sich aber anhand der folgenden Kriterien kategorisieren:

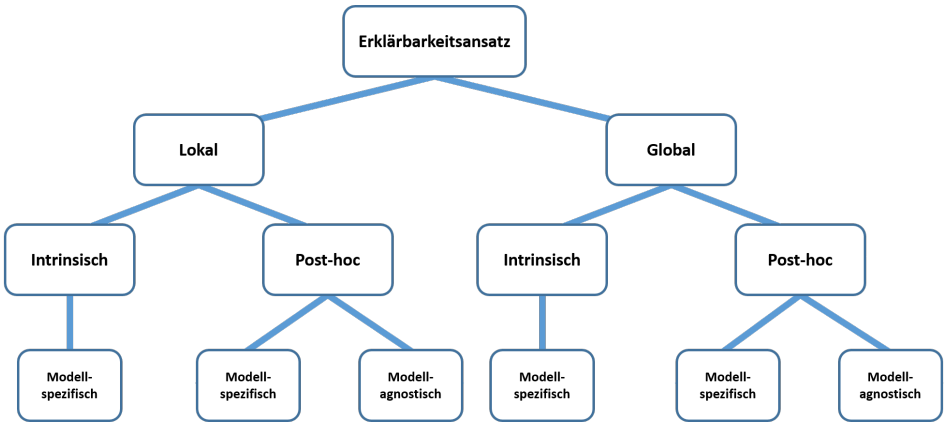


Abbildung 3.1: Taxonomie der Erklärbarkeitsansätze

Intrinsisch vs. Post-hoc

Grundsätzlich kann auf zwei verschiedenen Wegen ein interpretierbares Modell erzeugt werden: Entweder durch Wahl eines maschinellen Lernverfahrens, das bereits aufgrund seiner einfachen Struktur als interpretierbar gilt. Hier wird die Interpretierbarkeit durch Einschränkung der Komplexität des maschinellen Lernmodells erreicht. Daher werden solche Modelle als intrinsisch erklärbar bezeichnet. Beispiele für solche Modelle sind kurze Entscheidungsbäume oder einfache lineare Modelle.
Alternativ kann ein (von seiner Struktur meist deutlich komplexeres) Modell durch den Einsatz von Analysemethoden nach dem Training (post-hoc) erklärt werden. Sensitivitätsanalysen oder Feature Importance sind Beispiele für einfache post-hoc-Ansätze, aber auch Verfahren wie LIME (Kapitel 4.4) oder Shapley Values (Kapitel 4.5.1) fallen in diese Kategorie. Post-Hoc-Methoden können auch auf intrinsisch interpretierbare Modelle angewendet werden. So ist die Berechnung der Feature Importance zum Beispiel auch für Entscheidungsbäume oder die logistische Regression möglich.

Modell-spezifisch vs. Modell-agnostisch

Modell-spezifische Erklärbarkeitsansätze sind auf bestimmte Modellklassen beschränkt. Die Interpretation von Regressionsgewichten in einer logistischen Regression ist modell-spezifisch, genau wie intrinsisch interpretierbare Modelle per Definition immer modell-spezifisch sind. Tools, die z.B. die spezielle Struktur eines Entscheidungsbaumes ausnutzen und dadurch nur für diese verwendet werden können, sind modell-spezifisch. Modell-agnostische Werkzeuge können hingegen auf jedem maschinellen Lernverfahren nach dessen Training (post-hoc) angewendet werden. Bei vielen agnostischen Methoden werden meist Paare von Inputfeaturen und der prognostizierte Output betrachtet. Per Definition können diese Methoden nicht auf

Eigenschaften des Modells wie Gewichte oder Strukturinformationen zurückgreifen, allerdings kann es die Berechnung von post-hoc-Erklärungen deutlich beschleunigen, diese Informationen in die Approximation mit einzubeziehen.

Lokale vs. globale Erklärung
Eine weitere Möglichkeit, die verschiedenen Ansätze zur Interpretierbarkeit zu unterscheiden, ist der Bereich, der durch das Verfahren erklärt wird: Erklärt die Interpretationsmethode eine einzelne Vorhersage oder das gesamte Modellverhalten? Erstere Verfahren werden als lokale, zweitere als globale Erklärungen bezeichnet. Das sind die beiden Extreme, zwischen denen sich Erklärungen eines Modells immer bewegen. Lokale Verfahren legen das Modell sinnbildlich unter das Mikroskop und erklären die Vorhersage für einen einzelnen Datenpunkt. Globale Verfahren betrachten hingegen das Modell als Ganzes und versuchen dessen Verhalten in seiner Gesamtheit zu erfassen. Wie auch im Labor, gibt es dazwischen noch zahlreiche Zwischenstufen, auf denen Analysen des Verhaltens eines maschinellen Lernverfahrens möglich sind [Molsu]. Durch Anwenden eines lokalen Erklärungsansatzes auf jede Beobachtung im Datensatz ist es möglich, auch Rückschlüsse auf das globale Verhalten des Modells zu ziehen.

3.2 EIGENSCHAFTEN UND ANFORDERUNGEN AN ERKLÄRUNGEN

Ziel ist es, die Vorhersagen eines maschinellen Lernverfahrens zu erklären. Um dies zu erreichen, werden oft Methoden eingesetzt, die selbst wiederum ein Algorithmus sind. Eine solche Erklärung stellt in der Regel eine Beziehung zwischen den Input-Features und dem Modell-Output her und präsentiert diese in einer für Menschen mehr oder weniger leicht verständlichen Art und Weise. Allerdings verrät dies nichts über die Qualität der Erklärung. Abhängig vom Motiv hinter dem Einsatz von Erklärbarkeitsverfahren sind verschiedene Eigenschaften der Erklärung wünschenswert. Die folgende Liste mit Anforderungen orientiert sich an [RSB18] und unterscheidet zwischen Eigenschaften des Erklärbarkeitsansatzes und den Eigenschaften einer einzelnen Erklärung (für eine einzelne Beobachtung). Bei vielen dieser Anforderungen ist allerdings unklar, wie eine quantitative Bestimmung dieser Eigenschaften aussehen könnte [Lip18].

3.2.1 *Eigenschaften von Erklärbarkeitsansätzen*

Expressive Power beschreibt die *Sprache* oder Struktur der Erklärungen, die durch eine Methode erzeugt wird. Eine Erklärungsmethode kann WENN-DANN-Regeln, Entscheidungsbäume, eine Visualisierung, natürliche Sprache oder etwas anderes erzeugen.

Transluzenz beschreibt, wie stark sich die Methode zur Erklärung des Modells mit jenem und seinen Parametern auseinandersetzt. Beispielsweise sind Erklärungsmethoden, die auf intrinsisch interpretierbaren Modellen, wie dem lineare Regressionsmodell, basieren, hochtransluzent. Manipuliert eine Methode hingegen lediglich den Input eines Modells und beobachtet den Output, so hat diese keine Transluzenz [Molsu].

Portabilität drückt die Bandbreite an maschinellen Lernverfahren aus, bei denen die Erklärbarkeitsmethode eingesetzt werden kann. Dieser Begriff ist eng mit der Transluzenz verbunden: Methoden mit geringer Transluzenz haben eine höhere Portabilität, da sie das Modell wie eine Black-Box behandeln. Methoden, die nur für einen speziellen Modelltyp funktionieren, haben eine geringe Portabilität.

Algorithmic Complexity misst den Rechenaufwand, der zur Berechnung einer Erklärung nötig ist. Gerade für globale Erklärungen und große Datensätze ist es wichtig diese Eigenschaft zu berücksichtigen, da hier die Rechenzeit schnell unverhältnismäßig groß werden kann.

3.2.2 *Eigenschaften der einzelnen Erklärungen*

Genauigkeit: Meist handelt es sich bei Erklärbarkeitsansätzen selbst um Modelle, anhand derer Beobachtungen prognostiziert werden können. Die Genauigkeit quantifiziert die Übereinstimmung der Erklärung mit der beobachteten Zielgröße auf bislang unbekannten Instanzen. Daher muss der Genauigkeit besonders viel Beachtung geschenkt werden, wenn das erklärende Modell anstelle des Originalmodells Verwendung finden soll. In der Regel ist die Genauigkeit des erklärenden Modells allerdings durch die Genauigkeit des Black-Box-Modells beschränkt, welches es zu erklären versucht.

Fidelity/Treue ist wohl eine der wichtigsten Eigenschaften einer guten Erklärung. Werden die Vorhersagen eines Black-Box-Modells durch den Erklärbarkeitsansatz gut approximiert, hat die Erklärung eine hohe *fidelity*. Im Gegensatz zur Genauigkeit misst die Treue nicht die Übereinstimmung der Erklärung mit der Zielgröße, sondern mit der Vorhersage des Black-Box-Modells.

Meist ist es sinnvoll, zwischen *local fidelity* (wie gut nähert das Verfahren eine Teilmenge an Beobachtungen) und *general fidelity* zu unterscheiden. Einige Erklärungen bieten nur lokale Treue (z.B. LIME), teilweise sogar nur für eine einzelne Dateninstanz (z.B. Shapley Values). Erklärbarkeitsverfahren mit einer niedrigen lokalen Fidelity sind quasi nutzlos, da sie nicht einmal imstande sind, eine einzelne Beobachtung zu approximieren.
Genauigkeit und Treue sind eng miteinander verbunden. Wenn das Black-Box-Modell eine hohe Genauigkeit und die Erklärung eine hohe Fidelity

aufweisen, hat die Erklärung auch eine hohe Genauigkeit.

Konsistenz: Wie stark unterscheiden sich Erklärungen verschiedener Black-Box-Verfahren, die für dieselbe Aufgabe trainiert wurden und ähnliche Vorhersagen liefern? Wenn die Erklärungen sehr ähnlich sind, sind die Erklärungen sehr konsistent. Hohe Konsistenz muss nicht immer wünschenswert sein. Es wäre durchaus denkbar, dass die beiden Black-Box-Modelle unterschiedliche Feature verwenden, aber ähnliche Vorhersagen treffen.

Stabilität: Während mittels der Konsistenz Erklärungen mehrere Black-Box-Modelle zueinander ins Verhältnis gesetzt werden, vergleicht Stabilität die Erklärungen für ähnliche Beobachtungen innerhalb eines Modells. Hohe Stabilität bedeutet, dass geringfügige Abweichungen in den Merkmalen einer Instanz die Erklärung nicht wesentlich verändern (es sei denn natürlich, dass dadurch auch die Vorhersage stark verändert wird). Ein Mangel an Stabilität weist auf eine große Varianz des erklärenden Modells hin. Diese Eigenschaft wird manchmal auch als **Robustheit** des Verfahrens bezeichnet.

Verständlichkeit beschreibt, wie gut ein Mensch die Erklärung verstehen kann. Diese Anforderung ist gleich aus mehreren Gründen problematisch. Zum einen ist sie stark von den Adressaten einer Erklärung abhängig. Ein Machine Learning Engineer mag eine Erklärung als verständlich bezeichnen, die für den Mitarbeiter der Fachabteilung nur unwesentlich an Komplexität gegenüber dem Originalmodell verloren hat. Zum anderen ist der Begriff Verständlichkeit schwierig zu definieren und noch schwieriger zu messen. Ideen zur Messung der Verständlichkeit umfassen unter anderem die Größe der Erklärung (z.B. Anzahl der Entscheidungsregeln) oder die Prüfung, wie gut Menschen das Verhalten des maschinellen Lernmodells aus den Erklärungen vorhersagen können.

Gewissheit: Spiegelt die Erklärung die Sicherheit wider, mit der ein ML-Modell eine Entscheidung trifft? Beispielsweise kann sich ein Klassifizierer bei seiner Prognose sehr sicher sein (in Form einer hohen Wahrscheinlichkeit für eine Klasse). Eine Erklärung kann diese Information widerspiegeln, muss sie aber nicht.

Grad der Wichtigkeit: Wie gut spiegelt die Erklärung die Bedeutung einzelner Feature oder Teile der Erklärung wider? Wird beispielsweise deutlich, welche Variable die wichtigste für die Vorhersage war?

Repräsentativität: Wie viele Instanzen umfasst eine Erklärung? Erklärungen können das gesamte Modell abdecken (z.B. Interpretation von Gewichten in einem linearen Regressionsmodell) oder nur eine einzelne Vorhersage (z.B. Shapley Values) erklären.

4 ERKLÄRBARKEITSANSÄTZE

Es folgt eine ausführliche Darstellung der in dieser Arbeit verwendeten Erklärbarkeitsansätze. Dabei werden zum einen nur post-hoc Verfahren betrachtet, da es das Ziel der Arbeit ist, intrinsisch erklärbare, einfache Verfahren, wie zum Beispiel eine logistische Regression, durch den Einsatz von post-hoc Erklärungen komplexerer Modelle zu verbessern.
Zum anderen beschränkt sich die Betrachtung hauptsächlich auf modell-agnostische Verfahren, da die Trennung von maschinellem Lernverfahren und dessen Erklärung eine Reihe an Vorteilen mit sich bringt [RSG16a]. Modell-agnostische Methoden sind gegenüber modell-spezifischen deutlich flexibler. Die Erklärung funktioniert unabhängig vom zugrunde liegenden maschinellen Lernverfahren, was auch den Vergleich von Erklärungen für mehrere Modelle verschiedenen Typs ermöglicht und die Entscheidung für oder gegen ein Modell unabhängig vom erklärenden Modell macht [ŠK14].

Bei [Molsu] findet sich ein erster Versuch einer übersichtlichen Darstellung verschiedener Ansätze zur Erklärbarkeit, auf die an dieser Stelle verwiesen sei.

Neben der intuitiven Motivation und der mathematischen Formulierung der Verfahren werden zu jedem Verfahren auch Beispiele betrachtet. Diese beziehen sich im Folgenden meist auf den Adult-Datensatz. In Kapitel 8 wird der Datensatz ausführlich vorgestellt und analysiert. An dieser Stelle reicht es aus, das Problem kurz zu beschreiben: Es stehen 12 verschiedene sozio-ökonomische Merkmale, wie z.B. Alter, Beruf, Beziehungsstatus usw. über circa 32.500 amerikanische Staatsbürger zur Verfügung. Das Ziel ist es, anhand dieser Feature vorherzusagen, ob das Einkommen des Staatsbürgers größer oder kleiner-gleich 50.000$ ist.

4.1 PARTIAL DEPENDENCE PLOTS (PDP)

Partial Dependence Plots (PDP) - oder partielle Abhängigkeitsdiagramme - zeigen, wie sich die Vorhersage eines Modells basierend auf den Ausprägungen einer (manchmal auch zweier) unabhängiger Variablen ändert, während die Auswirkungen aller anderen unabhängigen Variablen herausgemittelt werden [HTF09]. Dadurch ist eine grobe Analyse des funktionellen Zusammenhangs zwischen einer Variablen und dem Target möglich - insbesondere ob dieser linear, monoton, quadratisch oder deutlich komplexer ist. PDPs mit zwei unabhängigen Variablen sind besonders dann von großem Nutzen, wenn es um die Darstellung komplexer Interaktionen zwischen zwei unabhängigen Variablen geht. In Kapitel 4.6 wird eine weitere Möglichkeit prä-

sentiert, mittels der diese Interaktionen sichtbar gemacht werden können. Partial Dependence Plots sind wie folgt definiert:

$$\hat{f}_{x_S}(x_S) = E_{x_C}\left[\hat{f}(x_S, x_C)\right] = \int \hat{f}(x_S, x_C) d\mathbb{P}(x_C) \tag{4.1}$$

Wobei x_S die Merkmale bezeichne, deren Einfluss auf die Modellprognose untersucht wird. Dies sind folglich jene Merkmale, die im PDP dargestellt werden. Alle anderen Variablen, die im Modell $\hat{f}$ Verwendung finden, sind durch x_C repräsentiert. Die Merkmalsvektoren x_S und x_C bilden zusammen den gesamten Merkmalsraum X ab. Die partielle Abhängigkeit zwischen Feature und Vorhersage wird durch Marginalisieren der Modellvorhersage über die Verteilung der Merkmale aus x_C bestimmt. Dadurch wird der Zusammenhang zwischen den Merkmalen in der Menge S und dem vorhergesagten Output sichtbar. So entsteht eine Funktion, die nur von den Featuren in S abhängt. Vereinfacht gesagt, berechnet ein PDP die durchschnittliche Modellvorhersage über alle möglichen Ausprägungen der Merkmale in S.

Zur Approximation des obenstehenden Integrals (4.1) kommen Monte Carlo Simulationen zum Einsatz. Anstatt des exakten Integrals wird dazu die folgende Näherung berechnet:

$$\hat{f}_{x_S}(x_S) = \frac{1}{n}\sum_{i=1}^{n} \hat{f}(x_S, x_C^{(i)}).$$

$\hat{f}_{x_S}$ erklärt für gegebene Werte der Merkmale in S den durchschnittlichen marginalen Effekt auf die Vorhersage des Modells. $x_C^{(i)}$ sind hierbei die tatsächlichen Ausprägungen der nicht betrachteten Variablen im Datensatz, wobei n die Anzahl an Beobachtungen im Datensatz beschreibt.

Für Klassifizierungsprobleme zeigen die PDPs die Wahrscheinlichkeit für eine bestimmte Klasse bei unterschiedlichen Ausprägungen für die Merkmal(e) in S an.
Bevor ein beispielhafter PDP analysiert und Vor- und Nachteile der Methode erläutert werden, noch eine Anmerkung zum Umgang mit kategoriellen Variablen: Die Berechnung der partiellen Abhängigkeit ist hier sehr einfach. Für jede der Kategorien ergibt sich die PDP-Schätzung durch Erzwingen der gleichen Ausprägung des Merkmals über alle Dateninstanzen. Bei vier Ausprägungen der kategoriellen Variable ergeben sich dann vier verschiedene Werte. [Molsu].

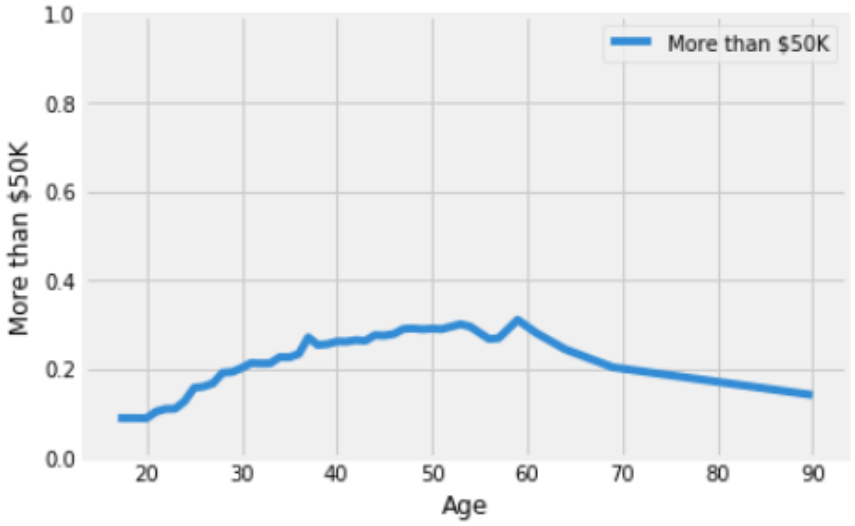


Abbildung 4.1: Partial Dependence Plot für Age (Adult-Datensatz)

In Abbildung 4.1 ist der PDP des Merkmals Age dargestellt: Bis zu einem Alter von 55 Jahren ist ein fast monotoner Anstieg der im Mittel vorhergesagten Wahrscheinlichkeit zu beobachten. Ab einem Alter von 60 Jahren fällt die Wahrscheinlichkeit wieder monoton ab.

Zusammenfassung Partial Dependence Plots

Mittels Partial Dependence Plots kann der marginale Einfluss einer erklärenden Variable im Black-Box-Modell auf die Vorhersage des Modells dargestellt werden. Dazu muss die durchschnittliche Modellvorhersage gegen die verschiedenen Ausprägungen des Merkmals aufgetragen werden.

Vorteile:

- Die Idee hinter PDPs ist sehr intuitiv - auch für Laien. Es handelt sich dabei lediglich um eine Analyse der Änderung der Vorhersage, wenn sich die Ausprägung eines Merkmals systematisch verändert.
- Durch Manipulieren eines Features und Messen der Vorhersageänderung erfolgt eine Analyse der Kausalzusammenhänge zwischen Merkmal und Vorhersage [ZH19]. Allerdings handelt es sich lediglich um die Kausalität aus Sicht des Modells. Diese muss nichts mit den Abläufen in der realen Welt zu tun haben.
- Wenn das Merkmal, dessen PDP berechnet werden soll, nicht mit den anderen Featuren korreliert, ergibt sich eine exakte Darstellung über den mittleren Einfluss dieses Merkmals auf die Vorhersage.
- PDPs sind modell-agnostisch. Das ermöglicht den einfachen Austausch des zugrunde liegenden Black-Box-Modells.

Nachteile:

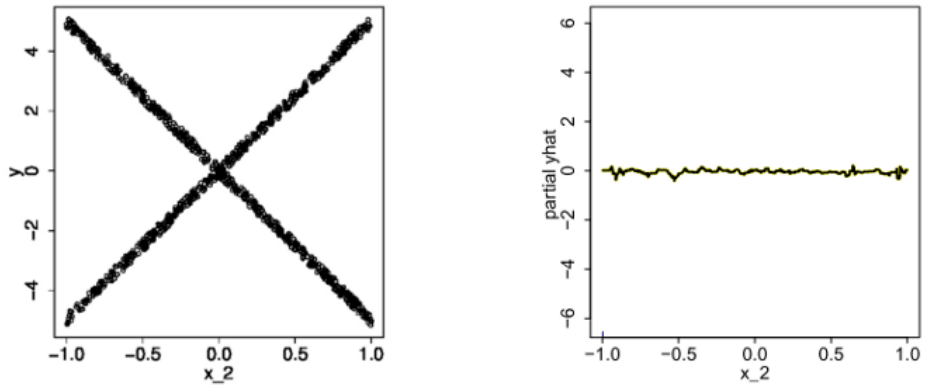
- Eine implizite Annahme der Berechnung ist die der Unabhängigkeit des zu erklärenden Merkmals von den anderen Featuren. Dies ist eine Konsequenz aus der Betrachtung des unbedingten Erwartungswerts $E_{x_C}\left[\hat{f}(x_S, x_C)\right]$. Dadurch entstehen unter Umständen Beobachtungen, die in der Realität nur mit sehr geringer Wahrscheinlichkeit beobachtet werden und fachlich sinnlos sind.
- Es lassen sich nur eindimensionale oder zweidimensionale Beziehungen zwischen Merkmalen darstellen. Diese Beschränkung ist mathematisch zwar nicht gegeben - die Anzahl an Merkmalen in der Menge S ist nicht auf 2 beschränkt - höher-dimensionale Modelle können allerdings nicht mehr auf verständliche Art und Weise visualisiert werden.
- Ohne Berücksichtigung der Verteilung des Merkmals können PDPs zu falschen Interpretationen bzw. Bedenken in Bezug auf das Verhalten des Modells führen, da Bereiche des Plots große Aufmerksamkeit erhalten, die nur sehr wenige Beobachtungen enthalten und daher für das Verständnis des Modellverhaltens von untergeordneter Priorität sind. Daher sollte die Merkmalsverteilung bei PDPs immer entsprechend berücksichtigt werden.
- Da ausschließlich durchschnittliche Randverteilungen betrachtet werden, kann es sein, dass heterogene Effekte nicht erfasst werden können. Dies führt schlimmstenfalls zu einem völlig falschen Bild vom Einfluss des Features; insbesondere wenn starke Interaktionen zwischen den Merkmalen vorliegen bzw. der Einfluss eines Merkmals stark streut. Vergleiche dazu das einführende Beispiel aus dem folgenden Kapitel (Abbildung 4.2).

4.2 INDIVIDUAL CONDITIONAL EXPECTATION (ICE)

Bei ICE-Plots (Individual Conditional Expectation) wird hingegen die Änderung der Vorhersage bei Manipulation der Merkmalsausprägung für jede Instanz dargestellt. Anstatt die durchschnittliche Änderung der Vorhersage zu betrachten, wird die Sensibilität jeder einzelnen Beobachtung auf Änderungen der Merkmalsausprägung analysiert.
So gesehen handelt es sich bei PDPs um eine globale Methode, den Einfluss eines Merkmals auf die Vorhersage zu betrachten; fokussiert sie sich doch auf den Gesamtdurchschnitt. ICE-Plots hingegen sind ein lokales Vorgehen, da dabei die Veränderung pro Instanz betrachtet wird [Gol+13]. Der PDP ergibt sich als Durchschnitt aller Linien eines ICE-Diagramms.

Um den ICE-Plot einer Instanz i zu berechnen, wird erneut der gesamte Merkmalsraum in x_C konstant fixiert und für die Variablen in x_S neue, synthetische Beobachtungen erzeugt, die durch das Black-Box-Modell f prognostiziert werden. Dies erfolgt mittels systematischen Ersetzens der Merkmalsausprägung durch Werte aus einem Raster der Feature in S. Als Ergebnis ergibt sich eine Punktmenge für jede Instanz mit dem Merkmalswert aus dem Raster und den jeweiligen Vorhersagen.

Bereits bei der Beschreibung der Nachteile des Partial Dependence Plots wurde auf die Probleme im Kontext von Wechselwirkungen zwischen Merkmalen hingewiesen. In der Regel liefern PDPs in diesen Fällen einen falschen Eindruck vom Einfluss einer Variable. Das folgende Beispiel illustriert diesen Effekt:



(a) Scatterplot of Y versus X_2 (b) PDP

Abbildung 4.2: Interaktion wird durch PDP nicht erkannt (Quelle: [Gol+13])

Durch die Interaktion mit einer anderen Variable ist der Zusammenhang zwischen X_2 und der Vorhersage $\hat{Y}$ nicht eindeutig, wie der Scatterplot 4.2 (a) zeigt. Die bei der Berechnung des PDPs durchgeführt Mittelung, sorgt allerdings dafür dass sich die Interaktionseffekte gerade aufheben, so dass

in Abbildung 4.2 (b) der Eindruck entsteht, dass X_2 im Durchschnitt kaum Einfluss auf die Vorhersage $\hat{Y}$ des Modells hat.

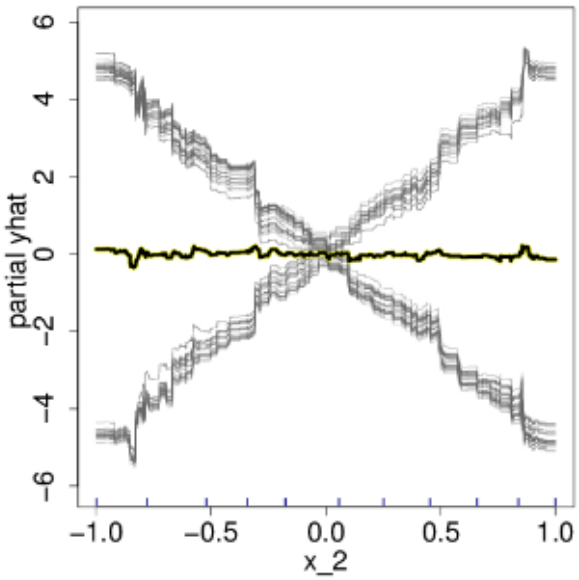


Abbildung 4.3: ICE-Plot für das Beispiel aus Abbildung 4.2 (Quelle: [Gol+13])

In Abbildung 4.3 ist der ICE-Plot des Merkmals X_2 aus Abbildung 4.2 dargestellt. Die dicke schwarze Linie repräsentiert die PD-Kurve. Es ist zu erkennen, dass die ICE-Linien den tatsächlichen Zusammenhang zwischen X_2 und der Zielgröße $\hat{Y}$ deutlich besser beschreiben.
Im Falle von Interaktionen bietet der ICE-Plot deutlich mehr Einblicke in den tatsächlichen Zusammenhang zwischen Merkmalsausprägung und Modell.

Unter Umständen ist es schwer zu untersuchen, ob sich die ICE-Kurven zwischen Individuen unterscheiden, da sie auf einem unterschiedlichen Niveau der Vorhersage beginnen. Eine einfache Lösung besteht darin, die Kurven an einem bestimmten Punkt im Merkmalsraum zu zentrieren und nur die Differenz in der Vorhersage zu diesem Punkt anzuzeigen. Meist wird die mittlere Vorhersage am unteren Ende der Feature-Ausprägungen als Referenzpunkt gewählt (dies entspricht gerade dem PDP-Wert an dieser Stelle). Die resultierende Darstellung wird als zentrierter ICE-Plot (c-ICE) bezeichnet. Formal ergeben sich die Linien für jede Instanz $i \in \mathbb{N}$ dann als

$$\hat{f}_{cent}^{(i)} = \hat{f}^{(i)} - \hat{f}(x^a, x_C^{(i)}),$$

wobei x^a den Referenzpunkt bezeichnet. Die zentrierten ICE-Plots erleichtern den Vergleich der Kurven einzelner Beobachtungen. Dies kann nützlich sein, wenn nicht die absolute Änderung eines vorhergesagten Wertes Zen-

trum der Analyse ist, sondern die Differenz der Vorhersage im Vergleich zu einem Festpunkt des Merkmalsbereichs [Molsu] dargestellt werden soll.

Zusammenfassung Individual Conditional Expectation Plots

Individual Conditional Expectation Plots stellen eine lokale Erweiterung von PDPs dar. Anstatt der mittleren Änderung der Vorhersage betrachten sie die Änderung für jede einzelne Instanz. Dadurch entsteht ein detaillierter Einblick in den Einfluss eines Merkmals - gerade im Kontext von Interaktionen. Zentrierte ICE-Plots bieten eine Möglichkeit, die Vorhersagen verschiedener Beobachtungen leichter miteinander zu vergleichen.

Vorteile:

- ICE-Kurven sind noch intuitiver zu verstehen als PDPs. Eine Linie repräsentiert die Vorhersagen für eine Beobachtung, wenn das Merkmal von Interesse variiert.
- Im Gegensatz zu partiellen Abhängigkeitsgrafiken können ICE-Kurven heterogene Beziehungen durch Interaktionen aufdecken.
- ICE ist modell-agnostisch. Das ermöglicht den einfachen Austausch des zugrunde liegenden Black-Box-Modells.

Nachteile:

- ICE-Kurven können nur ein Merkmal sinnvoll darstellen, da zwei Merkmale das Zeichnen mehrerer überlagerter Flächen erfordern würden.
- ICE-Kurven leiden unter dem gleichen Problem wie PDPs: Wenn das betrachtete Merkmal mit anderen Merkmalen korreliert ist, können ungültige Beobachtungen entstehen.
- Wenn zu viele ICE-Kurven gezeichnet werden, kann die Darstellung überfüllt sein und relevante Informationen überdecken.

4.3 GLOBALE MERKMALSRELEVANZ

In den meisten Fällen sind die Inputmerkmale eines Machine Learning Modells nicht gleichermaßen relevant. Oft haben nur wenige von ihnen einen wesentlichen Einfluss auf die Vorhersage; die verbleibenden könnten genauso gut nicht in das Training einbezogen werden, ohne dabei einen signifikanten Verlust an Performance zu erleiden [Bre+84]. Daher ist es oft nützlich, die globale relative Bedeutung oder den Beitrag jeder Variable zur Vorhersage zu ermitteln. Es gibt verschiedene Ansätze, die globale Wichtigkeit eines Merkmals zu definieren. Einige davon sind modell-agnostisch, andere sind auf baumbasierte Verfahren beschränkt.

4.3.1 *Gini Wichtigkeit*

Ein großer Vorteil der in Kapitel 2.3 vorgestellten baumbasierten Ensemble Methoden liegt in der implizit durchgeführten Merkmalsselektion des Verfahrens. Oft wird nur eine kleine Teilmenge besonders trennscharfer Merk-

male für die Vorhersage herangezogen, was gerade bei hochdimensionalen Problemen zu einer überlegenen Performance führt [Bre04]. Das Ergebnis dieser impliziten Merkmalsauswahl kann durch die Gini Wichtigkeit (auch Gini Importance oder Mean Decrease in Impurity) dargestellt werden und wurde 1985 von Breiman et al. als allgemeiner Indikator für die Relevanz eines Merkmals vorgeschlagen [Bre+84].
Die Gini Wichtigkeit ist dabei sozusagen ein Nebenprodukt des Trainingsprozesses von Entscheidungsbäumen. An jedem inneren Knoten τ eines binären Entscheidungsbaumes T wird mittels der Gini Unreinheit $g(\tau)$ der optimale Split bestimmt. Es wird derjenige Split ausgewählt, der zur größten Reduzierung der Unreinheit der Daten in Knoten τ führt. In Anlehnung an Kapitel 2.2 lässt sich der Gini Wert im Falle eines binären Klassifikationsproblems durch

$$g(\tau) = 1 - \hat{p}_0^2 - \hat{p}_1^2$$

bestimmen, wobei $\hat{p}_i$ die (geschätzte) Wahrscheinlichkeit der i-ten Klasse bezeichne. Werden ein Merkmal θ und eine Entscheidungsgrenze t_θ als Split ausgewählt, werden die Beobachtungen des Knoten τ auf zwei neue Knoten τ_l und τ_r aufgeteilt. Die Abnahme der Unreinheit durch das Durchführen dieses Splits ergibt sich dann als

$$\Delta_\theta g(\tau) = g(\tau) - p_l \cdot g(\tau_l) - p_r \cdot g(\tau_r).$$

p_l bezeichne dabei den Anteil aller Beobachtungen des Datensatzes, der im Knoten τ_l landet bzw. die geschätzte Wahrscheinlichkeit, diesen Knoten zu erreichen.
Ziel ist, aus allen am Knoten verfügbaren Merkmalen Θ (Random Forests schränken diese Menge an jedem Knoten zufällig ein (vgl. Kapitel 2.3.1)) jenes Tupel von Variable $\theta \in \Theta$ und geeignetem Schwellwert t_θ zu finden, das zum maximalen $\Delta_\theta g(\tau)$ führt. Der Rückgang der Gini Unreinheit, der sich aus dieser optimalen Aufteilung $\Delta_\theta g(\tau)(\tau, T)$ ergibt, wird für alle Knoten τ berechnet und anschließend für alle Variablen θ akkumuliert:

$$\sum_\tau \Delta_\theta g(\tau)(\tau, T)$$

Kann die Gini Wichtigkeit für einen einzelnen Entscheidungsbaum berechnet werden, lässt sich das Vorgehen auch leicht auf ein ganzes Ensemble von Modellen erweitern. Dazu muss lediglich der Wert aller Bäume T des Waldes aggregiert werden. Das führt dann zu

$$I_G(\theta) = \sum_T \sum_\tau \Delta_\theta g(\tau)(\tau, T)$$

Anschaulich ergibt sich mit der Gini Wichtigkeit I_G ein Maß für die Häufigkeit, mit der ein Merkmal θ als Split herangezogen wurde und wie groß dabei seine diskriminierende Wirkung war.
Das Vorgehen für Regressionsprobleme ist ähnlich: Anstatt der Gini Wich-

tigkeit wird in diesem Falle die Varianz der Daten innerhalb eines Knoten als Maß für die Unreinheit verwendet.

Zusammenfassung Gini Wichtigkeit

Die Gini Wichtigkeit I_G beschreibt, wie häufig ein Merkmal als Split herangezogen wurde und wie groß dabei seine diskriminierende Wirkung war. Dies ergibt sich als über alle Bäume gemitteltes Produkt aus der Unreinheitsreduzierung eines Merkmalssplits und der Wahrscheinlichkeit, diesen Split zu erreichen.

Vorteile:

- Ein großer Vorteil dieses Feature Importance Maßes ist seine einfache Berechnung. Die Wichtigkeit eines Merkmals lässt sich bereits beim Training berechnen.
- Die Wichtigkeit der Feature bietet einen hochkomprimierten, globalen Einblick in das Verhalten des Modells.
- Interaktionen zwischen Merkmalen werden durch dieses Vorgehen automatisch berücksichtigt, weil sie beim Training der Entscheidungsbäume Eingang finden.

Nachteile:

- Das Verfahren tendiert dazu, stetige Merkmale bzw. kategorielle Merkmale mit vielen verschiedenen Ausprägungen zu bevorzugen. Das liegt aber viel mehr in der Natur von Entscheidungsbäumen bzw. Random Forests und deren Trainingsprozess [HTF09].

4.3.2 *Permutationsbasierte Merkmalsrelevanz*

Ein anderer, modell-agnostischer Ansatz nähert sich dem Problem eher heuristisch: Die Wichtigkeit eines Merkmals kann mittels der Reduktion der Performance (Genauigkeit, F1-Score, R^2, etc.) analysiert werden, wenn ein Merkmal nicht mehr verfügbar ist. Die Relevanz eines Merkmals ergibt sich durch Entfernen des Merkmals aus dem Datensatz, Retrainieren des Schätzers und Überprüfen der Performanceveränderung. Das Retrainieren des Modells für jedes Merkmal wäre allerdings sehr rechenintensiv - insbesondere bei Problemen mit vielen Merkmalen. Darüber hinaus entspricht dies eigentlich der Wichtigkeit des Merkmals innerhalb des Datensatzes und nicht der Wichtigkeit des Merkmals für das trainierte Modell.

Um ein erneutes Training des Schätzers zu vermeiden, kann das Feature nur aus dem Testdatensatz entfernt werden. Anschließend erfolgt eine Reevaluation des Modells, ohne dabei dieses Feature zu verwenden. Allerdings kommen die meisten Modelle nicht mit fehlenden Inputmerkmalen zurecht. Anstatt ein Feature zu entfernen, ersetzen viele Ansätze es durch zufälliges Rauschen. Das Merkmal ist dann zwar noch vorhanden, enthält aber keine nützlichen Informationen mehr. Dazu muss das Rauschen lediglich

der gleichen Verteilung wie die ursprünglichen Merkmalswerte gehorchen. Der einfachste Weg, ein solches Rauschen zu erhalten, besteht im zufälligen Austausch der Werte eines Merkmals d.h. darin die Merkmalswerte anderer Beispiele zu verwenden. Alternativ kann die Verteilung anhand des Validierungsdatensatzes geschätzt und daraus zufällige Werte erzeugt werden. Im Falle von Random Forests bieten sich auch die Out-of-Bag-Daten an.
Führt dieses Vorgehen zu einem starken Ansteigen des Vorhersagefehlers, ist es naheligend, daraus eine hohe Relevanz des Merkmals schlusszufolgern. Das zu erklärende Modell macht seine Prognose stark von diesem Merkmal abhängig. Umgekehrt ist ein Merkmal *unwichtig*, wenn das Manipulieren seiner Ausprägung den Modellfehler nicht signifikant verändert. Die Messung der Wichtigkeit von Permutationsmerkmalen wurde von Breiman [Bre01] für Random Forests vorgestellt, lässt sich aber leicht auf andere Klassifikations- bzw. Regressionsmodelle übertragen. Strobl et al. konnten zeigen, dass dieses Vorgehen tendenziell miteinander korrelierende Merkmale in ihrer Bedeutung für das Modell überschätzt [Str+08].

Zusammenfassung Permutationsbasierte Merkmalsrelevanz

Die Bedeutung eines Merkmals wird durch die Zunahme des Vorhersagefehlers des Modells nach zufälligem Permutieren der Ausprägung des Merkmals gemessen. Dadurch wird die Beziehung zwischen dem Merkmal und dem tatsächlichen Ergebnis zerstört. Ein Merkmal ist unwichtig, wenn das Manipulieren seiner Ausprägung den Modellfehler nicht signifikant verändert. Anderenfalls handelt es sich um ein relevantes Feature.

Vorteile:

- Die Merkmalsrelevanz liefert einen hochkomprimierten, globalen Einblick in das Verhalten des Modells.
- Durch den modell-agnostischen Ansatz ist die Messungen der Merkmalswichtigkeit über verschiedene Probleme hinweg vergleichbar.
- Die permutationsbasierte Feature Importance berücksichtigt automatisch alle Interaktionen mit anderen Merkmalen. Durch die Veränderung des Features wird auch der Interaktionseffekt mit anderen Merkmalen zerstört. Daraus ergibt sich die Berücksichtigung von Haupt- und Interaktionseffekt auf die Modellperformance (siehe auch Nachteile).
- Es ist kein Retraining des Modells nötig, was unter Umständen sehr aufwendig sein könnte.

Nachteile:

- Es ist unklar, ob Trainings- oder Testdaten verwenden werden sollten, um die Wichtigkeit des Merkmals zu berechnen [Molsu].
- Die permutationsbasierte Feature Importance berücksichtigt automatisch alle Interaktionen mit anderen Merkmalen. Dies ist Vor- und Nachteil zugleich, da die Bedeutung der Interaktion zwischen zwei Merkmalen in die Wichtigkeitsmessungen beider Merkmale einbezogen wird. Das bedeutet, dass sich die Feature Importance nicht zum Gesamtperformanceverlust addiert, sondern größer ist. Nur wenn es keine Interaktion zwischen den Merkmalen gibt - wie zum Beispiel in einem linearen Modell - summieren sich die Wichtigkeiten entsprechend.
- Durch das zufällige Permutieren können - wie bei Partial Dependence Plots - unrealistische Beobachtungen entstehen, wenn es Korrelationen zwischen Merkmalen gibt.
- Die Ergebnisse sind stark von der zufälligen Permutation des Merkmals abhängig, was dem Ergebnis eine gewisse Zufälligkeit verleiht.

4.4 LIME

LIME ist ein Framework, das 2016 von Ribeiro et al. vorgestellt wurde und sich bei vielen Praktikern großer Popularität erfreut [RSG16a]. LIME steht dabei für Local Interpretable Model-agnostic Explanation und das Akronym beinhaltet alle wesentlichen Konzepte des Verfahrens.

Lokal: Eine zentrale Intuition hinter LIME ist die Erklärung einer einzelnen lokalen Beobachtung. Black-Box-Modelle sind von ihrer Natur aus sehr komplex in der Abbildung von Input- auf Outputwerte. Meist handelt es sich dabei um hochgradig nichtlineare, mehrdimensionale Funktionen, so dass es schwer ist, deren globales Verhalten zu erfassen. Eine einzelne Beobachtung (ein konkretes Mapping zwischen Input- und Outputbereich) enthält nur einen kleinen Teil dieser Komplexität. Daher ist es viel einfacher, ein Black-Box-Modell durch ein einfaches Modell lokal (in der Nähe der interessierenden Beobachtung) zu approximieren. Formal handelt es sich bei LIME um ein sogenanntes (lokales) Surrogate Modell oder auch Ersatzmodell. Dies sind Modelle, die die Vorhersagen, nicht aber unbedingt die Logik des zugrunde liegenden Black-Box-Modells approximieren.

Das zweite Konzept von LIME bezieht sich auf genau dieses Surrogat. Anstatt ein beliebiges Ersatzmodell zu trainieren, beschränkt sich die Methode dabei auf interpretierbare Modelle, wie z.B. Entscheidungsbäume oder eine lineare Regression. Die Erklärung zur Vorhersage des Black-Box-Modells entsteht durch Interpretation des lokalen, interpretierbaren Surrogates. Beispielsweise lassen sich im Falle einer linearen Regression die Produkte aus Koeffizienten und Merkmalen als Einfluss eines jeden Merkmals bezeichnen.

Zusätzlich soll die Erklärung modell-agnostisch sein. Daher muss sich die Methodik erneut auf das Manipulieren von Eingabemerkmalen und Beobachten der korrespondierenden Vorhersage reduzieren. Dies erweist sich allerdings sogar als Vorteil in Bezug auf die Interpretierbarkeit. Manipulationen der Inputvariablen beziehen sich immer auf für Menschen verständliche Komponenten (z.B. Wörter oder Bildteile), selbst wenn das Black-Box-Modell wesentlich komplexere Komponenten als Merkmale verwendet (z.B. Word-Embeddings) [RSG16a].

Im Folgenden soll das heuristische, algorithmische Vorgehen erläutert werden, anhand dessen LIME seine Erklärungen erzeugt. Zunächst eine intuitive Erklärung des Ansatzes:
Als Erstes muss eine zu erklärende Beobachtung x ausgewählt werden. Anschließend erzeugt LIME anhand der Verteilung eines jeden Merkmales im Datensatz neue, synthetische Beobachtungen (ohne dabei Interaktionen zu berücksichtigen). Dazu muss die Verteilung aller Merkmale zunächst geschätzt werden.

So entsteht ein neuer Datensatz, der aus permutierten Samples und den entsprechenden Vorhersagen des Black-Box-Modells f besteht. Jede Beobachtung wird anschließend anhand ihrer Distanz zur interessierenden Beobachtung x gewichtet. Auf diesem gewichteten Datensatz trainiert LIME ein erklärbares Modell - z.B. eine lineare Regression - mit den Vorhersagen des Black-Box-Modells als Zielgröße. Dieses gelernte Modell sollte eine gute lokale Näherung an die Vorhersagen des maschinellen Lernverfahrens sein.

Meist wird es aber keine gute globale Approximation darstellen. Das LIME-Surrogat hat demnach eine hohe *local fidelity*, aber meist keine gute *general fidelity* (vgl. hierzu 3.2).

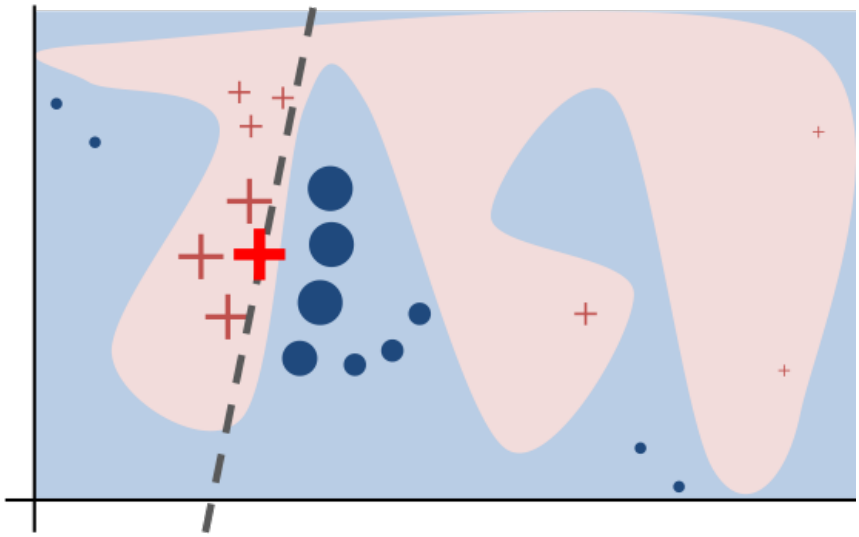


Abbildung 4.4: Intuition hinter LIME (Quelle: [RSG16a])

In Abbildung 4.4 ist das Verfahren graphisch illustriert. Die komplexe Entscheidungsfunktion f des Black-Box-Modells wird durch den blau/rosa-gefärbten Hintergrund dargestellt und kann durch ein lineares Modell nicht gut approximiert werden. Das dicke rote Kreuz ist die zu erklärende Instanz. LIME erzeugt Instanzen, prädiziert diese mittels f und wägt sie durch die Nähe zu der zu erklärenden Instanz ab (hier dargestellt durch Größe). Die gestrichelte Linie ist die gelernte Erklärung bzw. das Surrogat, das lokal (aber nicht global) treu ist.

Mathematisch sind lokale Surrogate mit den gerade vorgestellten Randbedingungen wie folgt darstellbar:

$$\eta(x) = \arg\min_{g \in G} L(f, g, \pi_x) + \Omega(g) \tag{4.2}$$

Ziel ist das Finden einer Erklärung η der Vorhersage des Black-Box-Modells f für die Beobachtung x. Dazu wird jede Erklärung g (z.B. eine lineare Regression) aus der Familie der erklärbaren Funktionen G (z.B. alle theoretisch denkbaren linearen Regressionsfunktionen) in Betracht gezogen, um diejenige zu extrahieren, die die Summe $L(f, g, \pi_x) + \Omega(g)$ minimiert.
L bezeichne dabei eine Loss-Funktion (z.B. die mittleren Fehlerquadrate), die die Differenz zwischen Black-Box-Modell f und dem lokalen Surrogat g misst. Zusätzlich ist die Verlustfunktion noch von einem lokalen Kernel π_x abhängig, der spezifiziert, wie die Distanz zwischen den generierten, synthetischen Dateninstanzen und x zu messen ist. $\Omega(g)$ ist ein Regularisierungsterm und beschreibt die Komplexität des erklärenden Surrogates g

[RSG16a].
In der Praxis müssen vom Anwender sowohl die Familie an interpretierbaren Funktionen G, die Verlustfunktion L, der lokale Kern π_g und der Regularisierungsterm $\Omega(g)$ festgelegt werden. Dies geschieht meist heuristisch, was das Verfahren in der Praxis sehr fehleranfällig und anspruchsvoll macht. Darüber hinaus wird in der Praxis nur der Loss minimiert, da der Anwender die Komplexität, d.h. die Anzahl an Merkmalen des Surrogates g vorgibt. Die Verwendung einer Backward- bzw. Forward-Selection oder das Anpassen des entsprechenden Regularisierungsparamters bei einer Lasso bzw- Ridge-Regression, stellen diese Anzahl an Merkmalen sicher.

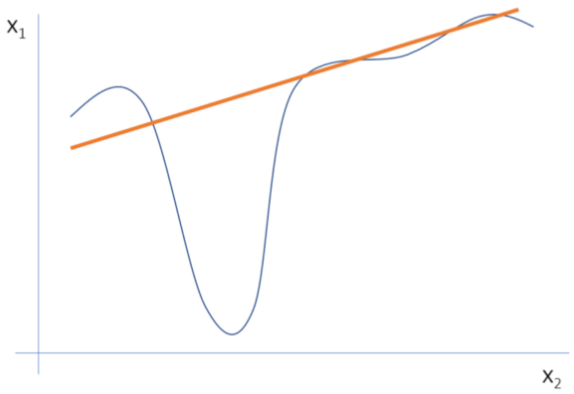


Abbildung 4.5: Problem bei der Approximation von lokalen Nichtlinearitäten

Zusammenfassung LIME

LIME liefert lokale Erklärungen und ermöglicht es, den Einfluss von Feature-Änderungen auf die Vorhersage zu untersuchen. Dazu wird das zugrunde liegende Black-Box-Modell lokal durch ein interpretierbares Modell approximiert. Die interpretierbaren Modelle sind auf synthetischen Beobachtungen trainiert und sollen nur eine gute lokale Annäherung liefern. Dadurch wird die Berechnung des Surrogates deutlich vereinfacht.

Vorteile:

- LIME ist modell-agnostisch. Das ermöglicht den einfachen Austausch des zugrunde liegenden Black-Box-Modells.
- Das Training und die Interpretation der als lokales Surrogat in Frage kommenden Modelle ist gut erforscht.
- Bei der Verwendung von Lasso oder kurzen Bäumen sind die daraus resultierenden Erklärungen kurz (im Sinne von selektiv) und können kontrastiv gestaltet werden.
- LIME kann für tabellarische Daten, Bilder & Texte angewendet werden.
- Mit LIME erstellte Erklärungen können andere Merkmale als das Black-Box-Modell verwenden. Dies kann ein großer Vorteil gegenüber anderen Methoden sein, insbesondere wenn die ursprünglichen Merkmale nicht interpretiert werden können.

Nachteile:

- Die Wahl des richtigen lokalen Kerns ist ein großes Problem bei LIME, hängen die Erklärungen doch stark von dessen Parametrisierung ab.
- Bislang erfolgt das Sampling der neuen Beobachtungen allein anhand der durch Normalverteilungen geschätzten Merkmalsverteilungen. Korrelationen zwischen Merkmalen werden dabei genauso wenig berücksichtigt wie die Tatsache, dass dadurch auch unrealistische Datenpunkte erzeugt werden können.
- Ein weiteres großes Problem ist die Instabilität des Verfahrens. Die Erklärungen von zwei sehr ähnlichen Beobachtungen können stark voneinander abweichen [AJ18].
- Die Komplexität der Erklärung muss im Voraus definiert werden. Dies ist allerdings nur ein kleines Problem, denn letztendlich ist es immer dem Anwender überlassen, den optimalen Trade-Off zwischen *fidelity* und *sparsity* zu finden (vgl. hierzu auch Abschnitt 3.2).
- Wird eine lineare Regression als Surrogat verwendet, ist die Interpretation stark vom Intercept der Regression abhängig. Es ist aber nach wie vor unklar, wie dieser in einer lokalen Region zu interpretieren ist.
- Es werden bislang nur lineare Modelle zur Approximation verwendet. Dies liegt in der heuristischen Annahme begründet, dass, wenn ein sehr kleiner Bereich um die zu erklärende Instanz herum betrachtet wird, das Black-Box-Modell annähernd linear ist. Durch die minimale Erweiterung dieses Bereichs ist es jedoch möglich, dass ein lineares Modell nicht leistungsfähig genug ist, um das Verhalten des ursprünglichen Modells zu erklären (vgl. hierzu Abbildung 4.5).

4.5 SHAPLEY VALUES

Wie die anderen bereits vorgestellten Verfahren, mittels derer ein Modell post-hoc erklärt werden kann, sind auch Shapley Values modell-agnostisch, d.h. auf die Manipulation des Modellinputs und Beobachten der Veränderung des Outputs beschränkt. Es wurde bereits mehrfach darauf hingewiesen, dass diese Einschränkung durchaus auch Vorteile in Bezug auf Portabilität und den Vergleich von Modellen hat.
Shapley Values ermöglichen es die Vorhersage des Modells so aufzuschlüsseln, dass die Auswirkungen eines einzelnen Features auf die Modellprognose sichtbar werden. Dadurch entsteht zunächst eine lokale Erklärung für jede Beobachtung. Allerdings ist es möglich, die Beiträge eines jeden Features über mehrere Beobachtungen zu aggregieren und so den globalen mittleren Einfluss des Features auf die Vorhersage zu erhalten.

Zu Beginn ein einfaches anschauliches Beispiel - ein lineares Regressionsmodell:

$$f(x) = f(x_1, \dots x_n) = \beta_0 + \beta_1 x_1 + \cdots \beta_n x_n,$$

wobei $x \in X$ eine Beobachtung aus dem Datensatz X bezeichne und dieser n verschiedene Merkmale X_i habe. Das Ziel ist es die Vorhersage für die Beobachtung x lokale zu erklären. Das Produkt aus Koeffizient und Merkmalsausprägung $\beta_i x_i$ kann als die (globale) Wichtigkeit des i-ten Features interpretiert werden. In der Praxis interessiert jedoch mehr, wie ein bestimmtes Feature die Vorhersage beeinflusst,

$$\phi_i(x) = \beta_i x_i - \beta_i E[X_i] \tag{4.3}$$

Der Einfluss ergibt sich aus der Differenz zwischen dem Feature-Effekt und dem Durchschnittseffekt. Etwas allgemeiner formuliert kann (4.3) auch wie folgt umgeschrieben werden:

$$\phi_i(x) = f(x_1, \dots x_n) - E[f(x_1, \dots, X_i, \dots, x_n)] \tag{4.4}$$

Gleichung (4.4) ist die Differenz zwischen der Vorhersage für eine Beobachtung und der erwarteten Vorhersage für diese Beobachtung, wenn das i-te Merkmal unbekannt ist. Im Falle eines linearen additiven Modells (ein Modell ohne Feature-Interaktion) entspricht dies gerade Gleichung (4.3). In der Praxis interagieren Feature allerdings oft. Um dieses Problem zu umgehen, muss jede Teilmenge an Featuren Berücksichtigung finden. Diese Verallgemeinerung ermöglicht das Betrachten von Interaktionen und führt auf die sogenannten Shapley Values. Eine genaue mathematische Herleitung dazu findet sich bei [ŠK14], der Beweis in [ŠK13].

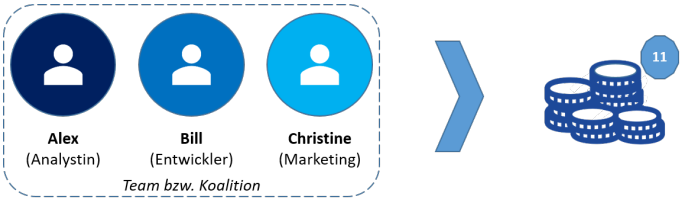
Der Shapley Wert entstammt ursprünglich der kooperativen Spieltheorie und befasst sich mit der fairen Allokation von Gewinnen in kooperativen Spielen. Genauer geht es um die folgende Fragestellung:

> Gegeben eine Koalition, die aus mehreren Spielern mit unterschiedlichen Fähigkeiten besteht, die einen kollektiven Gewinn erwirtschaften, wie sieht die fairste Verteilung des Gewinns auf die Spieler aus?

Bevor die Berechnungsformel für Shapley Values, ihre mathematische Herleitung und deren Bedeutung für die Interpretation von maschinellen Lernverfahren betrachtet werden, zunächst ein Beispiel zur Illustration der obigen Fragestellung und der Gestalt möglicher Lösungsansätze. Es wird sich später sogar zeigen, dass unter (wünschenswerten) Annahmen dies die eindeutige Problemlösung ist.

4.5.1 *Der Shapley Value - spieltheoretische Grundlagen*

Ein einführendes Beispiel



Angenommen, die Koalition besteht aus drei Personen (Alex, Bill und Christine) mit unterschiedlichen Fähigkeiten, die gemeinsam einen Gewinn von 11 erwirtschaftet haben. Zur Vereinfachung sei angenommen, dass die drei nacheinander zur Gruppe hinzustoßen. Jeder Spieler soll den Reward erhalten, den er zum Gesamtgewinn beigetragen hat (marginale Beiträge). Wenn Alex das erste Mitglied der Gruppe war, der Gruppengewinn 5 betrug und sich durch die folgenden Eintritte von Bill bzw. Christine zunächst auf 9 und schließlich auf 11 erhöhte, führt dies im Modell zu folgenden Auszahlungen: $(A, B, C) \longrightarrow (5, 4, 2)$.
Aber was passiert, wenn Bill und Christine ähnliche Fähigkeiten haben. Dann könnte es passieren, dass Bill einen höheren marginalen Betrag als Christine erhält, wenn er zuerst zur Gruppe stößt und damit auch zuerst die überlappenden Fähigkeiten zur Verfügung stellt. Die Reihenfolge der Eintritte in die Gruppe spielt folglich eine Rolle bei der Verteilung des Outputs. Wie lässt sich dann die fairste Auszahlungsverteilung bestimmen?
Ein einfacher Lösungansatz wäre es, für jede mögliche Reihenfolge der drei Spieler (ABC, ACB, CBA, CAB, BAC und BCA), die marginalen Auszahlungen eines jeden Spielers zu bestimmen, aufzusummieren und durch die Anzahl der möglichen Reihenfolgen zu teilen, um so den gemittelten marginalen Beitrag eines jeden Spielers zu erhalten. Vereinfacht gesagt, wird dies mittels Shapley Werten bestimmt. Allerdings ist dieses Vorgehen nicht geeignet, da ansonsten bei m Merkmalen $m!$ Reihenfolgen betrachtet werden müssten.

Es bedarf erst einiger Definitionen, bevor eine detaillierte Analyse der sogenannten Shapley Values möglich ist.

Definition 1 (Kooperatives Spiel) *Gegeben sei eine Menge an Spielern $N = \{1, ..., n\}$ mit $n \in \mathbb{N}$ und eine Funktion v, die sogenannte charakteristische Funktion, die jeder Teilmenge an Spielern eine reelle Zahl zuordnet:*

$$v : \mathcal{P}(N) \longrightarrow \mathbb{R}$$

und dabei die folgenden beiden Eigenschaften erfüllt:

1. $v(\emptyset) \longmapsto 0$
2. *(Super-Additivität) Für zwei disjunkte Koalitionen S und T ($S \cap T = \emptyset$) gilt immer:*

$$v(S) + v(T) \leq v(S \cup T)$$

Jede nichtleere Teilmenge $S \subseteq N$ von Spielern heißt Koalition und $v(S)$ bezeichne den Wert der Koalition - also den zu erwarteten Gesamtgewinn der Spieler durch das Koalieren. Das Tupel (v, N) bildet ein kooperatives Spiel.
Zusätzlich sei die Auszahlung eines einzelnen Spielers $i \in S$ als $\phi_i \leq v(S)$.

Nach der Definition des Spiels nun zur Frage nach der Verteilung des Gesamtgewinns der Koalition auf die verschiedenen Spieler. Dies kann auf verschiedene Art und Weise gelöst werden. Shapley Werte sind die fairste Verteilung des Gewinns im Sinne der später getroffenen, wünschenswerten Annahmen. Die Shapley Werte definieren sich wie folgt:

Definition 2 (Shapley Value) *Gegeben sei ein kooperatives Spiel (v,N), dann berechnet sich der Shapley Value für Spieler $i \in S \subseteq N$ mittels:*

$$\phi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!\,(|N| - |S| - 1)!}{|N|!} (v(S \cup \{i\}) - v(S)) \tag{4.5}$$

wobei über alle Teilmengen summiert wird, die den Spieler i nicht enthalten.

Zunächst soll der Formel etwas Intuition verliehen werden: Angenommen, die Koalition wird sequentiell gebildet, wobei jeder Akteur i seinen Beitrag zum Gesamtgewinn $v(S \cap i) - v(S)$ als Auszahlung erhält. Da bereits im einführenden Beispiel klar wurde, dass die Reihenfolge, in der die Spieler beitreten, eine Rolle spielt, erfolgt die Aggregation über alle möglichen Reihenfolgen der Spieler. Dabei lassen sich allerdings mehrere Spielerreihenfolgen zusammenfassen (beispielsweise ist es aus Sicht jedes Spielers irrelevant, welche Spieler nach ihm Teil der Koalition werden), so dass die Auszahlung an den Spieler i ($v(S \cap i) - v(S)$) noch entsprechend gewichtet werden muss. Zusammengefasst beschreibt der Shapley Wert eines Spielers die durchschnittliche Änderung des Gesamtgewinns der Spieler, die bereits Teil der Koalition sind, wenn der neue Spieler sich ihnen anschließt.

Bevor die interessante Eigenschaft der Shapley Werte ins Zentrum rückt, ist eine letzte Definition nötig; die der Wertfunktion:

Definition 3 (Wertfunktion) *Sei $\mathcal{G}_N$ die Menge aller kooperativen Spiele (v, N). Eine Wertfunktion Φ ordnet jedem kooperativen Spiel (v, N) eine optimale Allokation zu. Beispielsweise $\Phi : \mathcal{G}_N \mapsto \mathbb{R}^{|\mathbb{N}|}$ so dass $\Phi(v, N)(N) = v(N)$. Shapley Values sind eine Wertfunktion.*

Shapley Values haben eine anschauliche Interpretation, die sich mit dem Gerechtigkeitsempfinden vieler Menschen deckt; jeder Spieler erhält seinen Beitrag zum Gesamtgewinn der Gruppe als Auszahlung. Es lässt sich zeigen, dass diese Art der Güterverteilung die einzige ist, die die folgenden vier Axiome erfüllt:

1. **Optimalität:** Die Summe der Shapley Werte aller Spieler entspricht dem Wert der gesamten Koalition, so dass der Gewinn vollständig unter den Agenten verteilt wird:

$$\sum_{i \in N} \phi_i(v) = v(N)$$

2. **Symmetrie:** Spieler mit den gleichen marginalen Beiträgen zum Gesamtgewinn erhalten die gleiche Auszahlung:

$$v(S \cup \{i\}) = v(S \cup \{j\}) \implies \phi_i(v) = \phi_j(v) \qquad \forall S \setminus \{i, j\} \in N$$

3. **Linearität:** Wenn das Spiel in zwei unabhängige Spiele zerlegt werden kann, dann ist die Auszahlung jedes Spielers im zusammengesetzten Spiel die Summe der Auszahlungen in den aufgeteilten Spielen:

$$\phi_i(\alpha \cdot v + \beta \cdot w) = \alpha \cdot \phi_i(v) + \beta \cdot \phi_i(w)$$

4. **Dummy-Spieler:** Ein Spieler mit einem marginalem Beitrag von Null zum Gesamtgewinn der Koalition erhält eine Auszahlung von Null: Der Shapley Value $\phi_i(v)$ für einen Nullspieler i in einem Spiel v ist Null, wenn $v(S \cup i) = v(S)$ für alle Koalitionen S, die nicht i enthalten.

Satz 1 *Shapley Werte sind die eindeutige Wertfunktion, die die Eigenschaften der Effizienz, Symmetrie, Linearität und des Dummy-Spielers erfüllen.*

Ein Beweis des Satzes findet sich bei [Sha53].

4.5.2 *Der Shapley Value als Werkzeug zur Modellerklärung*

Um die Verknüpfung des spieltheoretischen Konzepts mit der Erklärbarkeit maschineller Lernverfahren zu erläutern, sei nochmals an die ursprüngliche Idee zu Beginn das Kapitels erinnert: Eine Vorhersage des Modells so aufzuschlüsseln, dass die Auswirkungen eines einzelnen Features auf die Modellprognose sichtbar wird. Shapley Values stellen eine faire Verteilung des Gesamtgewinns auf die Teilnehmer einer Koalition dar, abhängig von den Fähigkeiten der Teilnehmer. Wenn jede Featureausprägung als Teilnehmer eines Spiels und die Differenz zwischen der durchschnittlichen Vorhersage

des Modells und der Prognose der zu erklärenden Beobachtung als Gesamtoutput der Koalition betrachtet wird, lassen sich Shapley Values auf das Ausgangsproblem übertragen.

Das heißt, mittels Shapley Values kann jedem Feature ein Einfluss auf die Vorhersage des Modells zugeordnet werden. Aber nicht nur das: Es entsteht sogar zusätzlich die Gewissheit, dass diese Zuordnung die einzige Wertfunktion ist, die die vier Shapley Axiome erfüllt. Lundberg et al. verallgemeinern diese Eigenschaften für eine ganze Klasse an Erklärbarkeitsansätzen (sog. Additive Feature Attribution Methods) und zeigen damit, dass alle Erklärbarkeitsansätze dieser Klasse, die nicht auf Shapley Values beruhen, eines der vier Axiome bzw. daraus abgeleitete Eigenschaften verletzen [LL17]. Das daraus entwickelte, auf Shapley Values beruhende Verfahren SHAP wird in Kapitel 4.6 erläutert. Zuvor werden die Formel zur Berechnung der Shapley Werte und die vier Axiome noch einmal in Bezug zur Erklärbarkeit von Modellen gesetzt:

Die Übertragung der vier Axiome auf die Domäne maschineller Lernverfahren ist unmittelbar. Sie haben teilweise sogar eine sehr anschauliche und sinnvolle Interpretation:

- **Optimalität:** Die Feature-Beiträge sollen sich gerade zur Differenz der Vorhersage für x und der durchschnittlichen Vorhersage aufsummieren.

$$\sum_{j=1}^{p} \phi_j = \hat{f}(x) - E_X(\hat{f}(X))$$

- **Symmetrie** und **Dummy-Spieler** übertragen sich eins zu eins von Spielern auf Feature.

- **Linearität:** Dies wird anhand eines Beispiels klar: Gegeben sei ein Random Forest. Die Vorhersage setzt sich aus den Vorhersagen vieler Entscheidungsbäume zusammen. Die Eigenschaft der Linearität garantiert, dass für ein Feature der Shapley Wert für jeden Baum einzeln berechenbar ist und sich durch Mitteln der Shapley Werte für den Random Forest ergibt.

Nun zur Berechnungsformel. Dazu nochmals Gleichung (4.5):

$$\phi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!\,(|N| - |S| - 1)!}{|N|!} (v_x(S \cup \{i\}) - v_x(S))$$

wobei S die Teilmenge der im Modell verwendeten Merkmale, x der Vektor der Feature-Ausprägungen der zu erklärenden Instanz und N die Menge der Feature sind. $v(S)$ ist die Vorhersage des Modells für die Merkmalsausprägungen in der Menge S, d.h. so als wären die anderen Feature ($\bar{S}$) unbekannt.

Diese Vorhersage ergibt sich durch Marginalisieren der Feature, die nicht in S enthalten sind:

$$v_x(S) = \int f(x_1, \ldots, x_N) d\mathbb{P}_{x \notin S} - E_X(f(X)). \tag{4.6}$$

Diese Berechnung kann sehr schnell aufwendig werden, da bereits in einem einfachen Setting mit zum Beispiel fünf Variablen X_1, X_2, X_3, X_4, X_5 mehrfache Integration durchzuführen ist. Besteht die Menge S bspw. aus den Variablen X_1, X_2, so muss das unten stehende Integral gelösten werden, um $v(S)$ zu berechnen:

$$v_x(S) = v_x(\{x_1, x_2\}) = \int_{\mathbb{R}} \int_{\mathbb{R}} f(x_1, x_2, X_3, X_4, X_5) d\mathbb{P}_{X_3 X_4 X_5} - E_X(f(X)).$$

4.5.3 *Approximation von Shapley Values*

Die Berechnung der Funktion $v(S)$ wird durch eine weitere Tatsache zusätzlich erschwert: In der Regel ist die funktionale Form des Machine Learning Modells nicht bekannt. Selbst wenn dies der Fall wäre, handelt es sich dabei oft um mehrdimensionale, hochgradig nichtlineare und komplexe Funktionen, so dass die exakte Berechnung der Integrale meist nicht möglich ist.

Hinzu kommt, dass die exakte Berechnung der Lösung bereits für einige wenige Merkmale schwierig ist, da die Anzahl der möglichen Koalitionen exponentiell zunimmt, je mehr Feature hinzugefügt werden. Zusätzlich muss das *Fehlen* eines Merkmals durch Ziehen von Zufallsinstanzen simuliert werden, was die Varianz für die Schätzung der Shapley Werte erhöht.

Zusammengefasst existiert keine exakte Berechnungsvorschrift für Shapley Values (4.5), die nicht mindestens mit einer exponentiellen Komplexität - in Abhängigkeit der Merkmalsanzahl - verbunden wäre [FK92]. Daher schlagen Štrumbelj und Kononenko eine approximative Berechnung der Integrale bzw. des Shapley Values über Monte Carlo Simulationen vor [ŠK14]:

Dazu erfolgt zunächst eine Überführung von Gleichung (4.5) in die folgende äquivalente Darstellung:

$$\phi_i(v) = \frac{1}{n!} \sum_{\mathcal{O} \in \pi(N)} v_x(\text{Pre}^i(\mathcal{O}) \cup \{i\}) - v_x(\text{Pre}^i(\mathcal{O})) \tag{4.7}$$

wobei $\pi(N)$ die Menge aller geordneten Permutationen der Merkmalsindizes $\{1, 2, \ldots, n\}$ beschreibt. $\text{Pre}^i(\mathcal{O})$ ist die Menge aller Indizes, die dem i-ten Merkmal in der Permutation $\mathcal{O} \in \pi(N)$ vorausgehen.
Wenn die Berechnung der v-Terme mit Kosten von Null verbunden wäre, würde dies (4.7) eine Schätzung über einen simplen Monte Carlo Sampling Algorithmus ermöglichen, wobei $v_x(\text{Pre}^i(\mathcal{O}) \cup \{i\}) - v_x(\text{Pre}^i(\mathcal{O})$ eine Stichprobe repräsentiert. Ein solcher Algorithmus wurde beispielsweise von Ca-

stro et al. vorgeschlagen [CGT09].

Allerdings benötigt die Berechnung der v-Terme eine exponentielle Laufzeit. Daher erfolgt eine Einschränkung der Merkmalsverteilungen auf Variablen, die unabhängig voneinander verteilt sind [ŠK11]. Dadurch vereinfacht sich die Berechnung, was zu folgendem Monte Carlo Ansatz zur approximativen Berechnung führt:

$$\hat{\phi}_j = \frac{1}{M} \sum_{m=1}^{M} \left(\hat{f}(x_{+j}^m) - \hat{f}(x_{-j}^m) \right) \tag{4.8}$$

Dabei bezeichne $\hat{f}(x_{+j}^m)$ die Vorhersage für x. Allerdings wird eine zufällige Auswahl an Merkmalsausprägungen durch Merkmalswerte aus einem anderen zufälligen Datenpunkt z ersetzt. Ausgenommen ist dabei lediglich die Ausprägung des Merkmales j. Für x_{-j}^m erfolgt ein ähnliches Vorgehen wie bei x_{+j}^m, allerdings wird der Wert x_j^m aus dem gesampelten z übernommen. Dieses Vorgehen führt zu folgendem Algorithmus [ŠK14]:

Output : Shapley Wert für die Ausprägung des j-ten Features
Input : Anzahl der Iterationen M, Beobachtung von Interesse x, Feature-Index j, Datenmatrix X und Machine Learning Modell f

1 **for** $m = 1, \ldots, M$ **do**
2 Ziehe eine zufällige Instanz z aus der Datenmatrix X
3 Wähle eine zufällige Permutation $\mathcal{O}$ der Merkmalswerte
4 Ordne Beobachtung x: $x_{\mathcal{O}} = (x_{(1)}, \ldots, x_{(j)}, \ldots, x_{(p)})$
5 Ordne Beobachtung z: $z_{\mathcal{O}} = (z_{(1)}, \ldots, z_{(j)}, \ldots, z_{(p)})$
6 Erzeuge zwei neue Beobachtungen:
- mit j : $x_{+j} = (x_{(1)}, \ldots, x_{(j-1)}, x_{(j)}, z_{(j+1)}, \ldots, z_{(p)})$
- ohne j: $x_{-j} = (x_{(1)}, \ldots, x_{(j-1)}, z_{(j)}, z_{(j+1)}, \ldots, z_{(p)})$

Berechne den marginalen Beitrag des Features:
$\phi_j^m = \hat{f}(x_{+j}) - \hat{f}(x_{-j})$
7 **end**
8 Berechne den Shapley Value als Mittelwert der Schätzungen:
$\phi_j(x) = \frac{1}{M} \sum_{m=1}^{M} \phi_j^m$

Algorithmus 2 : Monte Carlo Simulation nach [ŠK14]

Štrumbelj et al. schlagen zwei Erweiterungen vor, die die Effizienz des Approximationsalgorithmus weiter erhöhen:
Zum einen stellt der Approximationsalgorithmus eine Form von Monte Carlo Integration dar. Daher wird durch den Einsatz von Quasi-Zufallsstichproben anstelle von Pseudozufallsstichproben eine schnellere Konvergenz des Verfahrens ermöglicht.

Außerdem muss, um die Erklärung für eine Instanz x zu berechnen, der Beitrag eines jeden der n Feature zur Vorhersage dieser Beobachtung berechnet werden. In der Praxis soll dies in möglichst kurzer kontrollierter Zeit geschehen und der allgemeine Näherungsfehler dennoch minimiert werden [ŠK14]. Der Näherungsfehler des Schätzers $\phi_i(x)$ hängt von der Varianz des Datensatzes ab, die nicht für alle Merkmale gleich ist. Da in der Praxis nur eine bestimmte Anzahl von Stichproben m zur Verfügung steht, erscheint es sinnvoll, die Anzahl m_i der für ein Merkmal i gezogenen Stichproben an die Varianz dieses Merkmals σ_i^2 anzupassen. Mehr zu den beiden Verbesserungen und dem sich daraus ergebenden Algorithmus findet sich in [ŠK14].

Bevor das auf Shapley Values beruhende Verfahren SHAP beleuchtet wird, noch eine kurze Zusammenfassung der Vor- und Nachteile von Shapley Values:

Zusammenfassung Shapley Values

Shapley Values berechnen die Merkmalsrelevanz für eine Beobachtung und liefern zunächst lokale Erklärungen. Dazu vergleichen sie vereinfacht gesagt für jedes Feature die Vorhersage des Modells mit und ohne der Kenntnis von dessen Ausprägung. Da die Reihenfolge der Merkmale die Vorhersage beeinflussen kann, betrachten sie alle möglichen Reihenfolgen und Teilmengen der anderen Variablen. Durch Anwenden der Berechnungsformel für jede Beobachtung können auch globale Eindrücke vom Modellverhalten gewonnen werden.

Vorteile:

- Shapley Values sind modell-agnostisch, d.h. für jedes ML-Modell anwendbar.
- Shapley Values ermöglichen es, sowohl lokale als auch globale Erklärungen des Modells zu generieren.
- Es ist möglich, kontrastive Erklärungen zu erzeugen. Anstatt die Vorhersage eines einzelnen Datenpunktes mit der durchschnittlichen Vorhersage des gesamten Datensatzes zu vergleichen, lässt sie sich auch zu einer Gruppe von Beobachtungen oder einer anderen Beobachtung ins Verhältnis setzen.
- Der Shapley Wert ist Stand heute die einzige Erklärungsmethode mit einer fundierten mathematischen Theorie. Die Axiome Effizienz, Symmetrie, Dummy-Spieler und Linearität stellen die Erklärung auf eine solide mathematische Grundlage. Andere Ansätze wie z.B. LIME gehen heuristisch von einem lokal linearen Verhalten des Modells aus, ohne dies entsprechend untermauern zu können.
- Die Differenz zwischen der Vorhersage für eine Beobachtung und der durchschnittlichen Vorhersage des Modells wird fair (im Sinne der Axiome) auf die Feature-Ausprägungen der Beobachtung verteilt. Das Axiom der Effizienz garantiert dies. Diese Eigenschaft unterscheidet den Shapley Wert von anderen Methoden wie z.B. LIME [Molsu].

Nachteile:

- Shapley Values erfordern oft sehr viel Rechenzeit, da Monte Carlo Simulationen zum Einsatz kommen. Für jede Ausprägung jedes Merkmals muss einmal Algorithmus 2 durchgeführt werden.
- Eine Erklärung, die mit Shapley Werten erstellt wurden, verwendet immer alle Feature. Der Mensch bevorzugt selektive Erklärungen, wie sie beispielsweise von LIME erzeugt werden. SHAP kann auch Erklärungen mit weniger Featuren liefern.
- Shapley Werte geben nur eine Zahl pro Feature zurück, aber kein Vorhersagemodell wie z.B. LIME. Dadurch sind auch keine Aussagen über Veränderungen der Prognose durch Anpassung von Merkmalen möglich. (z.B.: "Wenn ich 100 Euro mehr pro Jahr verdienen würde, würde sich meine Kreditwürdigkeit um 5 Punkte erhöhen.")
- Wie viele andere Erklärbarkeitsansätze, die mit Permutationen der Daten arbeiten, kann es auch bei Shapley Values zu unrealistischen Merkmalsausprägungen kommen, wenn Feature korreliert sind.

4.6 SHAP - **sh**APLEY **a**DDITIVE EX**p**LANATIONS

SHAP ist ein Verfahren zum Generieren von Erklärungen, das von Lundberg und Lee (2016) [LL17] vorgeschlagen wurde und auf Shapley Values beruht. Zunächst definieren sie eine Klasse von Erklärbarkeitsmodellen und zeigen dann, dass diese Klasse unter wünschenswerten Annahmen eine eindeutige Lösung hat. Anschließend definieren sie die SHAP Werte und leiten verschiedene Algorithmen zur Berechnung her.

4.6.1 *Additive Feature Attribution Methods*

Für ein einfaches Modell ist das Modell selbst immer die beste Erklärung. Bereits im Kapitel zu LIME (4.4) wurde auf die komplexen Abbildungen zwischen Input- und Outputwerten innerhalb von Black-Box-Modellen hingewiesen. Eine einzelne Beobachtung (ein konkretes Mapping zwischen Input- und Outputbereich) enthält nur einen kleinen Teil dieser Komplexität. Erneut bildet die Suche nach einer lokalen Erklärung durch ein einfacheres erklärendes Modell das Zentrum des Ansatzes. Diese sei als jede interpretierbare Approximation des Originalmodells definiert.
f sei wieder das Originalmodell/Black-Box-Modell, welches durch ein interpretierbares Modell g erklärt werden muss. Dabei beschränkt sich der Ansatz auf die Erklärung einer lokalen Vorhersage $f(x)$ anhand einer Beobachtung x. Ähnlich wie bspw. LIME erfolgt die Berechnung auf einem vereinfachten Input x', der mittels einer Abbildung $h_x(x') = x$ erzeugt wird. Dies ermöglicht das Erzeugen von Erklärungen, die sich auf leicht verständliche Variablen beziehen, auch wenn das eigentliche Black-Box-Modell auf komplizierteren Transformationen dieser basiert [RSG16b]. Beispielsweise kann die An-/Abwesenheit eines Wortes in einem Text oder eines Superpixels in Bildern mittels One-Hot-Encoding codiert werden, obwohl das Modell mittels Word Embeddings bzw. auf Pixelbasis arbeitet [ebd.]. Lokale erklärende Methoden suchen eine Funktion g, so dass $g(z') \approx f(h_x(z'))$ für $z' \approx x'$ gilt. Definiere:

Definition 4 ***Additive Feature Attribution Methods*** *sind ein erklärendes, lineares Modell mit binären Variablen:*

$$g(z') = \phi_0 + \sum_{i=1}^{M} \phi_i z'_i \tag{4.9}$$

mit $z' \in \{0,1\}^M$, wobei M die Anzahl vereinfachter Feature und $\phi \in \mathbb{R}$ ist.

Erklärbarkeitsverfahren die Definition 4 entsprechen, ordnen jedem Merkmal i einen Effekt ϕ_i zu. Summieren der Effekte aller Feature ergibt ungefähr die Vorhersage $f(x)$ des Originalmodells. Diese Darstellung weist einige Parallelen zur Definition linearer Modelle auf. Auch dort werden die Einflüsse von Merkmalen zur Vorhersage des Modells aufsummiert: Der Intercept b_0

entspricht gerade ϕ_0 und die ϕ_i sind im Falle eines linearen Modells gerade das Produkt aus Merkmalsausprägung und Koeffizient $b_i \cdot x_i$.
Viele erklärende Verfahren lassen sich als Modelle der Form von Definition 4 darstellen, so zum Beispiel LIME [RSG16b], DeepLIFT [SGK17] oder Shapley Regression Values [Lun+18].

Eine interessante Eigenschaft der Klasse der Additive Feature Attribution Methods ist das Vorhandensein einer eindeutigen Lösung unter der Annahme dreier wünschenswerter Eigenschaften. Diese drei Eigenschaften ähneln denen der Shapley Werte. Genau genommen handelt es sich dabei um äquivalente Darstellungen, die aus den vier Eigenschaften von Shapley (Effizienz, Symmetrie, Dummy-Spieler und Linearität) abgeleitet werden können:

- **Lokale Exaktheit:** Diese Eigenschaft garantiert, dass das erklärende Modell g für eine spezifische Beobachtung x zumindest die Vorhersage des Modells f für den vereinfachten Input x' vollständig erklärt:

  $$f(x) = g(x') = \phi_0 + \sum_{i=1}^{M} \phi_i x'_i \qquad \text{bzw.}$$

  $$\sum_{j=1}^{p} \phi_j = \hat{f}(x) - E_X(\hat{f}(X))$$

  Das entspricht gerade der Effizienz von Shapley.

- **Missingness:** Die zweite Eigenschaft ist die sogenannte Missingness. Wenn im vereinfachten Featurevektor ein Feature als *vorhanden* angezeigt wird, im Originalinput allerdings nicht vorhanden ist, so garantiert die Missingness, dass diesem Feature kein Einfluss im erklärenden Modell zugeordnet wird.

  $$x'_i = 0 \implies \phi_i = 0$$

  Diese Eigenschaft ist mehr theoretischer Natur. Die lokale Exaktheit wird mittels eines linearen Modells angegeben. Es ist theoretisch denkbar, dass x' einige Null-Einträge enthält (d.h. der Eintrag fehlte bereits als er durch h_x definiert wurde). Dann ist die lokale Exaktheit immer noch gewährleistet, egal welche Phi-Werte diese Null-Einträge erhalten. Daher wird dieser auf o gesetzt, um eine eindeutige Lösung zu finden. Das ist nicht weiter schlimm, sondern wünschenswert, da diese Einträge bereits fehlen und somit keine Auswirkungen haben. In der Praxis kann davon ausgegangen werden, dass ein Merkmal nie vollständig fehlt, es sei denn, der Wert dieses Merkmals ist über den gesamten Datensatz konstant.

- **Konsistenz (Monotonie):** Um dieser Eigenschaft zunächst Intuition zu verleihen, sei die folgende Situation gegeben:

Es wurde ein lineares Regressionsmodell trainiert und anschließend eine Erklärung nach Definition 4 erzeugt. Nun erhöht der Entwickler den Koeffizienten b_i des i-ten Merkmals. Dadurch hat er das Modell sozusagen künstlich dazu gebracht, bei dessen Prognose die Ausprägung des i-ten Merkmals stärker zu berücksichtigen. Die Konsistenz garantiert, dass beim erneuten Erzeugen von Erklärungen für dieses manipulierte Modell der Einfluss des i-ten Merkmals ϕ_i nicht niedriger wird. Etwas formaler:
Für zwei Modelle f und f' gilt, dass wenn

$$f'_x(S \cup \{i\}) - f'_x(S) \geq f_x(S \cup \{i\}) - f_x(S) \qquad \forall S \in Z \setminus \{i\},$$

dann

$$\phi_i(f', x) \geq \phi_i(f, x)$$

Dabei bezeichne S die Indexmenge von z_i, wobei die z_i ungleich Null sein müssen. Z sei die Menge aller M Inputfeature, d.h $x \in \mathbb{R}$.

Wie bereits für Shapley Values, lässt sich auch für Modelle nach Definition 4 unter Annahme der drei Axiome eine eindeutige Lösung bestimmen:

Satz 2 *Es existiert nur ein erklärendes Modell g nach Definition (4), welches die Eigenschaften der Lokalen Exaktheit, Missingness und Konsistenz erfüllt. Dieses Modell hat die folgende Form:*

$$\phi_i(f, x) \sum_{S \in Z \setminus \{i\}} \frac{|S|!(M - |S| - 1)!}{M!} [f_x(S \cup \{i\}) - f_x(S)] \tag{4.10}$$

Der Beweis dieses Satzes folgt aus dem Beweis des Satzes von Shapley, allerdings in leicht abgewandelter Form:
Young zeigt in [You85], dass die Linearität und das Dummy-Spieler-Axiom durch ein Monotonie-Axiom ersetzt werden können und der Shapley Value nach wie vor die einzige eindeutige Lösung ist. Dieses Monotonie-Axiom wurde hier durch die Bezeichnung Konsistenz ersetzt. Lundberg et. al. zeigen in ihrem Paper [LL17], dass für Machine Learning Modelle Monotonie bereits Symmetrie impliziert, so dass dann der Satz von Shapley anwendbar ist.

4.6.2 *Der SHAP Wert*

SHAP Werte sind Shapley Values eines bedingten Erwartungswertes für das Originalmodell [LL17]. Formal sind sie die Lösung für Gleichung (4.10), mit $f_x(z') = f(h_x(z')) = E[f(z)|z_S]$ und S als Indexmenge von z_i, wobei diese ungleich Null sein muss. Dies impliziert eine Abbildung der Original-Feature, auf den vereinfachten Feature-Space, so dass $h_x(z') = z_S$, wobei z_S für Feature, die nicht in S sind, keine Werte enthält. Dies stellt die meis-

ten ML-Verfahren allerdings vor große Probleme, da sie nicht mit fehlenden Input-Werten für einzelne Feature umgehen können. Daher wird $f(z_S)$ durch $E[f(z)|z_S]$ approximiert. Praktisch bedeutet dies, dass abwesende Feature - Merkmale, die dem Black-Box-Modell nicht zur Verfügung stehen sollen - durch deren geschätzten Erwartungswert ersetzt werden.

Das Ziel von SHAP - aber auch Shapley Werten - ist die Erklärung der Differenz zwischen den Vorhersagen des Black-Box-Modells für Beobachtungen x und der mittleren Vorhersage des Modells über dem Datensatz. Das wird insbesondere an der Eigenschaft der Lokalen Exaktheit bzw. Effektivität ($\sum_{j=1}^{p} \phi_j = \hat{f}(x) - E_X(\hat{f}(X))$) deutlich. Aus Sicht des Praktikers ist dies die Antwort auf die Frage: "Was war an Beobachtung x aus Sicht des Modells anders als beim Durchschnitt des Datensatzes?". Um diese Frage beantworten zu können, kommen bedingte Erwartungswerte zum Einsatz, die sequentiell auf die Merkmalsausprägungen der zu erklärenden Beobachtung bedingen. Zur Illustration das folgende Beispiel mit Abbildung 4.6. Diese stellt dar, wie für eine Beobachtung x die Differenz von der durchschnitlichen Modellvorhersage zur Vorhersage des Modells erklärt werden kann:

Beispiel zur Berechnung des SHAP Wertes

Wir arbeiten als Analyst bei einer Bank. Unsere Bank setzt zur Bearbeitung von Kreditanträgen ein Machine Learning Modell ein, dass die Ausfallwahrscheinlichkeit von Krediten berechnet. Dabei zieht das Modell vier Merkmale in seine Berechnung mit ein: das bisherige Geschäftsverhältnis zur Bank, das Alter der Person, deren Beruf und das Einkommen. Im Mittel fallen 4 Prozent der Kredite unserer Bank aus.
Alex ist ein 25-jähriger Student, der im letzten Jahr durch Aktiengeschäfte 5.035€ verdient hat und zum ersten Mal mit einem Kreditantrag an unsere Bank herantritt. Das ML-Modell prognostiziert für Alex eine Ausfallwahrscheinlichkeit von 0.61, so dass wir seinen Kreditantrag ablehnen. Alex beschwert sich daraufhin bei der Aufsichtsbehörde und möchte wissen, warum sein Kreditantrag abgelehnt wurde. Wir verwenden SHAP, um Alex eine entsprechende Erklärung zu liefern:

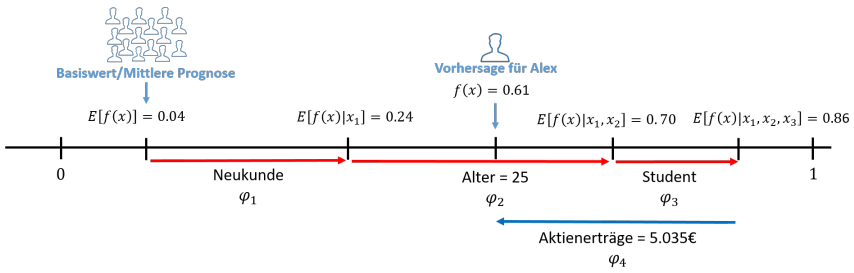


Abbildung 4.6: SHAP Values als Erklärung der Modellvorhersage (konzeptionell)

Wir kennen sowohl die Vorhersage des Black-Box-Modells f für Alex (x), nämlich $f(x) = 0.61$, als auch die mittlere Ausfallwahrscheinlichkeit $E[f(x)] = 0.04$. Das heißt, würden keine Informationen über Alex vorliegen, würden wir eine Ausfallwahrscheinlichkeit von 0.04 vorhersagen. Füttern wir das Modell nach und nach mit Informationen über Alex, verändert sich die erwartete Prognose: Die Tatsache, dass Alex ein Neukunde ist, erhöht die Wahrscheinlichkeit eines Zahlungsausfalls auf 0.24. Durch die Hinzunahme des Alters steigt diese auf 0.7 und der Beruf des Studenten führt zu einer Ausfallwahrscheinlichkeit von 0.86. Seine erfolgreichen Geschäfte an der Börse verringern die Chance eines Ausfalls auf 0.61 - die Prognose des Modells. Nun haben wir - ganz im Einklang mit dem Axiom der lokalen Exaktheit - die Differenz vollständig erklärt. Den (relativen) Einfluss eines Merkmals können wir durch die Länge der Pfeile in Abbildung 4.6 angeben. Das heißt, der Beitrag des Features Alter war $\phi_2 = 0.7 - 0.24 = 0.46$.

Anhand des Beispiels wurde deutlich, dass SHAP jedem Merkmal die Größe der erwarteten Änderung der Modellvorhersage, bei Bedingen auf dieses Merkmal, als Einfuss zuordnet. So erklären diese, wie das Modell vom *Basiswert/base value* $E[f(z)]$, der vorhergesagt würde, wenn keine Featureausprägung der aktuellen Beobachtung x bekannt wäre, zur Vorhersage $f(x)$ gelangt. Dieses Diagramm 4.6 zeigt eine einzelne Beobachtung. Wenn das Modell nichtlinear ist oder die Feature voneinander abhängen, spielt die Reihenfolge, in der die Feature zum Erwartungswert hinzugefügt wer-

den, eine große Rolle. Bezogen auf das Beispiel könnte es sein, dass es eine Wechselwirkung zwischen Neukunden und deren Alter gibt. Junge Neukunden haben tendenziell ein größeres Ausfallrisiko als ältere Neukunden. Der Effekt (und vor allem dessen Stärke) dieser Interaktion wird in Abbildung 4.6 vollständig dem Alter zugewiesen. Eine Betrachtung beider Merkmale in umgekehrter Reihenfolge hätte den Effekt stattdessen dem Merkmal Neukunde zugewiesen.
Im Falle von Nichtlinearitäten und Interaktionen ergibt sich der SHAP Wert durch Mittelung der ϕ_i-Werte über alle möglichen Reihenfolgen (vgl. hierzu Kapitel 4.5.1), um diese entsprechend zu berücksichtigen.
Die folgende Abbildung 4.7 zeigt die berechneten SHAP Values für eine konkrete Beobachtung aus dem *Adult-Datensatz*, wie sie durch das Python-Paket SHAP dargestellt werden. Dabei werden die SHAP Values zentral um die Vorhersage des Modells für die konkrete Beobachtung angeordnet.

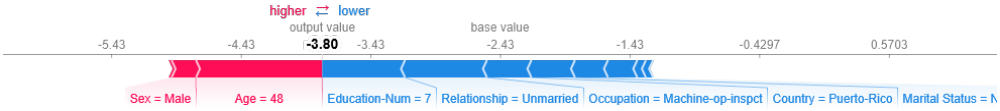


Abbildung 4.7: SHAP Values als Erklärung der Modellvorhersage (Python-Output)

In diesem Beispiel wäre *Age* die wichtigste positive Variable und *Education-Num* das wichtigste negative Merkmal für diese Beobachtung.

4.6.3 *Berechnung des SHAP Wertes*

Ähnlich wie bereits bei den Shapley Values ist die Berechnung von SHAP Werten alles andere als einfach, und es muss sich meist mit approximativen Lösungen zufrieden gegeben werden. Eine Annahme, die die Berechnung von SHAP Werten deutlich vereinfacht, ist die Unabhängigkeit der Features. Eine Annahme, die auch bei [RSG16b] oder [ŠK14] getroffen wird. Dann lässt sich die Berechnung wie folgt vereinfachen:

$$f(h_x(z')) = E[f(z)|z_S] \tag{4.11}$$

gilt nach der Definition der SHAP Values. Einsetzen der Definition des bedingten Erwartungswertes liefert

$$= \int_{\mathcal{Z}} f(z) \qquad dP(\cdot|z_S). \tag{4.12}$$

Dann ist auch eine entsprechende Integration über $z_{\bar{S}}|z_S$ möglich und es ergibt sich

$$= \int_{\mathcal{Z}} f(z) \qquad dP(z_{\bar{S}}|z_S) \tag{4.13}$$

$$= E_{z_{\bar{S}}|z_S}[f(z)] \tag{4.14}$$

Die Annahme der Unabhängigkeit der Feature ergibt

$$\approx E_{z_{\bar{S}}}[f(z)]. \tag{4.15}$$

Unter der zusätzlichen Annahme der Linearität des Modells vereinfacht sich die Berechnung zu:

$$= f([z_S, E[z_{\bar{S}}]]). \tag{4.16}$$

Wie lässt sich der bedingte Erwartungswert aus Gleichung (4.14) berechnen bzw. approximieren? Ausgehend von der bedingten Unabhängigkeit der Feature, kann der SHAP Werte direkt mit dem Verfahren von [ŠK14] geschätzt werden. Bei diesem Verfahren handelt es sich um eine stichprobenbasierte Approximation einer leicht abgewandelten Version der klassischen Shapley Werte Berechnungsformel (4.10), die in Algorithmus 2 vorgestellt wurde. Da dabei für jedes Feature eine separate Stichprobenschätzung durchgeführt werden muss, wird diese Berechnungsmethode insbesondere für viele Feature aufwendig, ist doch für jede Ausprägung jedes Merkmals einmal Algorithmus 2 auszuführen.

Lundberg et al. stellen einen alternativen Approximationsalgorithmus vor, der sich des Erklärbarkeitsverfahrens LIME bedient und mittels einer geschickten Wahl des Kerns (Shapley Kernel) deutlich weniger Funktionsauswertungen benötigt, um gute Approximationen der Shapley Werte zu berechnen [LL17]:

Ausgangspunkt dieser Überlegung ist die Klasse der Additive Feature Attribution Methods (vgl. Def. 4). LIME verwendet, wie in Kapitel 4.4 vorgestellt, ein lineares, erklärendes Modell zur lokalen Annäherung von f. Die Berechnung erfolgt dabei in einem vereinfachten binären Merkmalsraum. Zur Erinnerung nochmals die Formulierung des Regressionsproblems für LIME:

$$\eta(x) = \arg\min_{g \in G} L(f, g, \pi_x) + \Omega(g) \tag{4.17}$$

Auf den ersten Blick hat die Regressionsformulierung von LIME wenig mit der klassischen Berechnungsformel für Shapley Werte aus Gleichung (4.10) zu tun. Jedoch handelt es sich sowohl bei LIME als auch bei SHAP um eine Additive Feature Attribution Method. Darüber hinaus ist bekannt, dass diese Klasse mit Shapley Values eine eindeutige Lösung besitzt, wenn die Ei-

genschaften der lokalen Exaktheit, Missingness und Konsistenz erfüllt sind.

Es ist eine naheliegende Fragestellung, inwieweit LIME imstande ist, Lösungen zu produzieren, die die Shapley Eigenschaften erfüllen. Die Erklärungen von LIME hängen von der Wahl der Verlustfunktion L, dem Regularisierer Ω und dem lokalen Kern π_x ab. Diese Parameter werden in der Praxis heuristisch gewählt, so dass in der Regel mittels LIME eine der drei Eigenschaften verletzt wird. Lundberg et al. zeigen, dass es durch die folgende Parametrisierung möglich ist Shapley Values, mittels LIME zu schätzen:

$$\begin{aligned} \Omega(g) &= 0 \\ \pi_{x'} &= \frac{M-1}{(\binom{M}{|z'|})|z'|(M-|z'|}) \\ L(f,g,\pi') &= \sum_{z' \in Z} \left[f(h_x(z')) - g(z') \right]^2 \pi_{x'}(z') \end{aligned}$$

wobei $|z'|$ die Anzahl Elemente in x' ungleich o beschreibt. Ein Beweis findet sich im Anhang zu [LL17]. Die Autoren bezeichnen diese Methode als Kernel SHAP.

Da nach wie vor die Annahme der Linearität von g gilt und das obige L eine quadratische Verlustfunktion beschreibt, ist die LIME-Schätzung mittels linearer Regression möglich. Somit können die Shapley Werte mit Hilfe einer gewichteten linearen Regression berechnet werden. Das ermöglicht eine regressionsbasierte, modell-agnostische Schätzung von SHAP Werten. Die Schätzungen der SHAP Werte via linearer Regression haben eine deutlich bessere Effizienz in Bezug auf die benötigten Funktionauswertungen als die Verwendung der klassischen Shapley Gleichungen, wie Abbildung 4.8 zeigt [LL17]:

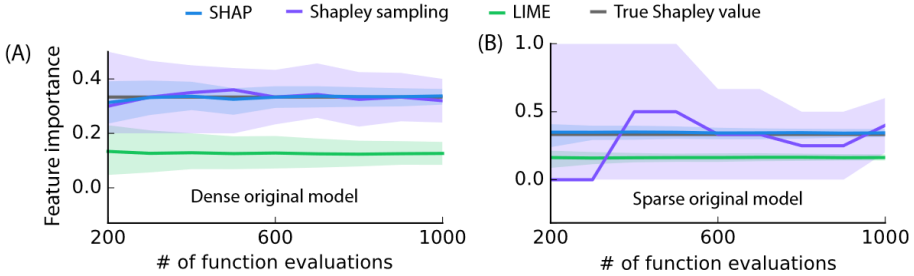


Abbildung 4.8: Vergleich von drei Additive Feature Attribution Methods (Quelle: [LL17])

Dabei wurden die folgenden drei Erklärungen verwendet: Kernel-SHAP (unter Verwendung eines Lasso-Ansatzes), Shapley Permutation Sampling nach Algorithmus 2 und LIME. Die Schätzung der Wichtigkeit eines Merkmals für eine Beobachtung wird für zwei verschiedene Modelle A und B dargestellt. Auf der x-Achse findet sich die steigende Anzahl an Funktions-

auswertungen des Verfahrens. Als Schätzung für die Varianz der Approximationen sind das 10. und 90. Perzentil eingezeichnet (farbige Flächen). Bei Modell A handelt es sich um einen Entscheidungsbaum, der alle 10 Input-Merkmale verwendet. Eingabe B - ebenfalls ein Entscheidungsbaum - verwendet hingegen nur 3 von 100 Feature.

Es ist zu erkennen, dass LIME - aufgrund der Robustheit der linearen Regression - Schätzungen mit deutlich geringerer Varianz produziert als die klassische Monte Carlo Schätzung des Shapley Wertes. Allerdings konvergiert die LIME-Schätzung gegen einen falschen Wert, da LIME die Eigenschaft der Konsistenz nicht erfüllt. Der Ansatz über Monte Carlo Methoden konvergiert zwar gegen den richtigen Wert, allerdings benötigt er dabei sehr viele Funktionsaufrufe - also eine große Anzahl an Iterationen M in Algorithmus 2. Kernel-SHAP kombiniert sozusagen die Vorzüge der beiden Ansätze. Es entstehen Schätzungen des korrekten Shapley Wertes mit einer deutlich geringeren Varianz der Schätzung, so dass deutlich weniger Funktionsauswertungen bzw. Samples nötig sind, um eine gute Approximation zu erhalten.

Bei Modell B ist sichtbar, wie sich die Verwendung der Lasso-Regularisierung positiv auf die Varianz auswirkt - die irrelevanten 97 Merkmale werden gar nicht erst in Betracht gezogen, um Erklärungen zu erzeugen. Der Ansatz über Monte Carlo Simulationen benötigt, aufgrund des großen Anteils irrelevanter Merkmale deutlich länger, bis sich die Varianz der Schätzung verkleinert.

Lundberg et al. haben darüber hinaus schnelle, modell-spezifische Algorithmen für baumbasierte Modelle entwickelt, die die Berechnung der SHAP Werte erheblich beschleunigen. Für eine detailierte Herleitung sei an dieser Stelle auf ihre Arbeit [LEL18] verwiesen.

4.6.4 *Von lokalen zu globalen Modellerklärungen*

Insbesondere bei baumbasierten Verfahren ist die Darstellung der globalen Feature Importance in Form von Balkendiagrammen ein gängiges Vorgehen, um die globale Bedeutung einzelner Merkmale darzustellen. Aber auch für andere ML-Verfahren ist diese Art der Darstellung möglich (vgl. hierzu Kapitel 4.3). Eine weitere Analysemöglichkeit ist es, die *durchschnittliche* Auswirkung der Änderung eines Features über den gesamten Datensatz zu betrachten - sogenannte Partial Dependence Plots (vgl. dazu Kapitel 4.1).
SHAP Werte hingegen sind zunächst eine lokale Erklärungsmethode. Das Berechnen dieser für jede Beobachtung bzw. eine ausreichend große Stichprobe (in der Regel sind bereits ungefähr 2500 Beobachtungen ausreichend) aus dem Datensatz ermöglicht ebenfalls informative Visualisierungen zum

globalen Modellverhalten.

SHAP Summary Plots sind eine alternative Darstellung zu den klassischen Balkendiagrammen der globalen Feature Importance [LEL18]: Diese klassischen Diagramme liefern zwar eine Intuition über die relative Wichtigkeit eines Features, aber keinerlei Information, wie stark der Einfluss des Features abhängig von dessen Ausprägung ist. Auch über die Größe und Verteilung des Merkmalseinflusses sind keine Aussagen möglich. *SHAP Summary Plots* hingegen liefern all das und nutzen dabei die individuellen (im Sinne von für jede Beobachtung) Feature Beiträge der lokalen Erklärungen. Dabei gehen sie wie folgt vor:
Zunächst sortieren sie die Merkmale $j \in J$ nach ihrem globalen Einfluss auf das Modell. Dies erfolgt über die Berechnung des mittleren Absolutbetrags eines jeden Merkmals $\sum_j = 1^N |\phi_i^{(j)}|$. Anschließend wird der SHAP Wert $\phi_i^{(j)}$ jeder Beobachtung horizontal dargestellt. Dabei werden mehrfach vorkommende Werte vertikal aufeinander gestapelt. Diese vertikale Stapelung erzeugt eine Darstellung, die mit Violin-Plots vergleibar ist. Jeder Punkt wird entsprechend der Ausprägung seines Merkmales codiert, von niedrig (blau) bis hoch (rot). In Abbildung 4.9 findet sich ein beispielhafter *SHAP Summary Plot* für eine Gradient Boosting Machine auf dem Adult-Datensatz. Mehr zu diesem Modell in Kapitel 8.2.

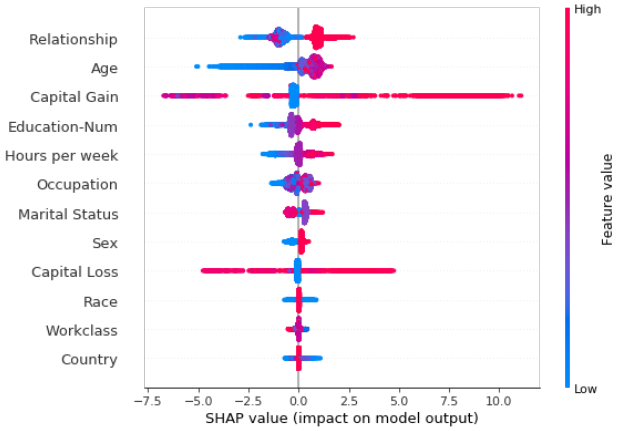


Abbildung 4.9: *SHAP Summary Plot* für eine Gradient Boosting Machine auf dem Adult-Datensatz

Anhand des Plots ist beispielweise zu erkennen, dass *Relationship* die wichtigste Variable ist und *Age* vor allem bei jungen Menschen einen stark negativen Einfluss auf die Modellprognose (Wahrscheinlichkeit für ein Einkommen $>$ 50.000$) hat. Junge Menschen haben in diesem Modell eine deutlich niedrigere Wahrscheinlichkeit für ein hohes Jahreseinkommen. Die Variable *Capital Gain* hat anscheinend nur bei wenigen Personen einen Einfluss, wenn sie allerdings einen Einfluss auf die Modellvorhersage hat, ist dieser sehr stark.

Es sollte deutlich geworden sein, dass ein detailreicherer Einblick in das Modell möglich ist, als dies bei einem klassischen Feature Importance Plot der Fall ist.

Wie bereits in Gleichung 4.1 in Kapitel 4.1 beschrieben, stellen Partial Dependence Plots (PDP) die erwartete Vorhersage eines Modells dar, wenn die Ausprägung eines Features (oder einer Gruppe von Featuren) konstant gehalten wird. Die Werte der fixierten Variablen werden variiert und die daraus resultierende erwartete Modellvorhersage geplottet. Das Darstellen der Änderung des Modelloutputs bei Manipulation des Merkmals zeigt, wie sehr die Modellvorhersage von diesem Feature abhängt. Mittels SHAP Werten können ähnliche, sehr aufschlussreiche Darstellung erzeugt werden, sog. *SHAP Dependence Plots*.

Diese stellen den SHAP Wert eines Features auf der y-Achse und die Ausprägung des Features auf der x-Achse dar. Durch wiederholtes Darstellen diese Werte für viele Individuen aus dem Datensatz ist es möglich, den Zusammenhang zwischen Merkmal und dessen Einfluss auf die Vorhersage zu beschreiben. Während klassische Partial Dependence Plots lediglich Linien erzeugen, erfassen *SHAP Dependence Plots* die vertikale Streuung aufgrund von Interaktionseffekten im Modell - ähnlich wie ICE-Plots. Diese Effekte lassen sich durch Einfärben aller Punkte anhand der Werte eines interagierenden Features (vgl. Abbildung 4.10) visualisieren.

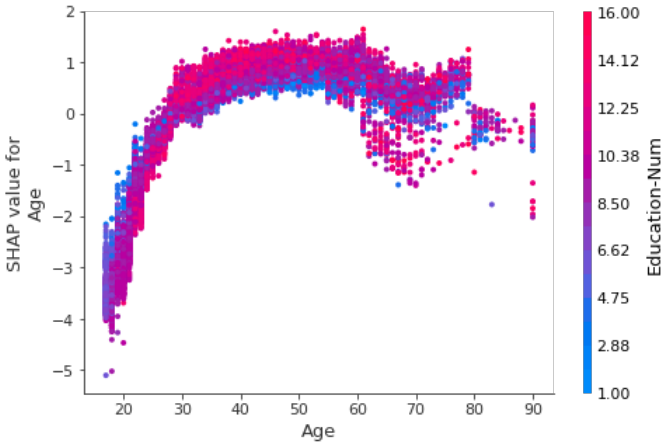


Abbildung 4.10: *SHAP Dependence Plot* für eine Gradient Boosting Machine auf dem Adult-Datensatz

In diesem Fall fällt zunächst auf, dass Alter für Personen unter 30 einen negativen Einfluss auf die Prognose hat. Danach ist der Einfluss positiv, bis er ab 60 Jahren stark zu streuen beginnt. Zusätzlich ermöglicht die farbliche Codierung Aussagen über den Einfluss von Alter über verschiedene Bildungsschichten hinweg und im Vergleich zueinander. So haben Menschen mit einem niedrigen Bildungsniveau (hier blau codiert) in jungen Jahren eine bessere Prognose als Personen mit langer Ausbildungsdauer (rot). Ab 30

Jahren kippt dieses Verhältnis. Dann ist es besser - in Bezug auf die Prognose - eine gute Schulausbildung genossen zu haben. Menschen mit mittlerem Bildungsabschluss (lila) bewegen sich stets zwischen den beiden anderen Gruppen und folgen im wesentlichen dem Trend der Kurve.

Die marginalen Beiträge werden typischerweise auf die Input-Feature verteilt - einen für jedes Feature. Durch das Aufteilen dieses Einflusses in Haupt- und Interaktionseffekte mit anderen Merkmalen ergibt sich ein noch tieferes Modellverständnis. Bereits bei der Beschreibung von Abbildung 4.10 wurde deutlich, dass das Analysieren von Interaktionen interessante Erkenntnisse zu Tage fördert. Die Berechnung resultiert in einer Matrix mit den Einflüssen von paarweisen Featurekombinationen auf die Modellvorhersage als Einträge, wobei die Hauptdiagionale die Haupteffekte beschreibt. Diese Berechnung wird erneut durch ein Konzept der Spieltheorie ermöglicht: mit Hilfe des sogenannten Shapley interaction index [FKM06]:

$$\Phi_{i,j} = \sum_{S \subseteq N \setminus \{i,j\}} \frac{|S|!(M - |S| - 2)!}{2(M-1)!} \nabla_{ij}(S) \tag{4.18}$$

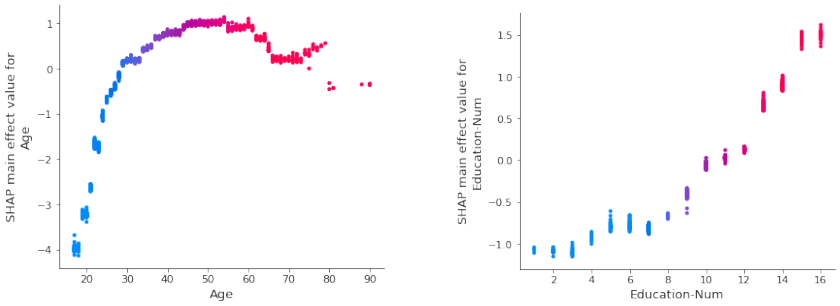
wobei $i \neq j$ und

$$\nabla_{ij}(S) = f_x(S \cup \{i,j\}) - f_x(S \cup \{i\}) - f_x(S \cup \{j\}) + f_x(S) \tag{4.19}$$

In Gleichung 4.18 wird die *SHAP Interaction* zwischen Feature i und Feature j gleichmäßig auf jedes Merkmal aufgeteilt, so dass $\Phi_{i,j} = \Phi_{j,i}$ gilt und der gesamte Interaktionseffekt durch $\Phi_{i,j} + \Phi_{j,i}$ gegeben ist. Die Haupteffekte für eine Vorhersage können dann als die Differenz zwischen dem SHAP Wert und der *SHAP Interaction* für ein Merkmal definiert werden:

$$\Phi_{i,i} = \phi_i - \sum_{j \neq i} \Phi_{i,j}$$

Ein Algorithmus zur effizienten Berechnung der *SHAP Interaction* findet sich bei [LEL18], der allerdings erneut auf baumbasierte Ensemble beschränkt ist. Die Kombination von *SHAP Dependence Plots* mit SHAP Interaktionswerten kann helfen, globale Interaktionsmuster aufzudecken. Die folgende Abbildung zeigt, wie der Einfluss des Features aus Abbildung 4.10 in die beiden Haupteffekte und den Interaktionseffekt aufzuteilen ist. Insbesondere Abbildung 4.11 (c) macht dies deutlich, ist hier doch genau der Zusammenhang zwischen Alter und Bildungsniveau zu erkennen, der bereits in Abbildung 4.10 zu vermuten war.



(a) (b)

(c)

Abbildung 4.11: Aufteilung des marginalen Beitrags in Main-Effect von *Age* (a), *Education Num* (b) und Interaction-Effect (c)

Zusammenfassung SHAP Werte

SHAP Werte sind Shapley Values eines bedingten Erwartungswertes über das Black-Box-Modell f. Sie sind folglich die Lösung für Gleichung (4.10), mit $f_x(z') = f(h_x(z')) = E[f(z)|z_S]$ und S als Indexmenge von z_i, wobei diese ungleich Null sein müssen.

Vorteile:

- Alle Vorteile, die für Shapley Values im Allgemeinen gelten, übertragen sich auf SHAP Werte.
- Ein großer Vorteil von SHAP liegt in der Schätzung der Werte über LIME und die lineare Regression. Durch die deutlich effizienteren Algorithmen für baumbasierte Verfahren wird es möglich, die SHAP Werte für sehr viele Beobachtungen zu berechnen.
- Die effiziente Berechnung ermöglicht auch das Betrachten von Interaktionen zwischen Merkmalen, wodurch ein noch tieferer Eindruck vom Verhalten des Modells möglich wird.
- Im Gegensatz zu *normalen* Shaple Values können mittels SHAP aufgrund der Verwendung der Merkmalstransformation auch selektive Erklärungen mit weniger Merkmalen erzeugt werden.

Nachteile:

- Wie schon bei Shapley Werten liefert auch SHAP nur eine Zahl und kein lokal gültiges Modell, mittels dessen sich andere Beobachtungen prognostizieren lassen.
- Wie viele Verfahren leidet auch SHAP daran, dass durch das Erzeugen von permutierten Beobachtungen unrealistische Datenpunkte entstehen können.
- Da die Schätzung mittels LIME erfolgt, übertragen sich auch einige der Nachteile aus Kapitel 4.4.

Teil II

FEATURE ENGINEERING MITTELS BLACK-BOX-MODELLEN

5

PROBLEMSTELLUNG, ANNAHMEN UND MODELLAUSWAHL

Nachdem zahlreiche Verfahren zur Generierung von Erklärungen erläutert wurden, befasst sich dieser Teil der Arbeit mit der eigentlichen Fragestellung - dem Feature Engineering mittels den Erklärungen von Black-Box-Modellen. Dazu wird in Abschnitt 5.1 zunächst die Problemstellung dargelegt, bevor daran anschließend die Annahmen des entwickelten Methodik, sowie die Auswahl und Evaluation der in dieser Arbeit verwendeten Modelle ins Zentrum rückt.

5.1 PROBLEMSTELLUNG

Ziel dieser Arbeit ist es, zu untersuchen, inwieweit sich die Performanz von erklärbaren Modellen mittels Erklärungen von performanteren Black-Box-Modellen verbessern lässt, ohne dabei deren intrinsische Erklärbarkeit zu stark einzuschränken. Bei vielen Problemen weisen komplexen ML-Verfahren eine, im Vergleich zu White-Box-Modellen, bessere Vorhersagegüte auf nicht-vor-verarbeiteten Daten vor. In dieser Arbeit sollen Methoden entwickelt werden mittels deren Informationen aus den Black-Box-Verfahren extrahiert und in erklärbare Modelle integriert werden können.
Zunächst ist es aber nötig zu definieren welche Verfahren als erklärbar gelten. Daran anschließend müssen die Annahmen der White-Box-Modelle mit ihren Auswirkungen auf die Performance analysiert werden. Daraus lassen sich Gründe für den Performanceunterschied ableiten, welche anschließend in den Modellbildungsprozess eines White-Box-Modells integrierbar sind. Dies ist zunächst einmal unabhängig vom betrachteten Problemtyp (Regressions- oder Klassifikationsproblem).

5.2 ANNAHMEN

Die beiden Methoden, die in Kapitel 6 und 7 erarbeitet werden, sind mit impliziten Annahmen verbunden, die hier kurz erläutert werden:

- Die überlegene Performance des Black-Box-Modells gegenüber erklärbaren Ansätzen ist Grundannahme der Arbeit. Ist dies nicht der Fall, beschreibt das erklärbare Modell mit seinen vereinfachenden Annahmen den tatsächlichen Zusammenhang genau so gut oder sogar besser als das Black-Box-Modell. Dann besteht keine Notwendigkeit, Feature Engineering mittels des komplexen, maschinellen Lernverfahrens zu betreiben.

- Dieses Vorgehen geht zudem davon aus, dass ein Modell umso erklärbarer ist, desto weniger Merkmale es zur Entscheidungsfindung heranzieht. Daher wird es unter anderem ein Ziel sein, Merkmale mit niedriger prädiktiver Power aus den Modellen zu entfernen.
- Der Original-Merkmalsraum ist immer die erklärbarste Repräsentation der Daten. Mit jeder Transformation des Datenraums, egal ob nur ein einzelnes Feature transformiert wird oder aus den Daten neue Merkmale generiert werden, verringert sich die Erklärbarkeit des Problems. Dies verringert die Nachvollziehbarkeit aus Sicht des Laien. Wie die Erklärbarkeit eines Modells gemessen werden kann und was diese bedeutet, ist eine nach wie vor unbeantwortete Forschungsfrage (vgl. hierzu u.a. [Lip18] und [Gil+18]), so dass in dieser Arbeit die Beschreibung des Problems ausreichend sein muss.

Zusätzlich zu diesen Annahmen, kommen in der Arbeit (aus zwei Gründen) ausschließlich baumbasierte Ensemble-Modelle zum Einsatz:

Die Berechnung der SHAP Werte für baumbasierte Ensemble kann auf Grund der speziellen Architektur des Ensembles deutlich beschleunigt werden. Lundberg et al. haben dafür eigene modell-spezifische Algorithmen entwickelt [LEL18]. Insbesondere ist die Berechnung der SHAP Interaction Werte bislang nur für diesen Modelltyp möglich. Neben diesem praktischen Argument spielt auch der Rahmen der Masterarbeit eine Rolle. Eine Betrachtung weiterer Black-Box-Methoden, wie Neuronaler Netze oder Support Vector Machines, würde den Rahmen sprengen. An dieser Stelle muss betont werden, dass das hier vorgestellte Vorgehen zum Feature Engineering mit Black-Box-Modellen an keiner Stelle Annahmen über die Struktur des Black-Box-Modells trifft. Solange die Berechnung von Modellerklärungen möglich ist, lässt sich das Vorgehen auch auf andere ML-Verfahren übertragen.

5.3 MODELLAUSWAHL UND -EVALUATION

Bevor Methodiken zum systematischen Feature Engineering mittels Erklärungen von Black-Box-Modellen entwickelt werden können, muss zunächst geklärt sein, welche Eigenschaften ein Modell erklärbar machen und welche ML-Verfahren als White-Box-Modell kategorisiert sind. Dies geschieht in Abschnitt 5.3.1. Darüber hinaus standardisiert Abschnitt 5.3.2 die Vorverarbeitung der Datensätze in Bezug auf den Umgang mit fehlenden oder unvollständigen Beobachtungen und die Codierung kategorieller Variabeln. Die verwendeten Evaluationsmetriken sowie Ansätze, die Methoden aus Kapitel 6 und 7 zu evaluieren, sind Thema der Abschnitte 5.3.3 und 5.3.4.

5.3.1 *Was sind erklärbare ML-Algorithmen?*

Der einfachste Weg, Interpretierbarkeit der Modelle zu gewährleisten, besteht in der Verwendung einer Teilmenge von Algorithmen, die interpretierbare Modelle erzeugen. Denn die beste Erklärung eines Modells ist immer

das Modell selbst [LL17]. Die Lineare und Logistische Regression, sowie der Entscheidungsbaum sind häufig verwendete interpretierbare Modelle - die auch von Aufsichtsbehörden (im Finanzbereich) seit langer Zeit akzeptiert werden. Aber was unterscheidet diese Modelle von anderen Algorithmen?

Im wesentlichen gibt es vier Eigenschaften, die die Idee der Interpretierbarkeit eines Modells wiedergeben können: die Anzahl an Merkmalen, Linearität, Monotonie und das Vorhandensein von Interaktionen [Mar+11] bzw. der Grad der Interaktion. Erstere liegt meist in der Hand des Entwicklers eines Modells, während die drei verbleibenden Eigenschaften abhängig vom verwendeten Modelltyp sind:

- Ein Modell ist linear, wenn die Zuordnung zwischen Merkmalen und der Zielgröße linear modelliert wird.
- Ein Modell mit Monotoniebeschränkung stellt sicher, dass die Beziehung zwischen einem Merkmal und dem Zielergebnis über den gesamten Bereich des Merkmals immer in die gleiche Richtung geht: Eine Erhöhung des Merkmalswerts führt entweder immer zu einer Erhöhung oder immer zu einer Verringerung der Zielgröße.
- Einige Modelle können automatisch Interaktionen zwischen Merkmalen darstellen. Darüber hinaus sind Entwickler imstande, in jeden Modelltyp Wechselwirkungen manuell zu integrieren, indem sie eine entsprechende Funktion definieren. Interaktionen können die Vorhersageleistung verbessern, beeinträchtigen allerdings - insbesondere wenn mehrere Wechselwirkungen modelliert werden - die Interpretierbarkeit des Modells.

Die folgende Tabelle (angelehnt an [Molsu]) zeigt den Zusammenhang zwischen Modell und dessen Eigenschaften:

Algorithmus	**Linear**	**Monoton**	**Interaktion**
Lin. Regression	✓	✓	✗
Log. Regression	(✓)	✓	✗
Entscheidungsbaum	✗	(✓)	✓

Tabelle 5.1: Eigenschaften erklärbarer Modelle

Auf den ersten Blick scheint der Entscheidungsbaum eine vergleichsweise geringe Erklärbarkeit zu haben, ist er doch nur in manchen Fällen monoton und nie linear. Die Tatsache, dass er immer Interaktionen berücksichtigt - sofern diese vorhanden sind - erschwert die Interpretation zusätzlich. Die Struktur des Baums garantiert allerdings ein gewisses Maß an Transparenz, solange die Tiefe des Baums nicht zu groß wird. In zahlreichen experimentellen Studien konnte gezeigt werden, dass auch Laien imstande sind, Entscheidungen eines kurzen Entscheidungsbaums nachzuvollziehen (vgl. u.a.

[VAB07], [Mar+08] und [RSG16a]). Um eine möglichst gute Erklärbarkeit des Entscheidungsbaums zu gewährleisten, werden in dieser Arbeit ausschließlich Bäume mit einer maximalen Tiefe von Vier betrachtet.

Die Darstellungen in Tabelle 5.1 sind mit Annahmen bezüglich der Eigenschaften eines Modelltyps und den Möglichkeiten des für diesen Typ entwickelten Feature Engineerings verbunden. In dieser Arbeit erfolgt eine isolierte Betrachtung von Modelltyp und den für diesen entwickelten Erweiterungen bzw. Vorverarbeitungsschritten. Insbesondere für die lineare und logistische Regression existieren zahlreiche Ansätze, die Schwächen des Verfahrens durch ein aufwendiges Feature Engineering teilweise auszumerzen:

Zur Auswahl von relevanten Merkmalen stehen Verfahren wie *Forward-*, *Backward-* oder *Stepwise-Selection* zur Verfügung. Je nach Methodik werden dabei systematisch Merkmale aus dem Modell entfernt bzw. hinzugefügt und der Einfluss auf die Performance der resultierenden Modelle betrachtet (mehr zu diesen Verfahren findet sich u.a. bei [Jam+14] oder [HTF09]). Dies entspricht einem systematischen Ausprobieren verschiedener Merkmalsräume für das Training der Verfahren. Neben dem - abhängig von der Menge der zur Verfügung stehenden Merkmale - großen Aufwand, für jeden Merkmalsraum ein neues Modell zu berechnen, werden bei diesem Vorgehen keine Interaktionen zwischen den Featuren berücksichtigt, da Merkmale immer isoliert hinzugefügt bzw. entfernt werden. Darüber hinaus besteht die Möglichkeit, Regularisierungsverfahren wie z.B. Lasso-Regression einzusetzen.

Zur Integration von Interaktionen existiert ein ähnliches Vorgehen. Zunächst werden alle möglichen paarweisen Interaktionen zwischen den Merkmalen im Modell gebildet. Anschließend erfolgt die Anwendung eines der gerade vorgestellten Verfahren zur Merkmalsselektion, um die statistisch signifikanten Interaktionen in das Modell aufzunehmen. Neben der Tatsache, dass es sich erneut um ein systematisches Ausprobieren handelt, wird der Entwickler des Modells zusätzlich mit der Herausforderung konfrontiert, die in Frage kommenden Interaktionen zu spezifizieren. Je mehr Merkmale und Transformationen dieser zugelassen werden, umso mehr potenzielle Interaktionseffekte entstehen, die einzeln überprüft werden müssen.

Im späteren Verlauf der Arbeit wird die Integration von nichtlinearen Zusammenhängen eine große Rolle spielen. Derartige Effekte können durch die lineare bzw. logistische Regression ebenfalls nicht eigenständig detektiert werden. Erneut ist es möglich verschiedene nichtlinearer Transformationen des Merkmals über ein Selektionsverfahren systematisch auszuprobieren. Um potenzielle Transformationen zu erkennen, kann eine Residuenanalyse herangezogen werden. Mit den sogenannten *Weight of Evidenz* [JG60] besteht eine alternative Möglichkeit Merkmale so zu diskretisieren, dass der nichtlineare Zusammenhang approximiert wird.

All diese Verfahren sind fehleranfällig, da nie alle möglichen Transformationen, Interaktionen bzw. Konstellationen von Merkmalen betrachtet werden und sehr aufwendig. Wenn in dieser Arbeit von einem linearen bzw. logistischen Regressionsmodell die Rede ist, impliziert dies, dass keiner dieser Vorverarbeitungsschritte durchgeführt wurde. Es ist vielmehr Ziel dieser Arbeit, mittels den Erklärungen eines Black-Box-Modells, ein alternatives Vorgehen zum Featuren Engineering von White-Box-Modellen zu entwickeln.

5.3.2 *Aufbereitung der Daten*

Da die Frage des unterstützenden Feature Engineerings durch den Einsatz von Black-Box-Modellen im Zentrum der Arbeit steht, wird möglichst wenig Vorverarbeitung der Daten durchgeführt. Das heißt, lediglich notwendige Verarbeitungsschritte, die ein Trainieren der Modelle gewährleisten, finden Anwendung. Der Umgang mit unvollständigen Beobachtungen und die Codierung kategorieller Variablen stehen dabei im Vordergrund.
Prinzipiell gibt es im Umgang mit unvollständigen Dateninstanzen verschiedene Möglichkeiten. Das Vorgehen ist abhängig vom Anwendungsfall und der Menge an verfügbaren Daten:

- Stehen sehr viele, repräsentative Beobachtungen zur Verfügung, ist es üblich, unvollständige Dateninstanzen aus der Stichprobe zu entfernen. Dabei wird in Kauf genommen, unter Umständen relevante Informationen aus den Daten zu entfernen.

- Je nach der Beschaffenheit eines Merkmals kann das Fehlen der Ausprägung eine eigene Kategorie mit modellrelevanten Informationen darstellen. Wenn ein Kunde eines Online-Shops freiwillig weitere Angaben zu seiner Person (über Pflichtfelder hinaus) macht, könnte dies die Wahrscheinlichkeit einer betrügerischen Absicht des Kunden reduzieren. Das Gegenteil wäre allerdings ebenso denkbar.

- Darüber hinaus sind unvollständige Daten mittels sogenannter *Imputation* vervollständigbar. Ein besonders einfacher Ansatz ist das Ersetzen der fehlenden Ausprägung durch einen beliebigen Verteilungsparameter, wie beispielsweise den Mittelwert oder Median. Dabei werden keine Korrelationen zwischen Merkmalen berücksichtigt, so dass es zu unrealistischen Beobachtungen kommen kann. Abhängig von der Anzahl an unvollständigen Beobachtungen kann dieses Vorgehen die Trennschärfe eines Merkmals erheblich beeinflussen.

In dieser Arbeit werden je nach Problemstellung und der Größe des Datensatzes unvollständige Instanzen entweder entfernt oder als eigene Kategorie behandelt.

Die Codierung kategorieller Variablen kann die Performance eines ML-Modells stark beeinflussen. Während viele Verfahren ein sogenanntes One-Hot-Encoding - das Erzeugen einer Dummy-Variablen für jede Ausprägung

des kategoriellen Merkmals - bevorzugen, führt diese Codierung bei baumbasierten Verfahren schnell zu den folgenden beiden Problemen:

- Gerade bei kategoriellen Variablen mit vielen Klassen führt One-Hot-Encoding zu vielen dünnbesetzten Dummy-Merkmalen, so dass kontinuierliche Variablen in der Regel eine höhere Bedeutung während des Modelltrainings erhalten. Eine einzelne Ausprägung der kategoriellen Variable muss einen vergleichsweise hohen Reinheitsgewinn erbringen, um als Splitkriterium herangezogen zu werden. Dies kann die Vorhersagegüte negativ beeinflussen [HTF09].
- Darüber hinaus kann es durch One-Hot-Encoding zu entarteten bzw. dünn besetzen Entscheidungsbäumen kommen, da eine Merkmalsausprägung des kategoriellen Features nach der anderen als Splitvariable herangezogen wird.

Für baumbasierte Verfahren ist ein sogenanntes Label Encoding oft die bessere Wahl. Da in dieser Arbeit ausschließlich Gradient Boosting Machines als Black-Box-Modelle zum Einsatz kommen (vgl. hierzu auch Kapitel 5.3.1) und diese in der Regel der Performance eines linearen bzw. logistischen Regressionsmodells gegenübergestellt werden, finden immer beide Codierungen parallel Verwendung.

5.3.3 *Evaluation von ML-Modellen*

Es gibt verschiedene Möglichkeiten, die Performance eines maschinellen Lernverfahrens zu messen - dies ist u.a. abhängig vom bearbeiteten Problemtyp. In dieser Arbeit werden ausschließlich Regressions- bzw. Klassifikationsprobleme betrachtet.

5.3.3.1 *Evaluation von Klassifikationsproblemen*

Im Falle von Klassifikationsproblemen spielen zwei Dinge bei der Beurteilung der Modellqualität eine entscheidende Rolle: Zum einen die Verteilung der Zielklassen und zum anderen die Wahl der Cut-Off-Wahrscheinlichkeit, die nötig ist, um eine Beobachtung einer Klasse zuzuordnen:
Eine leichte Unbalanciertheit der Daten muss später bei der Evaluation des Modells berücksichtigt werden. Die Genauigkeit des Modells ist nach unten durch diese beschränkt. Gehören beispielsweise 70 Prozent der Beobachtungen einer Zielklasse an, so kann bereits ein generisches Modell, das einfach alle Beobachtungen in diese Kategorie einordnet, eine Genauigkeit von 70 Prozent erzielen. Wird in solchen Fällen die Genauigkeit als Evaluationsmaß herangezogen, ergibt sich zusätzlich die Herausforderung, die optimale Cut-Off-Wahrscheinlichkeit zu bestimmen, bei dem die Beobachtung einer Klasse zuzuordnen ist. Die Area-Under-the-Curve (AUC) der ROC-Kurve (Receiver Operating Characteristics) hingegen ist ein Evaluationsmaß, das unabhängig von der Verteilung der Zielgröße und ohne konkrete Cut-Off-Wahl ein Modell evaluiert.

Die ROC-Kurve ist eine Wahrscheinlichkeitskurve und deren AUC zeigt wie gut ein Modell in der Lage ist, zwischen Klassen zu unterscheiden. Dazu wird für jeden möglichen Schwellwert die Sensitivität (Richtig-Positiv-Rate) und Spezifität (Falsch-Pos-itiv-Rate) ermittelt. In einem Diagramm wird die Sensitivität als Ordinate und Spezifität als Abszisse gegeneinander abgetragen. Eine AUC von 1.0 entspricht einer perfekten Klassentrennung des Modells unabhängig von der gewählten Cut-Off-Wahrscheinlichkeit. Bei einer AUC von 0.5 hingegen weist das Modell keinerlei Trennkraft auf, und es entscheidet quasi zufällig über die Klassenzugehörigkeit. Mehr zu ROC-Kurven findet sich bei [Pow07].

5.3.3.2 *Evaluation von Regressionsproblemen*

Die Evaluation von Regressionsproblemen erfolgt häufig anhand der Wurzel der mittleren quadrierten Fehlerterme (Root Mean Squared Error - RMSE). Dieses Maß ist mit der Annahme verbunden, dass die Residuen unverzerrt normalverteilt sind. Es berechnet sich wie folgt:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(\hat{y}_i - y_i)^2}{N}}$$

Bei der Interpretation des RSME gilt es einiges zu beachten:

- Der RMSE ist stark durch Ausreißer in den Daten beeinflussbar.
- Der RMSE kann immer nur im Verhältnis zum Wertebereich der Zielgröße interpretiert werden. Er ist nicht über verschiedene Probleme hinweg vergleichbar.
- Im Umterschied zum mittleren absoluten Fehler werden große Abweichungen zwischen Prognose und tatsächlicher Ausprägung stärker bestraft.

Sinkt der RMSE, verbessert sich die Leistung des Modells, allerdings ist die Interpretation nicht so intuitiv. Im Falle eines Klassifizierungsproblems, das mittels ROC-AUC evaluiert wurde, lässt sich Performance immer in Bezug zu einem Zufallsmodell darstellen, das als Benchmark dient. Der RMSE hat hingegen keinen Benchmark.

Die Verwendung der adjustierten R-Squared Metrik (auch Bestimmtheitsmaß) erleichtert den Modellvergleich. Die Berechnung ergibt sich durch

$$\bar{R}^2 = 1 - (1 - R^2)\left[\frac{N-1}{N-(k+1)}\right] \qquad \text{mit} \qquad R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(\bar{y}_i - \hat{y}_i)^2}$$

wobei N die Anzahl an Beobachtungen und k die Anzahl der Merkmale im Regressionsmodell beschreibt. Ein Modell, das gleich der Basislinie performt, würde ein R-Squared von 0 erhalten. Je besser das Modell, desto höher der R^2-Wert. Das unadjustierte R-Squared berücksichtigt nicht die Tatsache, dass durch das Hinzufügen neuer Merkmale R^2 nicht fallen kann.

Im Gegensatz zum R^2 besteht beim adjustierten Bestimmtheitsmaß $\bar{R}^2$ die Möglichkeit, dass dieses auch Werte kleiner Null bzw. größer Eins annehmen kann. Dies ist bei der Interpretation zu berücksichtigen. Die Evaluation des Feature Engineerings bei Regressionsproblemen erfolgt daher immer anhand des adjustierten $\bar{R}^2$.

5.3.4 *Evaluation des Feature Engineerings*

Nachdem mittels den später in Kapitel 6 und 7 vorgestellten Methoden ein aufwendiges Feature Engineering betrieben wurde, stellt sich die Frage, wie der Nutzen des Vorgehens zu evaluieren ist.

Während der beiden Vorgehen wird stets die mittlere Performance einer k=5 Kreuzvalidierung als Schätzung der Trennschärfe eines Modells - und damit auch als Schätzung seiner Leistungsfähigkeit - herangezogen. Dieses eher heuristische Vorgehen genügt, um einen ersten einfachen Vergleich zwischen Machine Learning Modellen durchzuführen und den Nutzen einer Transformationen des Merkmalsraums abzuschätzen. Allerdings ergibt sich daraus noch keine Gewissheit, dass die größere Leistungsfähigkeit eines Modells nicht zufällig - beispielsweise durch eine günstige Aufteilung des Datensatzes in Train- und Testsample - zustande gekommen ist. Um diese Frage abschließend beantworten zu können, muss ein statistischer Test durchgeführt werden. Mehr zu statistischen Testverfahren, deren Annahmen und Anwendung findet sich unter anderem bei [Mono6] und [Fis25].

Leider ist die Wahl eines geeigneten statistischen Testverfahrens für den Vergleich der Leistungsfähigkeit von maschinellen Lernverfahren herausfordernd. Anhand des Vergleichs zweier Klassifikatoren für ein binäres, balanciertes Klassifikationsproblem mittels der ROC-AUC lassen sich die Herausforderungen gut beschreiben.

Eine naheliegende Evaluation ist die gerade beschrieben 5 bzw. 10-fache Kreuzvalidierung. Unter der Annahme einer normalverteilten Stichprobenschätzung, lässt sich die Leistungsfähigkeit eines Verfahrens durch Bildung des Mittelwertes schätzen. Dadurch wird eine Normalverteilung der Schätzungen dieser Leistungsfähigkeit impliziert. Werden die Klassifikatoren über eine 10-fache Kreuzvalidierung auf genau den selben Splits des Datensatzes bewertet, erzeugt dies paarweise verbundene Stichproben zwischen den beiden Klassifikatoren, die sich mittels eines verbundenen Student'schen t-Tests auf signifikante Unterschiede in den mittleren AUC-Werten untersuchen lassen. Ziel ist das Ablehnen der Nullhypothese, dass beide Stichproben die gleiche Verteilung aufweisen.

Das Problem dieser Methodik liegt in der Verletzung einer wichtigen Annahme des verbundenen Student'schen t-Tests:
Die Beobachtungen innerhalb jeder Stichprobe sind nicht unabhängig, eine

zentrale Annahme vieler statistischer Testverfahren, so beispielsweise auch beim nichtparametrischen Wilcoxon-Rangsummentest [Wil92].
Im Rahmen des k-fachen Kreuzvalidierungsverfahrens geht eine Beobachtung in k-1 Trainingsdatensätzen ein. Daraus folgt unmittelbar, dass die geschätzten Evaluationsmetriken voneinander abhängig sind und dass die Berechnung der t-Statistik und damit auch der Test als solcher verzerrt ist.

Bereits 1998 befasste sich Dietterich [Die98] mit dem Problem des statistischen Vergleichs verschiedener Klassifikationsmodelle. Der Artikel beschreibt fünf approximative statistische Tests zur Bestimmung der Überlegenheit eines maschinellen Lernverfahrens gegenüber eines konkurrienden Lernalgorithmus. Diese Tests wurden experimentell verglichen, um die Wahrscheinlichkeit eines Typ-1-Fehlers zu schätzen. Der Typ-1-Fehler beschreibt das fälschliche Erkennen einer Differenz zwischen den Modellen. Die gerade erläuterte Methodik mittels Kreuzvalidierung weist eine erhöhte Wahrscheinlichkeit für einen Fehler 1.Art auf [Die98]. Zu den beiden Fehlerarten vergleiche Tabelle 5.2:

		Realität	
		H_0 ist wahr	H_1 ist wahr
Entscheidung des Tests ...	für H_0	✓	Fehler 2.Art
	für H_1	Fehler 1.Art	✓

Tabelle 5.2: Fehlerarten bei statistischen Testverfahren

Der McNemar's-Test [McN47] zeigt hingegen einen niedrigen Typ-I-Fehler. Der von Dietterich vorgeschlagene fünfte Test, 5×2CV, basierend auf fünf Iterationen einer zufälligen k=2 Kreuzvalidierung mit anschließendem verbundenen t-Test, weist ebenfalls einen akzeptablen Fehler 1.Art auf. Dietterich misst darüber hinaus auch die Power der statistischen Tests - die Fähigkeit, Unterschiede zwischen den verschiedenen Modellen zu erkennen, wenn sie tatsächlich vorhanden sind. Der kreuzvalidierte t-Test ist der Test mit der meisten Power, hat allerdings eine erhöhte Wahrscheinlichkeit für ein Fehler 1.Art, und sollte daher nur Verwendung finden, wenn die Power des Test die relevantere Größe darstellt. Der 5×2CV-Test erweist sich mit Blick auf die Power als etwas leistungsfähiger als der McNemar-Test [Die98].

Neben der Vermeidung der Fehler 1.Art bzw. 2.Art spielen die Berechnungskosten bei der Wahl des statistischen Testverfahrens ebenfalls eine Rolle. Für Algorithmen, die auf Grund einer komplexen und langwierigen Berechnung nur einmal evaluiert werden können, ist der Test von Mc-Nemar, der einzige Test mit akzeptablem Typ-I-Fehler. Für Algorithmen, die mindestens 10 mal in angemessener Zeit ausführbar sind, empfiehlt Dietterich den 5×2CV-Test. Diese Empfehlung fußt hauptsächlich auf der höheren Power des Verfahrens und der Tatsache, dass dieses die Variationen aufgrund der

Wahl des Trainingssets direkt misst [Die98]. Da die in dieser Arbeit verwendeten Modelle in annehmbarer Zeit trainier- und evaluierbar sind, wird das Feature Engineering mittels des 5x2CV-t-Tests bewertet.

Die Interpretation der Ergebnisse dieser Testprozedur sollte allerdings immer mit Sorgfalt erfolgen. Zum einen ist die 5x2CV in Bezug auf die Verletzung der Unabhängigkeitsannahmen des t-Tests nicht ideal. Die einzelnen Train-Test-Splits sind zwar nicht mehr voneinander abhängig, aber jede Kreuzvalidierung arbeitet immer noch auf der gleichen Datenbasis, so dass hier Abhängigkeitsverhältnisse entstehen.
Im Vorgehen dieser Arbeit kommt eine weitere Komponente hinzu, die die Unabhängigkeitsannahme weiter einschränkt:

Alle Transformationen des Merkmalsraums durch das Feature Engineering werden auf Basis eines 80-20-Train-Test-Splits bestimmt. Das heißt, jede k=2 Kreuzvalidierung auf diesem transformierten Datensatz greift in beiden Samples auf Informationen aus dem anderen Teil der Daten zurück, da alle Transformationen auf Basis von 80 Prozent der Beobachtungen berechnet wurden. Daraus resultiert eine schwache Abhängigkeit aller erzeugten Kreuzvalidierungssamples der 5x2CV.

Zusammenfassend muss festgestellt werden, dass die in dieser Arbeit berechneten Tests mit Vorsicht interpretiert werden sollten und nur eine Idee von der statistischen Relevanz der Methodik liefern.

Bevor sich das nächste Kapitel dem Feature Engineering mittels den Erklärungen eines Black-Box-Modells widmet, noch eine abschließende Bemerkung zur Evaluation des Feature Engineerings bzw. was dabei nicht außer Acht gelassen werden sollte.

Globale Erklärungen werden häufig für den gesamten Datensatz - also Test- und Trainingsdaten - erzeugt [HG18]. Soll allerdings anhand dieser Erklärungen ein Feature Engineering für andere Modelle betrieben werden, darf die Berechnung der Erklärungen nur auf Basis der Trainingsdaten oder einem eigens dafür zurückgehaltenen Datensatz erfolgen. Die Verwendung des Testdatensatzes ist zu vermeiden, da sonst die Gefahr besteht, Informationen aus den Testdaten - anhand derer später auch das White-Box-Modell evaluiert wird - in das Modelltraining zu *leaken*. Mehr zum Phänomen des sogenannten *Data Leakage* findet sich unter anderem in [KRP11].

6

FEATURE ENGINEERING MITTELS ERKLÄRUNGEN EINES BLACK-BOX-MODELLS

In vielen praktischen Problemen weisen Black-Box-Modelle eine gegenüber erklärbaren Modellen überlegene Leistungsfähigkeit auf. Ziel der Arbeit ist es, Prognosen und Erklärungen solcher performanten ML-Verfahren zu nutzen, um ein unterstützendes Feature Engineering für White-Box-Modelle durchzuführen. Soll die Performance erklärbarer Verfahren verbessert werden, ist es ein naheliegender Ansatz, die strukturellen Unterschiede zwischen Black- und White-Box-Modellen zu untersuchen und daraus Rückschlüsse auf die Leistungsfähigkeit der Verfahren zu ziehen.
Die Analyse dieser Unterschiede in Kombination mit der Generierung von Erklärungen ermöglicht, das Verhalten von Black-Box-Modellen zu verstehen. Die in diesem Kapitel vorgestellte Methodik verfolgt das Ziel, dieses Verhalten bestmöglich durch ein White-Box-Modell zu imitieren, um dessen Performance zu verbessern.
Ausgangspunkt des Vorgehens ist die Frage: *Wie kommt das Black-Box-Modell zu seiner Prognose, und lässt sich dessen Weg der Entscheidungsfindung imitieren?*

6.1 PERFORMANCEUNTERSCHIEDE: BLACK- VS. WHITE-BOX-MODELLE

Die in Tabelle 5.1 dargestellten Eigenschaften erklärbarer Modelle bilden die Anknüpfungspunkte für die Analyse möglicher Ursachen des Performanceunterschieds zwischen Black- und White-Box-Modellen.
Während Black-Box-Modelle (wie Neuronale Netze, Gradient Boosting oder Support Vector Machines) komplexe, nichtlineare, nichtmonotone Funktionen mit zahlreichen Interaktionen abbilden können, sind erklärbare Modelle immer an die in Tabelle 5.1 dargestellten Eigenschaften in Bezug auf Linearität und Monotonie der Modelle, sowie Interaktionen gebunden. Was im Kontext der Interpretierbarkeit ein großer Vorteil dieser Modelle ist, erweist sich mit Hinblick auf die Prognosequalität oft als Nachteil. Sind die Zusammenhänge zwischen einem Merkmal und der Zielgröße nichtlinear und/oder nichtmonoton, sind sie beispielsweise mit einer linearen oder logistischen Regression nicht adäquat abbildbar, was sich negativ auf die Performance des Modells auswirken kann.

Im Folgenden werden drei wesentliche Unterschiede zwischen Black- und White-Box-Modellen betrachtet, die alle Einfluss auf die Performance eines Modells haben und dadurch die Unterschiede, hinsichtlich der Prognosequalität, in Teilen erklären können: Die Auswahl der Merkmale, Nichtlineare Effekte und Interaktionen.

- **Auswahl der Merkmale:** Bereits in Kapitel 4.3 wurde darauf hingewiesen, dass die Inputmerkmale eines ML-Modells in vielen Fällen nicht gleichermaßen relevant sind. Die Auswahl der Merkmale für das Modelltraining ist - gerade für statistische Regressionsverfahren - aus zwei verschiedenen Gründen alles andere als trivial.

  Zum einen sind Regressionsverfahren sehr anfällig für verrauschte Daten bzw. Ausreißer, was die Performance negativ beeinflussen kann [HTF09]. Zum anderen ist es nicht immer leicht, die Trennkraft eines Merkmals zu beurteilen. Mögliche Anhaltspunkte liefert die Berechnung von Korrelationen zwischen Input- und Zielgrößen bzw. die Analyse der Verteilung der Zielgröße über die Merkmalsausprägungen hinweg. Dies ist aber aufwendig und fehleranfällig, da beispielsweise Wechselwirkungen zwischen Variablen nur eingeschränkt Berücksichtigung finden. Diese Einschränkung trifft auch auf die in Kapitel 5.3.1 erläuterten Selektionsverfahren zu.

  Viele Black-Box-Verfahren führen diese Auswahl prognoserelevanter Merkmale autonom, während des Trainings durch. Beispielsweise sind baumbasierte Ensemble oder Neuronale Netze auf Grund ihrer Struktur und des iterativen Trainings dazu imstande, ihre Prognosen auf besonders trennscharfe Merkmale zu begründen und irrelevante Information anderer Feature nur selten zu berücksichtigen, was die Performance und die Erklärbarkeit oft verbessert.

- **Nichtlineare Effekte:** Keines der erklärbaren Verfahren aus Tabelle 5.1 ist in der Lage komplexe nichtlineare Effekte adäquat zu repräsentieren. Der erklärbare Entscheidungsbaum (mit maximaler Tiefe Vier) erzeugt zwar nichtlineare Beziehungen zwischen Merkmal und Zielgröße, ist aber durch die geringe Tiefe nicht fähig, für mehrere Merkmale komplexe nichtlineare Effekte darzustellen.

  Sowohl die lineare als auch die logistische Regression modellieren Zusammenhänge zwischen Merkmalen und der Zielvariablen linear. Eine manuelle Integration nichtlinearer Effekte in erklärbare Modelle ist möglich, erfordert allerdings ausgeprägtes Domänenwissen oder eine aufwendige explorative Analyse des Datensatzes (vgl. hierzu auch Kapitel 5.3.1). Black-Box-Modelle sind hingegen nicht auf eine bestimmte funktionale Form des Zusammenhangs zwischen Input- und Outputgrößen angewiesen und erkennen dadurch auch nichtlineare Beziehungen.

- **Interaktionen:** Der Entscheidungsbaum ist das einzige der Verfahren aus Tabelle 5.1, welches imstande ist, eigenständig Interaktionen zwischen erklärenden Variablen zu erkennen. Sowohl in die lineare als auch die logistische Regression müssen Informationen über Wechselwirkungen zwischen Merkmalen manuell integriert werden. Erneut er-

fordert dies vom Entwickler viel Erfahrung in der entsprechenden Domäne, oder es läuft auf eine ausführliche explorative Analyse bzw. systematisches Ausprobieren hinaus (vgl. hierzu auch Kapitel 5.3.1). Hier sind Black-Box-Modelle aufgrund ihrer Struktur im Vorteil. Sie detektieren Interaktionen zwischen Merkmalen selbstständig und sind dabei nicht nur auf zweidimensionale Wechselwirkungen begrenzt, sondern können sogar Interaktionen beliebigen Grades erkennen. Dies ist in Hinblick auf die Performance der Black-Box-Modelle ein großer Vorteil, bedingt aber gleichzeitig deren vergleichsweise geringe Erklärbarkeit.

Es gibt also verschiedene Ursachen für große Performanceunterschiede zwischen erklärbaren und komplexen ML-Verfahren. Bisher wurde ausschließlich die Prognosequalität eines Modells betrachtet. Das Ziel ist allerdings die Verbesserung der Performance eines erklärbaren Modells, bei möglichst geringer Einschränkung der intrinsischen Erklärbarkeit. Unter diesem Gesichtspunkt, müssen die gerade erläuterten Ursachen, und damit auch die potenzielle Ansätze für ein verbessertes Feature Engineering durch Erklärungen von Black-Box-Modellen, nochmals neu bewertet werden:

Bereits bei den Annahmen in Kapiel 5.2 wurde auf die verbesserte Interpretierbarkeit von Modellen mit weniger Merkmalen hingewiesen. Die Nachvollziehbarkeit nichtlinearer Effekte hängt hingegen stark von der funktionalen Form des Zusammenhangs ab. Einfache, polynomiale Transformationen oder grobe Klassierungen eines Merkmals sind sicherlich leichter vermittelbar als Transformationen mittels komplexerer Funktionen. Ähnliches gilt für Interaktionen: So lange diese durch *einfache* funktionale Zusammenhänge repräsentierbar sind und die Performance des erklärbaren Modells deutlich verbessern, ist die Einschränkung der Erklärbarkeit sicherlich verschmerzbar. Ist dies allerdings nicht der Fall, bzw. wird die Anzahl an Interaktionen im Modell zu groß, schränken diese die Erklärbarkeit des Modells stark ein.

Sowohl bei der Integration von Nichtlinearitäten als auch Wechselwirkungen muss stets die menschliche Denkweise berücksichtigt werden. Diese neigt dazu, Zusammenhänge stets als monoton (und mit Vorliebe linear) anzunehmen. Darstellungen oder Modelle, die diesem Vorurteil nicht Rechnung tragen, erhalten weniger Beachtung/Aufmerksamkeit oder stoßen auf Widerspruch. Dieses Phänomen ist in der Psychologie als *Confirmation Bias* bekannt [Nic98]. Daher sollte es das Ziel sein, Transformationen so zu gestalten, dass sie entweder den Vorurteilen entsprechen oder möglichst leicht nachzuvollziehen sind.

6.2 VORGEHEN

Das Vorgehen zum Feature Engineering mittels den Erklärungen eines Black-Box-Modells lässt sich in verschiedene Teilschritte unterteilen, die in den folgenden Abschnitten detailliert beschrieben werden. Die Methodik orientiert

sich dabei stark an den in Kapitel 6.1 analysierten Gründen für die Leistungsunterschiede zwischen Black- und White-Box-Modellen. In Abbildung 6.1 findet sich eine Übersicht des gesamten Prozesses.

Zu Beginn stehen die Aufbereitung des Datensatzes mittels der in Kapitel 5.3 vorgestellten Methodik. Die Auswahl der White- bzw. Black-Box-Modelle erfolgt mittels einer k=5 Kreuzvalidierung und der 5x2CV-Methodik. Die Bestimmung der relevanten Merkmale und aller anderen Transformationen des Merkmalsraums erfolgt dann exklusiv auf einem 80-20 Train-Test-Split. In Unterkapitel 6.2.1 steht die Auswahl der Merkmale für das erklärbare Modell im Fokus, gefolgt von der Integration von Nichtlinearitäten (Kapitel 6.2.2) und Interaktionen (Abschnitt 6.2.3). Zu guter Letzt wird das Vorgehen anhand der in Abschnitt 5.3.4 dargestellten Testprozedur evaluiert.

Jede in Frage kommende Manipulation des Original-Datenraums wird durch die identische k=5 Kreuzvalidierung evaluiert. Verbessert sich die mittlere Performance des Modells durch die Transformation, wird diese beibehalten. Zur Einschränkung dieser Evaluation sei nochmals auf Kapitel 5.3.4 verwiesen.

6.2.1 *Auswahl der Merkmale*

Nachdem die Daten anhand des Vorgehens aus Kapitel 5.3.2 vorbereitet und die in Frage kommenden Black- und White-Box-Modelle trainiert und evaluiert wurden, folgt das Erzeugen verschiedener Erklärungen des komplexen maschinellen Lernverfahrens und daran anschließend das Ermitteln der Wichtigkeit eines jeden Features für das Modell.

Merkmale, denen das Modell keine große Bedeutung zumisst, werden aus dem Featurespace entfernt, wenn dadurch die Performance des erklärbaren Modells nicht deutlich absinkt. Aus Sicht der Performance des erklärbaren Modells kann die Merkmalsentfernung im ersten Moment kontraproduktiv erscheinen, da sich diese unter Umständen minimal verschlechtert. Gleichzeitig verringert sich allerdings die Komplexität des Modells, was zu einer Erhöhung der Erklärbarkeit führt.

In Kapitel 4.3 wurden verschiedene Maße zur Ermittlung der globalen Wichtigkeit von Merkmalen eingeführt. Darüber hinaus ermöglichen SHAP Werte die Berechnung einer Merkmalsrelevanz (Kapitel 4.6.4). Die Gradient Boosting Machines (GBM) werden in der Regel mittels des Python-Package *scikit-learn* implementiert, welches die globale Feature Importance über die Gini Wichtigkeit bestimmt. Diese berechnet sich, als über alle Bäume gemitteltes Produkt aus der Unreinheitsreduzierung eines Merkmalssplits und der Wahrscheinlichkeit, diesen Split zu erreichen (vgl. hierzu auch 4.3.1). Die SHAP Merkmalswichtigkeit wird als gemittelter Absolutbetrag der SHAP Values über alle Beobachtungen ermittelt .

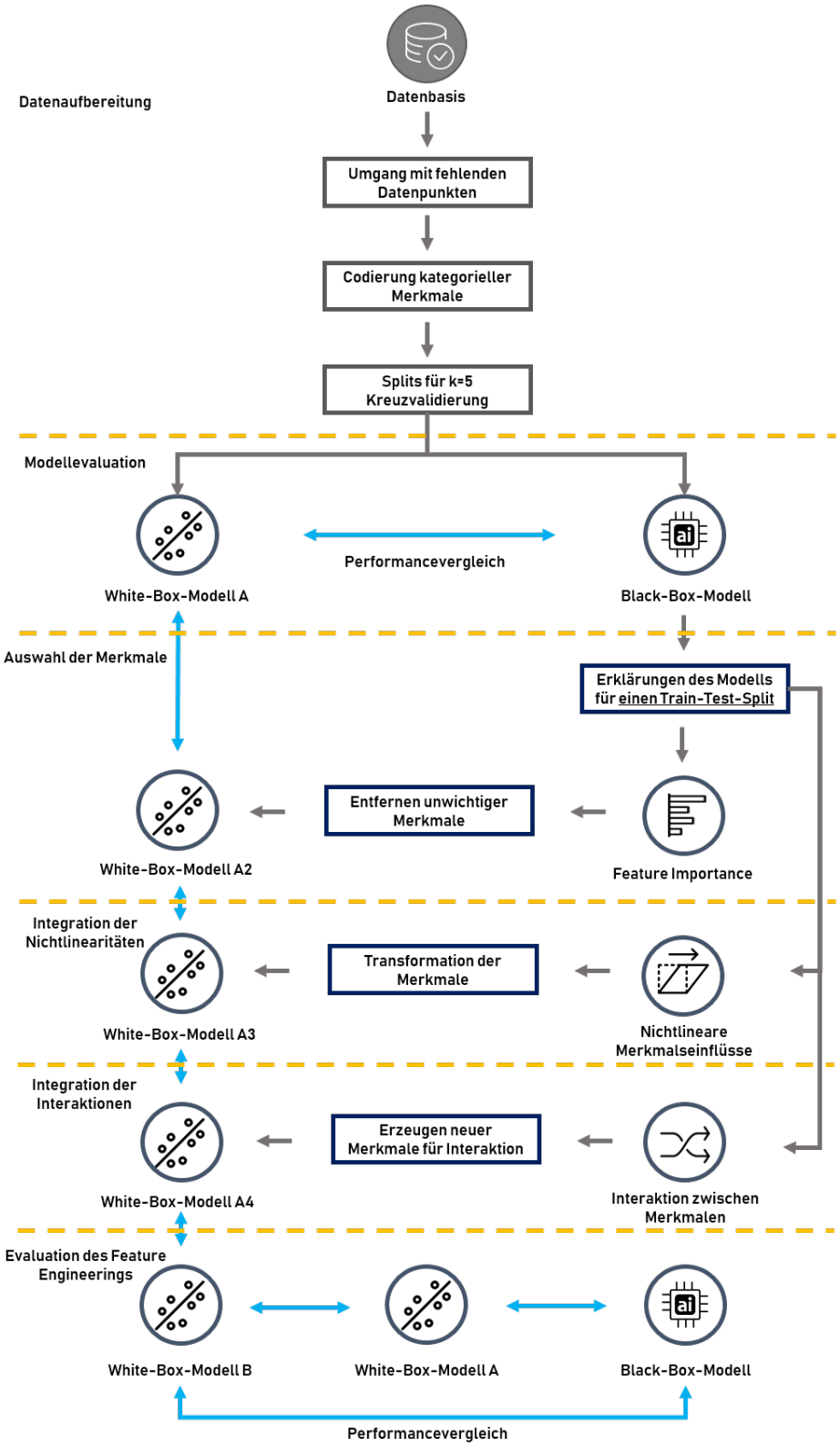


Abbildung 6.1: Graphische Darstellung des Vorgehens

6.2.2 *Integration von Nichtlinearitäten*

Nun ist die Verbesserung der Performance des Modells durch Merkmalstransformationen (ohne die Erklärbarkeit zu stark einzuschränken) das Ziel.

Dazu erfolgt die Transformation der verbleibenden Merkmale mittels SHAP-Erklärungen des Black-Box-Modells. Die Transformation ist abhängig vom Merkmalstyp und der Gestalt des *SHAP Dependence Plots*:

6.2.2.1 *Transformation kontinuierlicher Merkmale durch Polynome*

Die Idee zum Umgang mit kontinuierlichen Merkmalen wird anhand eines Beispiels besonders klar: Dazu sei der *SHAP Dependence Plot* einer Gradient Boosting Machine für das Merkmal (*Age*) gegeben. Da im nächsten Schritt (Kapitel 6.2.3) auch noch Interaktionen zwischen den Featuren in das erklärbare Modell integriert werden sollen, erfolgt hier zunächst ausschließlich eine Analyse der Haupteffekte des Merkmals. Zu *SHAP Dependence Plots* und der Aufteilung in Haupt- und Interaktionseffekt vergleiche Kapitel 4.6.4.

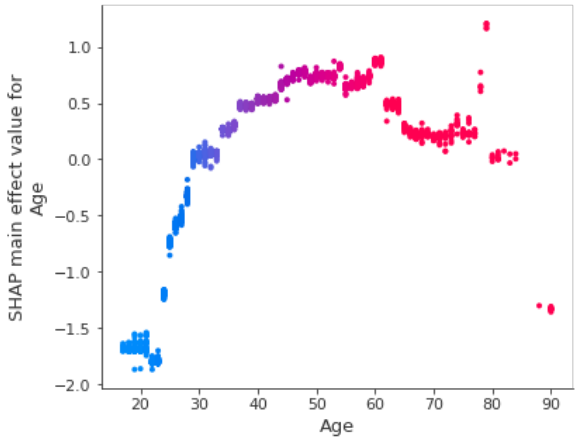


Abbildung 6.2: SHAP Dependence *Age* Haupteffekt (Adult-Datensatz)

Auf der x-Achse ist die Merkmalsausprägung - in diesem Fall das Alter - abgetragen. Auf der y-Achse befindet sich der SHAP Wert der Beobachtung. Jede Beobachtung wird in das Koordinatensystem eingetragen, wobei in diesem Plot die Farbcodierung keine inhaltliche Interpretation hat. Es ist klar zu erkennen, dass der Einfluss von Alter nichtlinear ist, sondern vielmehr ein quadratischer bzw. polynomialer Einfluss vorliegt. Es stellt sich die Frage, wie dieser Zusammenhang zu transformieren ist, um ihn für die logistische Regression abbildbar zu machen. Die Idee lässt sich wie folgt beschreiben:
Eine Approximation des Einflusses eines Merkmals durch eine *einfache* Funktion (wie z.B. ein Polynom vom Grad kleiner 6) und anschließendes Transformieren des Merkmal mittels dieser führt zu einer Transformation des Datenraums. Dadurch wird der nichtlineare Zusammenhang in eine approximativ lineare Beziehung überführt, welche auch von der logistischen Regression angenähert werden kann. In Abbildung 6.3 ist dieses Vorgehen exemplarisch für das Merkmal *Age* dargestellt.

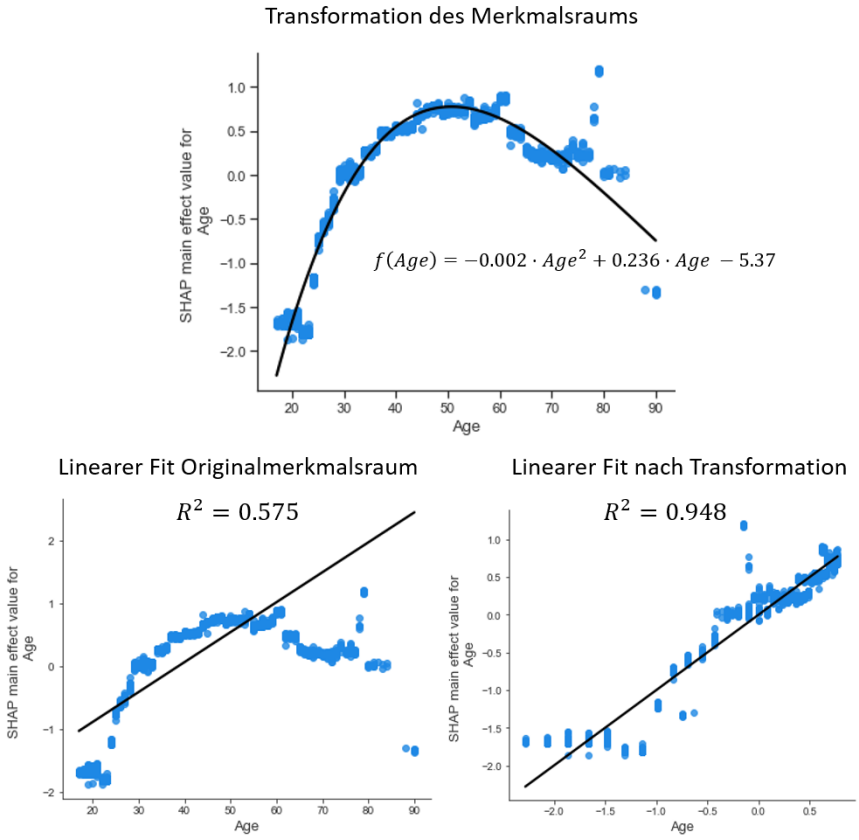


Abbildung 6.3: Transformation des Merkmals *Age* mittels eines Polynoms (Adult-Datensatz)

Für lineare und logistische Regressionsmodell ist die Beziehung zwischen den SHAP Werten und den Ausprägungen eines Merkmals linear. Das heißt, eine logistische Regression würde auf dem Originalmerkmalsraum (Abbildung 6.3 Unten Links) den durch die GBM detektierten Einfluss des Features *Age* nur unzureichend beschreiben, und erzielt lediglich ein R^2 von 0.575. Die Transformation des Merkmals *Age* mittels eines Polynoms zweiten Grades, welches den Einfluss des Merkmals auf die Vorhersage approximiert (Abbildung 6.3 Oben), ermöglicht eine deutlich bessere Anpassung der linearen Funktion an den Einfluss des transformierten Merkmals, mit einer R^2 von 0.948 (Abbildung 6.3 Unten Rechts).

Die Auswirkungen dieser Transformation auf die logistische Regression und den zugehörigen Einfluss des Merkmals *Age* sind in Abbildung 6.4 dargestellt:

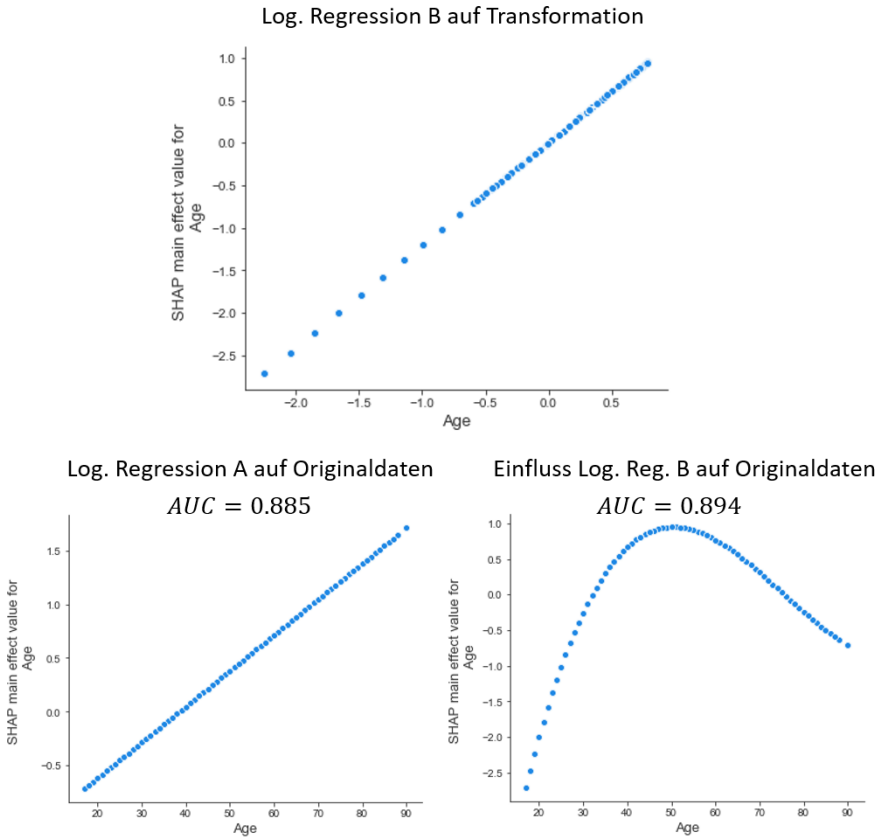


Abbildung 6.4: Auswirkungen der Transformation von *Age* auf die logistische Regression (Adult-Datensatz)

Unten Links ist der Einfluss des Features im Originaldatenraum zu erkennen. Wie zu erwarten ist der Einfluss linear. Eine logistische Regression A, die auf diesem unveränderten Merkmalsraum trainiert wurde, erzielt auf den Testdaten eine AUC von 0.885. Zum Vergleich: Die GBM aus Abbildung 6.3 erzielt auf dem selben Train-Test-Split eine AUC von 0.912 - eine deutlich bessere Performance.
Die Transformation des Merkmals *Age* anhand des Polynoms aus Abbildung 6.3 (Oben), und anschließendes Trainieren der logistischen Regression B auf dem sonst identischen Datenraum und gleichen Train-Test-Split ergibt eine AUC von 0.894. Das heißt, durch die Transformation hat sich die Performance deutlich verbessert. In Abbildung 6.4 (Oben) ist der Einfluss des transformierten Merkmals *Age* in der logistischen Regression B dargestellt. Dieser ist erneut linear, allerdings ist das Merkmal so schwerer zu interpretieren. Um die Erklärbarkeit, der so entstandenen logistischen Regression B zu gewährleisten, kann das Merkmal für das Erzeugen des *SHAP Dependence Plots* einfach wieder in den ursprünglichen Datenraum zurück transformiert werden (Abbildung 6.4 Unten Rechts). Nun ist besser zu erkennen, wie die Modellierung des Alters die Vorhersage des Modells beeinflusst.

Die Approximation durch ein Polynom stellt eine Vereinfachung des durch die performantere GBM beschriebenen Zusammenhangs zwischen Modellvorhersage und Merkmalsausprägung dar. Diese Vereinfachung beeinträchtigt mit Sicherheit die Performance der logistischen Regression im Vergleich zur GBM, allerdings hat sie in Bezug auf die Erklärbarkeit einen entscheidenden Vorteil. Während der Verlauf des Einflusses von *Age* bei der GBM abschnittsweise stark schwankt und auch Ausreißer auftreten (vgl. z.B. den Abschnitt zwischen 70 und 80 Jahren in 6.3 Unten Links), entsteht durch das Anpassen eines Polynoms eine stetige Funktion ohne Ausreißer oder Sprünge, was die Interpretation der Zusammenhänge - insbesondere für Laien - deutlich vereinfacht.

6.2.2.2 *Transformation kontinuierlicher Merkmale durch Klassierung*

Nicht immer ist der Einfluss eines Merkmales so eindeutig durch einen funktionalen Zusammenhang beschreibbar. Zum Beispiel sei der folgende *SHAP Dependence Plot* des Merkmals *Capital Loss* (Abbildung 6.5) gegeben.

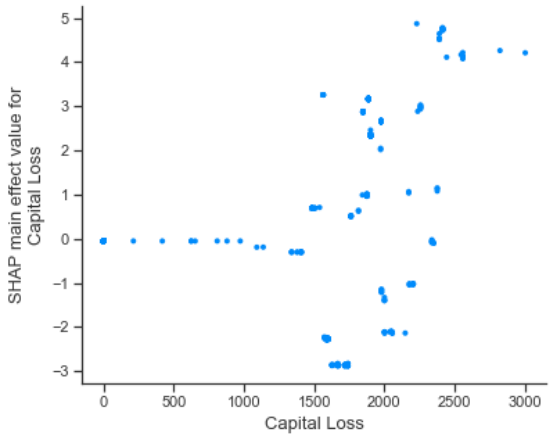


Abbildung 6.5: SHAP Dependence *Capital Loss* Haupteffekt (Adult-Datensatz)

Hier ist auf den ersten Blick kein funktionaler Zusammenhang erkennbar. Allerdings hat ein *Capital Loss* von kleiner als 1300 keinen Einfluss auf die Modellvorhersage, und für Werte größer 2400 ist der Einfluss sehr stark positiv. Der Bereich zwischen 1300 und 2400 lässt sich hingegen nur schwer charakterisieren. Daher ist in solchen Fällen das geschickte Kategorisieren des Features, mit dem Ziel möglichst viel Information über den Einfluss eines Merkmals zu erhalten, ein geeigneter Ansatz. In diesem Beispiel bietet es sich an, die bereits beschriebenen Grenzen als Split heranzuziehen. Abbildung 6.6 zeigt die Klassierung der Variable und die Konsequenzen auf den Einfluss im GBM-Modell.

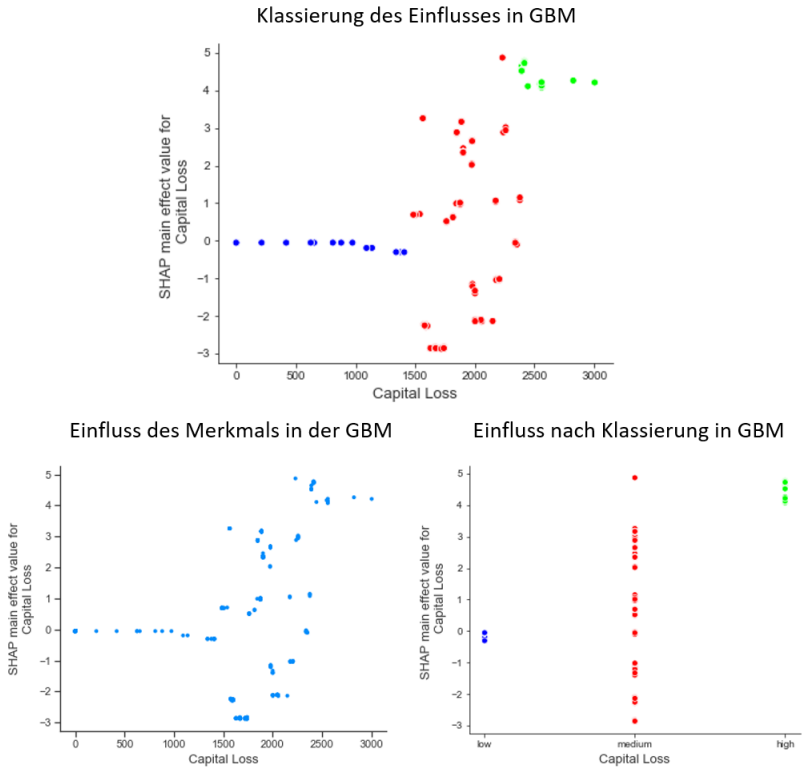


Abbildung 6.6: Transformation des Merkmals *Capital Loss* mittels Klassierung (Adult-Datensatz)

In Abbildung 6.6 Unten Rechts ist eindeutig zu erkennen, dass es sich bei dieser Klassierung um eine sehr starke Komprimierung des Einflusses der Variable *Capital Loss* handelt - insbesondere in dem als *medium* eingestuften roten Bereich.
Diese Klassierung kann in die logistische Regression als kategorielles Merkmal (nach einem entsprechend durchgeführten One-Hot-Encoding) einfließen. Zur Illustration des positiven Effekts dieser Kategorisierung findet sich in Abbildung 6.7 (Unten Links) zunächst der Einfluss von *Capital Loss* als stetiges Feature in der logistischen Regression. Es ist erneut die Linearitätsannahme der logistischen Regression zu erkennen. Darüber hinaus ist sichtbar, dass die logistische Regression A den Einfluss in Relation zur (trennschärferen) GBM im *low* Bereich eher überschätzt, während sie ihn im *high* Bereich eher unterschätzt.

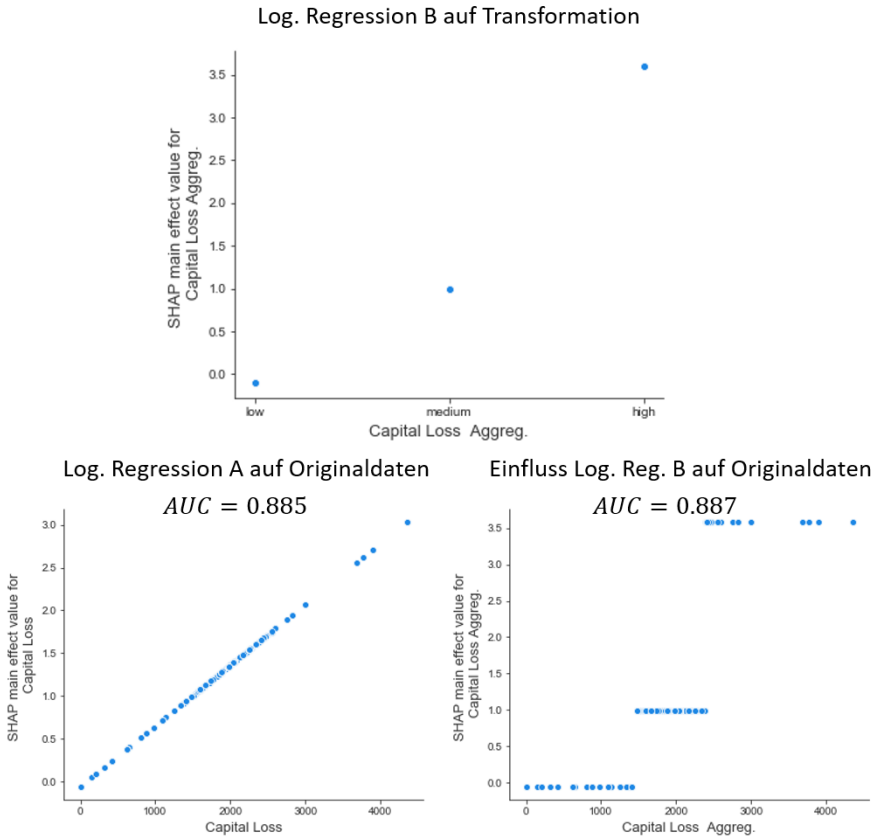


Abbildung 6.7: Auswirkungen der Klassierung von *Capital Loss* auf die logistische Regression (Adult-Datensatz)

Durchführen der Transformation anhand der Klassierung des Einflusses von *Capital Loss* im Black-Box-Modell führt für jedes Dummy-Merkmal zu einem eigenen Plot. Die Effekte der einzelnen Dummy-Variablen müssen entsprechend aggregiert werden, um den Gesamteffekt der Dummy-Klasse zu ermitteln (um den Effekt der Ausprägung *high* zu erhalten, müssen die SHAP-Values der Beobachtungen mit *medium*=0, *low*=0 und *high*=1 addiert werden). Eine entsprechende Darstellung, der so ermittelten Effekte, findet sich in Abbildung 6.7 (Oben). Wie bereits bei der Transformation durch Polynome geht auch bei der Klassierung die Beziehung zwischen Originalmerkmal und Einfluss auf die Modellvorhersage verloren. Durch eine Rücktransformation für die Erklärung lässt sich die Transparenz dieser Beziehung sicherstellen (vgl. Abbildung 6.7 Unten Rechts).

Auch hier führt die Transformation zu einer - in diesem Beispiel geringfügigen - Verbesserung der Trennschärfe des Modells von 0.885 auf 0.887. Dabei wurde der gleiche Train-Test-Split wie zuvor betrachtet und erneut der ursprüngliche Datenraum mit dem Datenraum nach der ausschließlichen Transformation des betrachteten Merkmals verglichen.
Dass der Gewinn an Performance hier eher gering ausfällt, ist angesichts

der starken Komprimierung der Information im mittleren Bereich der Merkmalsausprägung in der Kategorie *medium* nicht überraschend. Wie bereits bei der Transformation durch Polynome entsteht auch bei der Klassierung eine, im Vergleich zur GBM, deutlich leichter verständliche Beziehung zwischen Einfluss auf die Vorhersage und Ausprägung des Merkmals.

6.2.2.3 *Transformation kategorieller Merkmale*

Bereits in Kapitel 5.3.2 wurde darauf hingewiesen, dass der Umgang mit kategoriellen Merkmalen sehr stark vom zugrunde liegenden ML-Modell abhängig ist. Dennoch lassen sich aus dem Black-Box-Modell interessante Informationen bzgl. des Einflusses solcher Feature auf die Vorhersage gewinnen und diese sowohl im Kontext der Erklärbarkeit eines Modells als auch in Bezug auf die Performance nutzen.
Anders als bei kontinuierlichen Merkmalen sind für kategorielle Merkmale keine stetigen funktionalen Transformationen ermittelbar. Kategorielle Merkmale verfügen meist über keine Ordnung im Sinne von Rangfolge. Selbst wenn ordinale Variablen vorliegen, sind immer noch keine Aussagen über die Differenz zwischen den verschiedenen Kategorien und deren inhaltliche Bedeutung möglich. Daher bleibt nur die Möglichkeit, Kategorien, die einen ähnlichen Einfluss auf die Vorhersage des Modells haben, zusammenzufassen. Das Vorgehen ist sehr ähnlich zu dem gerade vorgestellten Klassieren kontinuierlicher Merkmale. Zur Illustration sei der *SHAP Dependence Plot* der Variable *TRX HOUR* (Abbildung 6.8) gegeben.

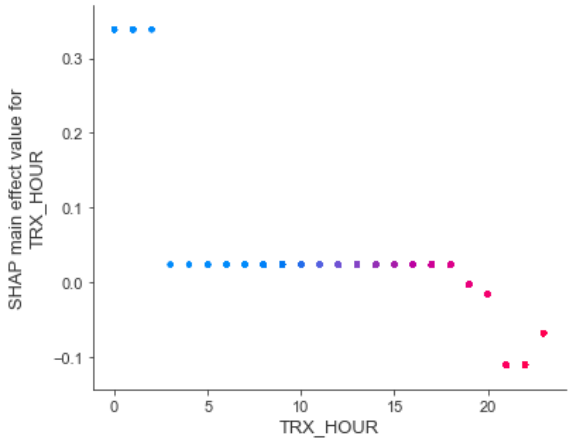


Abbildung 6.8: SHAP Dependence *TRX HOUR* Haupteffekt

Es ist zu erkennen, dass sich eine Uhrzeit zwischen 3 und 18 identisch, leicht positiv auf die Vorhersage auswirkt. Uhrzeiten zwischen Mitternacht und 2 Uhr haben einen starken positiven Einfluss. Alle anderen Zeiten des Tages wirken sich aus Sicht der GBM eher negativ auf die Prognose aus.
Eine naheliegende Transformation dieses Merkmals ist die Zusammenfassung der Zeiten zwischen 3 und 18 Uhr zu einer gemeinsamen Klasse. Gleiches gilt für die Zeiten zwischen 0 und 2 Uhr bzw. zwischen 21 und 22 Uhr.

Die sich daraus ergebende Klassierung aus Sicht der GBM befindet sich in Abbildung 6.9.

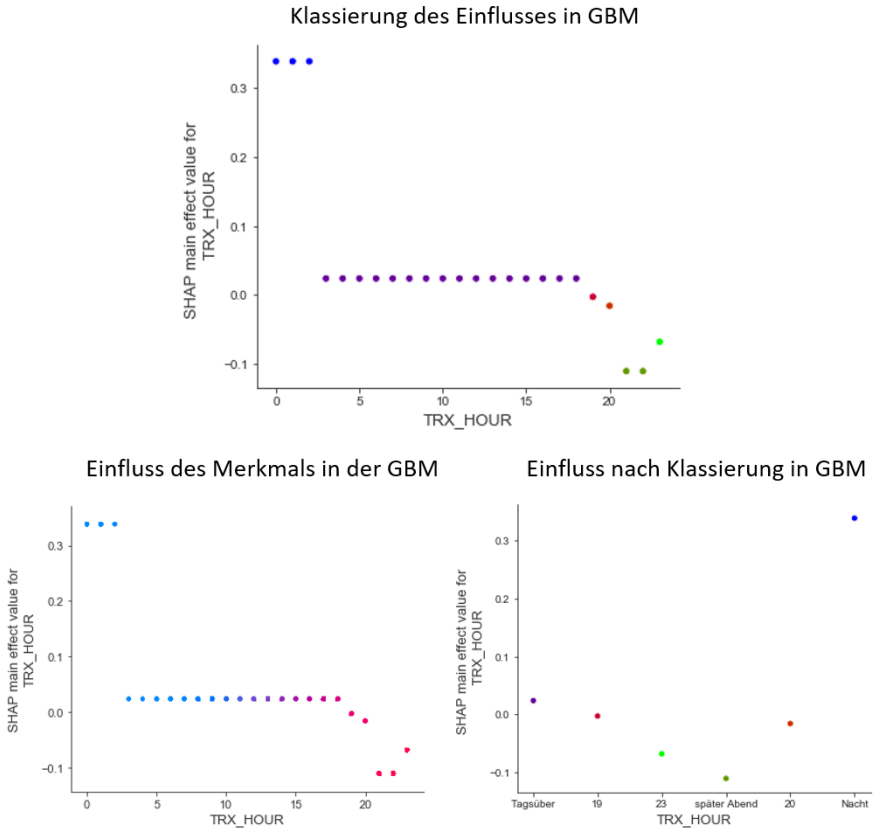


Abbildung 6.9: Transformation des Merkmals *TRX HOUR* mittels Klassierung

Anders als bei der Klassierung des stetigen Merkmals, wurde hier aus Sicht der GBM nur wenig Information komprimiert. Diese Klassierung läßt sich in der logistischen Regression als kategorielles Merkmal berücksichtigen, indem sie als Dummy-Variablen in das Modelltrainig integriert wird. Dieses Mal handelt es sich allerdings um eine kategorielle Variable, die bereits One-Hot-Encoded (mit 23 Dummy-Variablen) in die logistischen Regression A eingegangen ist (Abbilduung 6.10 Unten Links).
Erneut wurde der Effekt einer Kategorie durch Addieren der Einflüsse über alle Dummy-Variablen ermittelt. Daher ist in diesem Beispiel die implizite Annahme der Linearität der logistischen Regression nicht erkennbar. Der Einfluss streut sehr stark. Es ist kein tageszeitübergreifender Zusammenhang zwischen dem Merkmal und dessen Einfluss feststellbar. Die Erklärung der GBM ist in diesem Beispiel leichter nachzuvollziehen als die logistische Regression.

Die AUC dieser logistischen Regression A liegt bei 0.857. Die der GBM, welche den *SHAP Dependence Plot* in Abbildung 6.9 erzeugt hat, liegt bei 0.865 - also nur minimal über der der logistischen Regression A.

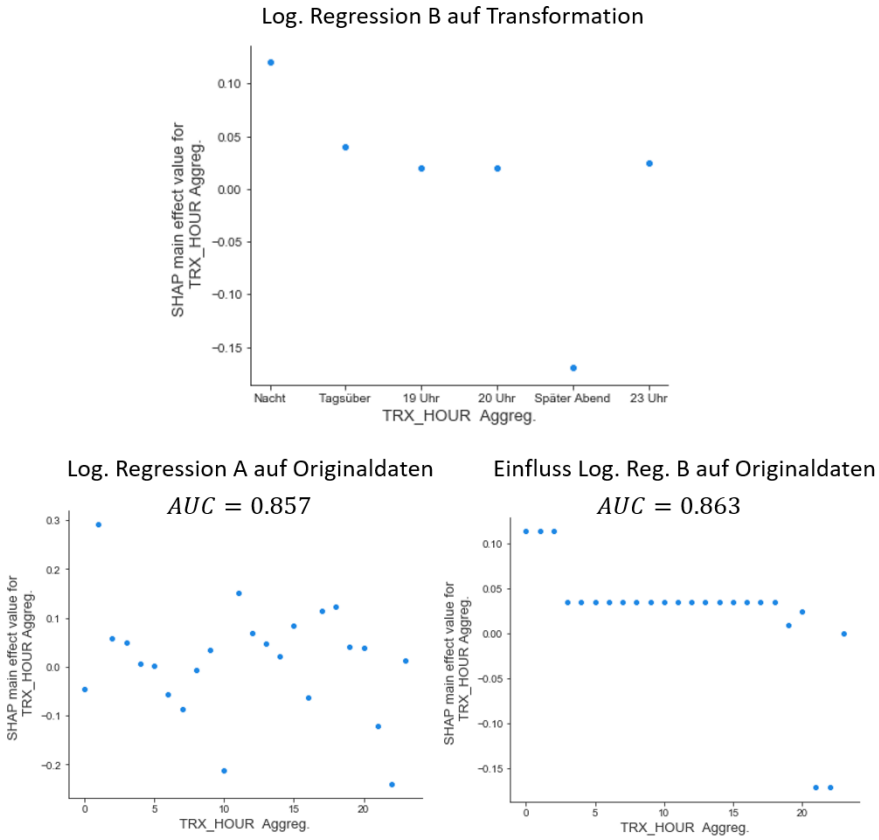


Abbildung 6.10: Auswirkungen der Transformation des Merkmals *TRX HOUR* auf die logistische Regression

Durchführen der Klassierung des Merkmals *TRX HOUR* anhand der Erklärung der GBM (Abbildung 6.10 Oben) fasst in diesem Fall mehrere - in der logistischen Regression bislang eigenständige - Klassen zusammen. In Abbildung 6.10 (Unten Links) wird deutlich, dass dadurch der logistischen Regression viel Information verloren geht. Dadurch verringert sich die Dimensionalität und damit auch die Komplexität der logistischen Regression. Es erhöht sich in jedem Falle die Erklärbarkeit des Modells, allerdings möglicherweise mit der Konsequenz einer verringerten Performance. Trainieren des Modells B auf dem entsprechend transformierten, ansonsten aber identischen Merkmalsraum führt zu einer AUC von 0.863 - also einer leichter Verbesserung der Performance. Wie bereits erwähnt, hätte der Effekt auch umgekehrt sein können. Ein möglicher Grund für die Verbesserung der Performance könnte eine bessere Generalisierung oder eine robustere Vorhersage des Modells sein.

Völlig unabhängig von der Performance wird durch das Zusammenfassen der Klassen die Erklärbarkeit des Modells erhöht. Ein Vergleich des Einflusses des Merkmals nach Rücktransformation in den originalen Datenraum (Abbildung 6.10 Unten Rechts) mit dem im Falle des unklassierten One-Hot-Encoding in Modell A (Abbildung 6.10 Unten Links) zeigt dies. Die logistische Regression B ist intuitiver und leichter nachvollziehbar.

6.2.3 *Integration von Interaktionen*

Nachdem bereits irrelevante Merkmale entfernt und nichtlineare Effekte integriert wurden, nun zur dritten Ursache für Performanceunterschiede zwischen Black- und White-Box-Modellen: zweidimensionale Interaktionen
Während in Kapitel 6.2.2 der Haupteffekt eines Merkmals isoliert betrachtet wurden, fokussiert sich dieser Abschnitt ausschließlich auf die Interaktionseffekte mit anderen Merkmalen. Erneut sei zu Details bezüglich der Aufteilung in Haupt- und Interaktionseffekte auf Kapitel 4.6.4 verwiesen.

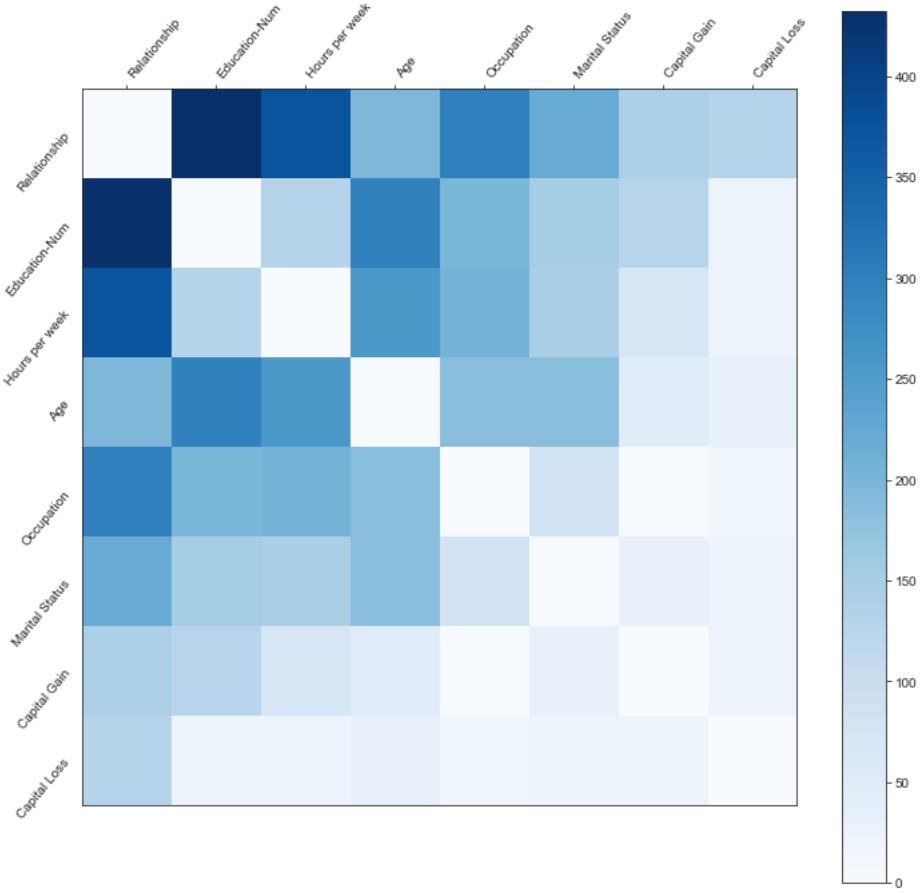


Abbildung 6.11: Heatmap des *SHAP Interaction Index*

Bevor zweidimensionalen Interaktionen in das erklärbare Modell integriert werden können, müssen zunächst die besonders starken Interaktionen von den schwachen Interaktionen differenziert werden. Zum einen gibt es bei n

Merkmalen $\frac{n \cdot (n-1)}{2}$ viele paarweise Interaktionen. Zum anderen beeinflusst die Größe des Interaktionseffekts die Performance des White-Box-Modells. Wird der Effekt zu klein, führt die Integration bestenfalls noch zu marginalen Performanceverbesserungen oder kann schlimmsten falls das Modell sogar verschlechtern. Beispielsweise ist die logistische Regression ein Verfahren, das ein gutes *Signal-zu-Rausch-Verhältnis* voraussetzt - eine Einschränkung vieler ML-Algorithmen. Ist das Signal einer Interaktion zu schwach und kaum von Rauschen zu unterscheiden, verschlechtert sich das *Signal-zu-Rausch-Verhältnis* und damit auch die logistische Regression.

Um die großen Interaktionseffekte zu erkennen, ist die in Abbildung 6.11 dargestellte Heatmap hilfreich. In dieser sind die Summen der *SHAP Interaction* Werte für jede Kombination von Merkmalen auftragen. Je größer diese Summe, desto dunkler die Fläche in der Heatmap und desto stärker die Interaktion zwischen den Merkmalen. In Hinblick auf die Modellperformance ist durch das Hinzufügen von starken Interaktionen eine größere Steigerung der Performance zu erwarten als bei eher schwachen Interaktionen.

Auf der Hauptdiagonalen betragen die Interaktionseffekte Null. Nach der Berechnungsformel für *SHAP Interaktionen* (vgl. Gleichung 4.18) wäre die Summe der Haupteffekte auf der Hauptdiagonalen der Heatmap konsistent. Dies hat allerdings einen entscheidenden Nachteil in Bezug auf das Detektieren der relevanten Interaktionen:
In der Regel sind Haupteffekte deutlich größer als Interaktionseffekte. Da in der Heatmap die Summe der jeweiligen Effekte über alle Beobachtungen betrachtet wird und diese Größe nach oben unbeschränkt ist, würden die Haupteffekte die Farbskala dominieren, so dass relevante Interaktionen schwerer von weniger relevanten unterscheidbar wären. Daher ist es anschaulich, nicht die Summe der Haupteffekte, sondern Null einzutragen.

In Abbildung 6.11 fallen zwei Interaktionen besonders stark auf - jene zwischen *Relationship* und *Education-Num* bzw. *Hours per week*. Erneut wird das Vorgehen anhand eines Beispiels dargestellt. Dazu sei die Interaktion zwischen *Relationship* und *Education-Num* in Abbildung 6.12 dargestellt.

Die Darstellung in diesem Plot ist vergleichbar zu der des *SHAP Dependence Plots*. Allerdings werden die Interaktionseffekte und nicht mehr die Haupteffekte aufgetragen. Darüber hinaus ergibt sich durch eine entsprechende Farbcodierung eine dritte Dimension der Darstellung, die in Abbildung 6.12 der Ausprägung des Merkmals *Relationship* entspricht.

In dieser Grafik sind mindestens zwei unterschiedliche Funktionsverläufe zu erkennen. Scheinbar verhält sich in diesem Modell die Kategorie *Verheiratete* (*Husband* und *Wife*) in Kombination mit dem Bildungsniveau anders, als die vier anderen Ausprägungen von *Relationship*. Um diese komplexe Interaktion zwischen Merkmalen auch der logistischen Regression zur Ver-

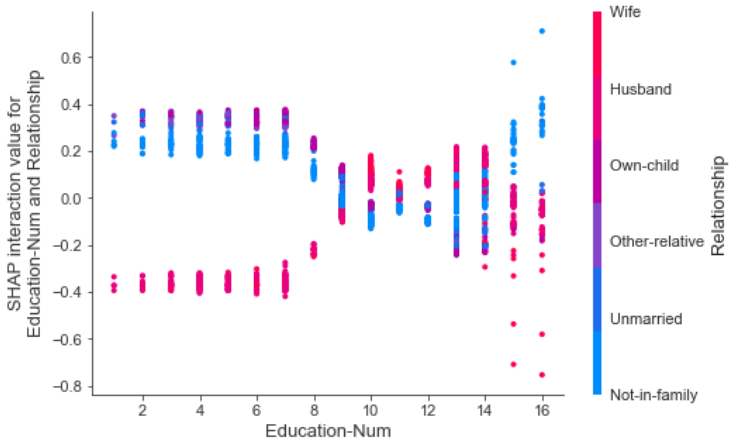


Abbildung 6.12: SHAP Interaction zwischen *Education-Num* und *Relationship*

fügung zu stellen, wird auf die Ansätze aus Kapitel 6.2.2 zurückgegriffen.

Statt einer Transformation für den kompletten Datenraum, erfolgt eine abschnittsweise Definition verschiedener Transformationsfunktionen im zweidimensionalen Datenraum der interagierenden Merkmale. In diesem Beispiel ist folglich

$$\text{Inter}(\text{Edu}, \text{Rel}) := \begin{cases} s(\text{Edu}) & \text{für Rel} \in \{\text{Wife}, \text{Husband}\} \\ t(\text{Edu}) & \text{für Rel} = \text{Not-in-family} \\ u(\text{Edu}) & \text{sonst.} \end{cases} \tag{6.1}$$

gesucht. Zur Schätzung von s, t und u wird erneut ein möglichst einfaches Polynom herangezogen. Wurde der Datenraum anhand der Bedingungen der abschnittsweisen Funktion eingeschränkt, entspricht das Vorgehen dem in Kapitel 6.2.2. Abbildung 6.13 zeigt das Vorgehen exemplarisch. Zunächst erfolgt ein Splitten des Interaktionseffekts anhand der Ausprägungen von *Relationship* in drei disjunkte Datenräume (vgl. Abbildung 6.13 Oben). Daran anschließend wird der nichtlineare Zusammenhang zwischen dem zweiten Merkmal *Education-Num* und dem Interaktionseffekt auf die Vorhersage, mittels eines einfachen Polynoms, approximiert. So entstehen Schätzungen für die Funktionen s, t und u.

Auch wenn das Vorgehen über das Definieren einer abschnittsweisen Funktion (Gleichung 6.1) motiviert wurde, ist es für die Integration dieser Information in das White-Box-Modelle wichtig, jede Teilfunktion der Interaktion s, t und u (in Kombination mit der entsprechenden Ausprägung des anderen interagierenden Merkmals) als eigenständiges Feature zur Verfügung zu stellen. Anderenfalls geht der Zusammenhang zwischen den beiden Merkmalen verloren.

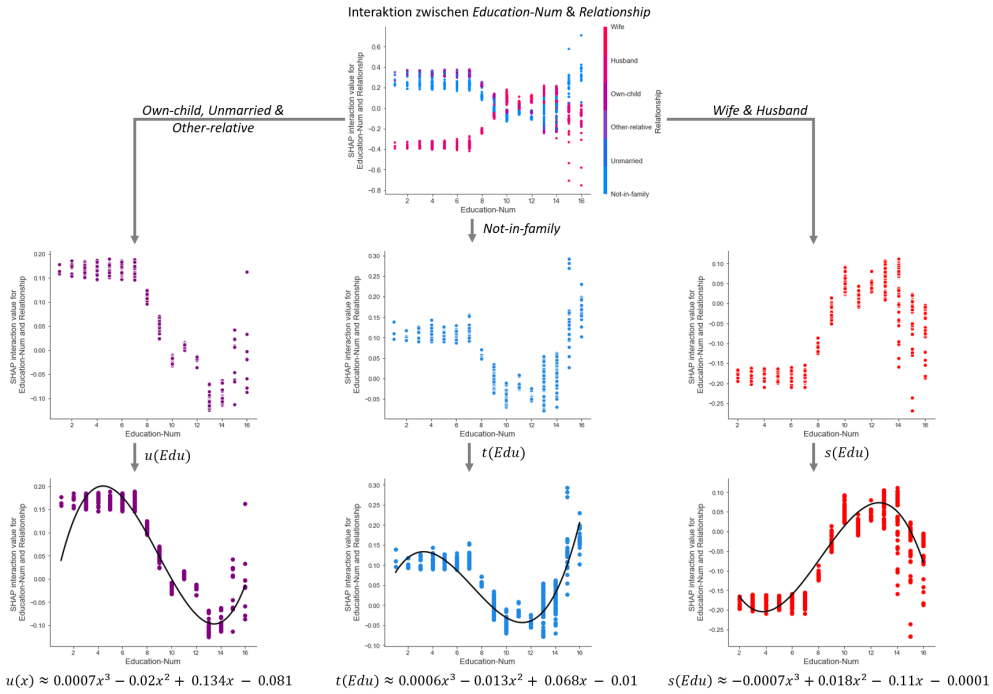


Abbildung 6.13: Approximation von Interaktionen durch Aufsplitten des Datenraums und Polynome

In diesem Beispiel müssen drei neue Merkmale generiert werden, die die Interaktion zwischen *Relationship* und *Education-Num* beschreiben:

$$\text{Inter}_1(\text{Edu}) = \begin{cases} s(\text{Edu}, \text{Rel}) & \text{für Rel} \in \{\text{Wife}, \text{Husband}\} \\ 0 & \text{sonst.} \end{cases}$$

$$\text{Inter}_2(\text{Edu}) = \begin{cases} t(\text{Edu}, \text{Rel}) & \text{für Rel} = \text{Not-in-family} \\ 0 & \text{sonst.} \end{cases}$$

$$\text{Inter}_3(\text{Edu}) = \begin{cases} u(\text{Edu}, \text{Rel}) & \text{für Rel} \notin \{\text{Wife, Husband, Not-in-family}\} \\ 0 & \text{sonst.} \end{cases}$$

Das Beispiel zur Interaktion zwischen *Relationship* und *Education-Num* bezieht sich auf die Interaktion zwischen einer kategoriellen und einer kontinuierlichen Variable. Allerdings beschränkt sich die Methodik nicht nur auf diese Kombination von Merkmalstypen. Auch Interaktionen zwischen zwei kategoriellen bzw. stetigen Merkmalen lassen sich mit diesem Vorgehen darstellen. Zuvor nochmals die drei Schritte zur Integration von zweidimensionalen Wechselwirkungen:

Ohne Beschränkung der Allgemeinheit sei Merkmal m_1 jenes Merkmal, anhand dessen die Partition des Datenraums durchgeführt wurde, und Merk-

mal m_2 jenes, welches zur Bestimmung der Transformation dient. Dann lässt sich das Vorgehen wie folgt beschreiben:

1. Zunächst muss eine möglichst einfache und geeignete Partition des durch m_1 und m_2 aufgespannten Datenraums gefunden werden. Dabei wird zunächst nur Feature m_1 als Splitkriterium betrachtet. Ziel ist es dabei, solche Partitionen zu finden, dass sich bezüglich der zweiten Dimension m_2 ein möglichst einfacher funktionaler Zusammenhang zwischen Merkmalsausprägung von m_2 und dem Interaktionseffekt ergibt.

2. Nach dem Ermitteln einer solchen Partition des zweidimensionalen Datenraums wird jede dieser Partitionen als eindimensionaler Merkmalsraum aufgefasst. Das heißt der Interaktionseffekt hängt jetzt nur von Merkmal m_2 ab. Dadurch wird es möglich, den funktionalen Zusammenhang mittels einer der Methoden aus Kapitel 6.2.2 zu beschreiben.

3. Um diese Informationen in White-Box-Modellen zu nutzen, erfolgt eine Kombination der Informationen aus beiden Schritten. Das heißt, es folgt die Erzeugung eines neuen Merkmals für jede Partition des Datenraums mittels m_1, welches die berechnete Transformation von m_2 für diese Partition durchführt.

In keinem der drei Verarbeitungsschritte werden Annahmen bzgl. der Merkmalstypen getroffen. Sowohl für kategorielle als auch für stetige Merkmale lassen sich Partitionen berechnen. Das Approximieren der funktionalen Zusammenhänge in Schritt 2 ist für beide Typen möglich. Vergleiche hierzu Kapitel 6.2.2.

7

FEATURE ENGINEERING ANHAND DER VORHERSAGEDIFFERENZ

Das gerade vorgestellte Vorgehen orientiert sich sehr stark an den limitierenden Eigenschaften vieler erklärbarer Modelle. In diesem Kapitel wird ein alternativer Ansatz vorgestellt, Merkmale anhand des Black-Box-Modells zu generieren. Anstatt das Black-Box-Modell in seiner Gesamtheit zu erklären und daraus Erkenntnisse über sinnvolle Transformationen bzw. neue Feature abzuleiten, betrachtet dieses Vorgehen ausschließlich die Beobachtungen, für die das Black-Box-Modell zu einer anderen Einschätzung wie das White-Box-Modell gelangt. Es interessiert also nicht mehr wie ein Black-Box-Modell zu seiner Prognose kommt, sondern stattdessen: *Wie unterscheiden sich die Beobachtungen, die nur mittels eines Black-Box-Modells erfasst werden können, vom Rest der Beobachtungen im Datensatz?*

7.1 ERKLÄRBARKEIT DES PERFORMANCEUNTERSCHIEDS

Die Teilmenge der Daten, die bereits von einem einfachen erklärbaren Modell korrekt klassifiziert werden, ist nicht besonders spannend; selbst wenn das komplexe Black-Box-Modell auf anderem Wege zu seiner Entscheidung gekommen ist. Von deutlich größerem Interesse sind die Bereiche der Daten, in denen es zu einer großen Differenz zwischen den Vorhersagen des komplexen und des einfachen Modells kommt. Denn in diesem Bereich scheinen die vereinfachenden Annahmen des White-Box-Modells nicht adäquat, um die Komplexität des Problems abzubilden, und das Black-Box-Modell liefert einen echten Mehrwert.

7.1.1 *Die Vorhersagedifferenz*

Ein erster naheliegender Ansatz, um Bereiche in den Daten zu identifizieren, in denen das Black-Box-Modell seine Stärken ausspielen kann, ist die exklusive Betrachtung von, die nur durch das Black-Box-Modell richtig klassifiziert werden. Dies führt allerdings erneut zum Problem der Cut-Off-Wahrscheinlichkeiten aus Kapitel 5.3.3. Viele Klassifikationsalgorithmen liefern eine Wahrscheinlichkeitsverteilung über die möglichen Zielklassen als Ergebnis, so dass eine Grenze zu wählen ist, ab der eine Beobachtung hart einer Zielklasse zugeordnet wird. Die Wahl eines solchen Schwellwerts ist zum einen stark problemabhängig und geht zusätzlich mit einem enormen Informationsverlust einher, wie das folgende Beispiel illustriert:

Im Falle eines binären Klassifikationsproblems mit einen Schwellwert von 0.5 erhält eine Beobachtung mit einer Wahrscheinlichkeitsverteilung von 0.51 zu 0.49 über die Zielklassen die gleiche Ausprägung der Zielvariable, wie eine Beobachtung mit der Verteilung 0.91 zu 0.09. Dies ist ein enormer Verlust an relevanter Information, geben diese Wahrscheinlichkeiten doch so etwas wie die *Sicherheit* des Klassifikators bei seiner Prognose wieder.

Beim Vergleich der Ergebnisse zweier oder mehrerer Klassifikationsverfahren wird dieser Effekt sogar noch stärker, da dieser Verlust bzgl. der Sicherheit einer Prognose nicht nur zwischen verschiedenen Beobachtungen auftritt, sondern sogar für ein und dieselbe Beobachtung zwischen mehreren Klassifikatoren. Im Falle des Schwellwertes von 0.5 und einer geschätzten Verteilung durch Klassifikator A von 0.51 zu 0.49 über die Zieklassen und Klassifikator B (für die gleiche Beobachtung) zu 0.49 zu 0.51, führt diese harte Klassifikation zu der Schlussfolgerung, dass A eine andere Entscheidung getroffen hat, als B, obwohl beide Verfahren zu einer ähnlichen Einschätzung bezüglich der Verteilung der Zielklassen für diese Beobachtungen gekommen sind.

Aus diesem Grund wird im weiteren Verlauf ein anderer Ansatz verfolgt. Für jede Beobachtung erfolgt eine Bestimmung der Differenz der Vorhersagen beider Klassifikatoren. Im Falle binärer Klassifikationsprobleme, ergibt sich diese einfach durch Fixieren einer Zielklasse und Berechnung der Differenz für die Wahrscheinlichkeiten dieser Klasse. Dadurch bleibt die Sicherheit, mit der ein Verfahren zu einer Klassifikation kommt, erhalten. Im obigen Beispiel würde sich eine Klassifikationsdifferenz zwischen A und B für die erste Zielklasse von 0.51-0.49 = 0.02 ergeben, was die Situation deutlich besser widerspiegelt.

Ist die Differenz der Vorhersagen zwischen White-Box- und Black-Box-Modell gering, so schätzen beide Verfahren die analysierte Beobachtung ähnlich ein, und die vereinfachenden Annahmen des erklärbaren Modells beschreiben die Situation bereits ausreichend. Ist die Differenz groß, deutet es auf eine falsche Einschätzung durch eines der Modelle hin. Wann eine Differenz groß bzw. klein ist, hängt vom betrachteten Problem und den verwendeten ML-Verfahren ab. Es stellt sich nun das gleiche Problem wie beim ersten naiven Ansatz: Die Bestimmung eines geeigneten Schwellwerts. Allerdings ist in diesem Setting eine systematische Bestimmung dieses Wertes möglich.

7.1.2 *Bestimmung des Schwellwertes*

Die Performance eines erklärbaren Modells W wird mit der eines Black-Box-Modells B verglichen. Dabei erfolgt dieser Vergleich nur, wenn die Performance des Black-Box-Modells der von W überlegen war (vergleiche hierzu die Annahmen aus Kapitel 5.2). Bei der Berechnung der Vorhersagedifferenz

zwischen B und W, ist - aufgrund der überlegenen Trennschärfe des Black-Box-Ansatzes - davon auszugehen, dass diese Differenz vor allem dann betragsmäßig große Werte annimmt, wenn Beobachtungen analysiert werden, für die das Black-Box-Modell zu einer besseren Einschätzung kommt als das erklärbare Modell W. Mit dieser Annahme im Hinterkopf, erfolgt die experimentelle Bestimmung des Schwellwerts für die Vorhersagedifferenzen:

Dazu wird der Schwellwert systematisch von sehr niedrigen Werten - z.B. 0.05 - ausgehend, schrittweise erhöht. Anschließend erfolgt eine Teilung des Datensatzes anhand dieses Cut-Offs in Beobachtungen mit kleiner Vorhersagedifferenz und solche mit einer großen Differenz. Für jeden dieser beiden Teildatensätze wird die Performance des Black-Box-Modells B und des White-Box-Ansatzes W bestimmt. Die Erhöhung des Schwellwertes erfolgt so lange, bis die Performance des erklärbaren Verfahrens W auf Beobachtungen mit großer Differenz unter die Trennschärfe eines generischen oder zufälligen Modells fällt. Im Falle eines mittels ROC-AUC evaluierten Klassifikationsproblems, wird also solange erhöht, bis das Modell W auf den Beobachtungen mit großer Prognosedifferenz lediglich eine AUC von ungefähr 0.5 erzielt. Wenn die oben getroffene Annahme der überlegenen Performance des Black-Box-Modells zutrifft, sollte sich dessen Performance auf den Dateninstanzen mit großer Differenz deutlich weniger stark reduzieren, wenn der Schwellwert erhöht wird.

7.1.3 *Analyse der Vorhersagedifferenz*

Wurde mittels der gerade vorgestellten Methode ein Cut-Off-Wert und damit eine Aufteilung des Datensatzes in Beobachtungen mit kleiner bzw. großer Differenz bestimmt, bietet SHAP die Möglichkeit, Erklärungen beider Modelle für beide Teildatensätze zu erzeugen (ohne das Modell neu trainieren zu müssen). Dies ermöglicht Vergleiche der beiden Modelle und Datensätze, woraus sich Erkenntnisse über mögliche Gründe für die überlegene Performance des Black-Box-Modells gewinnen lassen.

Dabei sind zwei Vergleiche von besonderem Interesse: Zum einen der Vergleich über die beiden Teildatensätze hinweg. Dabei spielen die drei folgenden Fragen eine große Rolle:

- Wie verändert sich die Verteilung der einzelnen Merkmale? Haben vielleicht nur Teilmengen der Merkmalsräume einen Einfluss bei den Instanzen mit großer bzw. kleiner Vorhersagedifferenz?
- Wie verändert sich die globale Relevanz einzelner Merkmale, wenn Beobachtungen mit kleiner und solche mit großer Vorhersagedifferenz betrachtet werden?
- Wie verändert sich der Merkmalseinfluss über den Merkmalsraum?

Neben diesem Vergleich über die beiden Datenräume hinweg spielt auch der Vergleich zwischen White- und Black-Box-Modell innerhalb desselben Merkmalsraums eine große Rolle. Dabei stehen ähnliche Fragen im Zentrum der Analyse:

- Wie unterscheidet sich die Relevanz der Merkmale zwischen den Modellen innerhalb eines Teildatensatzes?
- Lassen sich (insbesondere auf den Dateninstanzen mit großer Differenz) Merkmalseinflüsse erkennen, die die große Differenz zwischen Black- und White-Box-Modell erklären können?

Eine systematische Darstellung des Vorgehens zur Analyse von Prognosedifferenzen und deren Auswirkungen auf Verteilung und Relevanz von Merkmalen findet sich in Abbildung 7.1.

7.2 MERKMALSGENERIERUNG MITTELS DER VORHERSAGEDIFFERENZ

Mittels der in Abschnitt 7.1 vorgestellten Methodik können Gründe für die überlegene Performance von Black-Box-Modellen auf den Daten mit großer Vorhersagedifferenz identifiziert werden. Darauf aufbauend folgt eine Analyse, inwieweit die gewonnenen Erkenntnisse über abweichende Merkmalsverteilungen und -einflüsse nutzbar gemacht werden können, um das White-Box-Modell zu verbessern. Zunächst wird dabei das Ziel verfolgt, die Performance des erklärbaren Verfahrens auf Beobachtungen mit großer Vorhersagedifferenz zu verbessern. Auf Grund der Struktur erklärbarer Modelle ist es notwendig, stets die Gesamtperformance auf dem kompletten Datensatz im Auge zu behalten, da sonst die Gefahr besteht, das Modell auf diese speziellen Beobachtungen zu overfitten.

Es scheinen zwei Ansätze geeignet, um auf Basis der Analyse von Vorhersagedifferenzen neue Merkmale für erklärbare Modelle zu generieren:

1. Bei der Analyse dieser Daten werden neue, bislang nicht detektierte nichtlineare Effekte oder Wechselwirkungen erkannt, die mittels einer der in Kapitel 6.2.2 bzw. 6.2.3 vorgestellten Methoden in das Modell integriert werden können.
2. Darüber hinaus besteht die Möglichkeit, auf den Daten mit großer Vorhersagedifferenz zwischen White- und Black-Box-Modell ein neues erklärbares Modell zu trainieren.

7.2.1 *Feature Engineering mittels Erklärungen der großen Vorhersagedifferenz*

Der erste Ansatz unterscheidet sich in seinem Vorgehen nicht besonders stark von dem in Kapitel 6 vorgestellten. Der einzige Unterschied ist der Datenraum, auf dem die Methodik Anwendung findet. Dieser ist nun deutlich

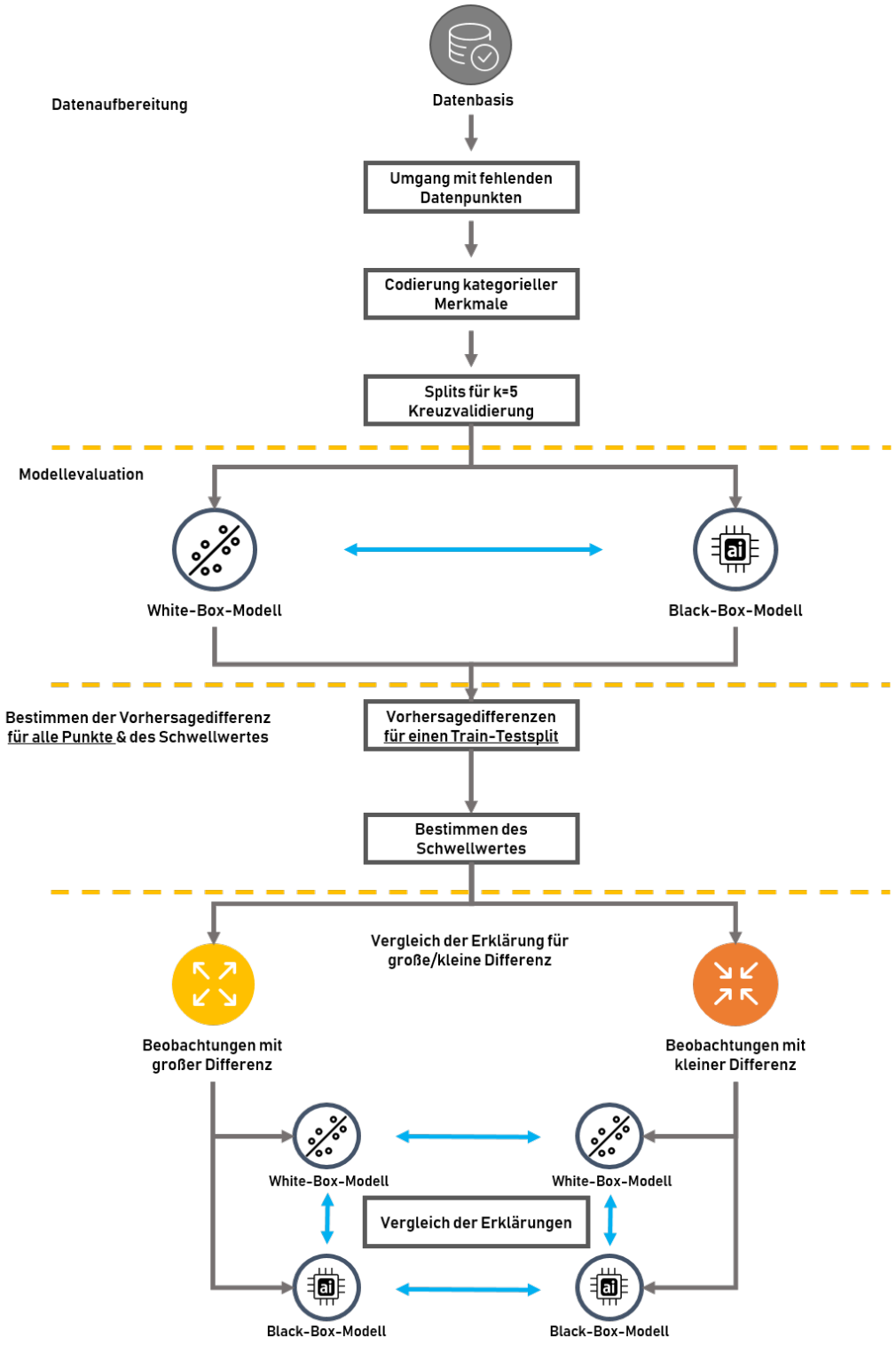


Abbildung 7.1: Vorgehen zur Analyse der Prognosedifferenz

kleiner und beinhaltet lediglich die Beobachtungen, bei denen die Vorhersage des erklärbaren Modells stark von der des Black-Box-Modells abweicht.

Die Tatsache, dass nun ein deutlich kleinerer Datenraum betrachtet wird, muss bei der Integration der neuen Merkmale beachtet werden. Es besteht die Möglichkeit, dass die gefundenen funktionalen Approximationen von Nichtlinearitäten bzw. Wechselwirkungen nicht mehr auf den vollständigen Merkmalsraum verallgemeinerbar sind und so zu einer schlechteren Gesamtperformance des Modells führen.

7.2.2 *Erklärbare Modelle auf großer Vorhersagedifferenz*

Alternativ ist es möglich, ein völlig neues, erklärbares Modell C auf den Dateninstanzen mit großer Vorhersagedifferenz zu trainieren. Die Idee hinter diesem Ansatz lässt sich wie folgt beschreiben:
Der Schwellwert bzgl. der Vorhersagedifferenz wurde in Kapitel 7.1.2 so bestimmt, dass die Performance des erklärbaren Modells auf Beobachtungen mit großer Differenz nur noch mit der eines generischen, zufälligen Modells vergleichbar ist. Daraus folgt, dass die in dieser Teilmenge des Datenraums befindlichen Datenpunkte mit den Mitteln des White-Box-Modells nicht besser als zufällig, einer der beiden Klassen zugeordnet werden können. Daraus lässt sich schlussfolgern, dass sich diese Beobachtungen fundamental von denen mit geringer Differenz unterscheiden müssen, da sie sonst auch durch das erklärbare Modell hätten beschrieben werden können.

Es gibt zwei mögliche Gründe warum das erklärbare Modell nicht imstande war, diese Dateninstanzen in gleicher Art und Weise wie das zum Black-Box-Modell zu beschreiben: Entweder ist der Zusammenhang zwischen der Zielgröße und den Merkmalen für diese Beobachtungen nicht durch ein erklärbares Modell darstellbar da seine Einschränkungen dies verhindern (vergleiche dazu auch Kapitel 6.1). In diesem Falle kann auch das Trainieren eines weiteren White-Box-Modells C keine neuen Erkenntnisse zu Tage fördern.

Darüber hinaus besteht aber auch die Möglichkeit, dass die Beziehung zwischen dem Merkmalsraum und der zu erklärenden Variable nur durch ein - im Vergleich zum White-Box-Modell W, mittels dessen die Differenz bestimmt wurde - völlig andersartiges Modell C beschreibbar sind. In diesem Fall können aus Modell C neue Informationen für den ursprünglichen erklärbaren Ansatz W gewonnen werden und lassen sich in diesen integrieren.

Wie sich Informationen des neuen Modells C in das bereits bestehende White-Box-Modell W integrieren lassen, ist zunächst nicht klar. Es sind mehrere Ansätze denkbar:

- Es lassen sich neue Merkmale aus Modell C extrahierern, die in das ursprüngliche, erklärbare Verfahren W integriert werden. Handelt es sich bei C um einen Entscheidungsbaum, wäre eine Extraktion der

Pfade zu besonders reinen, trennscharfen Knoten als neues Merkmal denkbar. Dieses Vorgehen wird in Kapitel 8 beispielhaft illustriert und ist in Abbildung 8.24 systematisch dargestellt.

- Darüber hinaus wäre eine Kombination der Vorhersagen dieser erklärbaren Modelle C mit den Vorhersagen des ursprünglichen White-Box-Ansatzes W über ein *(stacked) Ensemble* denkbar. Bei *Stacking*, manchmal auch Super Learning [LPH07] oder Stacked Regression [Bre96] genannt, handelt es sich um eine Klasse von Algorithmen, die auf zweiter Stufe einen *Meta-Lerner* trainieren, um eine optimale Kombination der Basislerner zu finden. Häufig wird eine logistische Regression als Meta-Lerner verwendet. Obwohl das Konzept bereits in den 90ern entwickelt wurde, erfolgte eine mathematische Begründung der Theorie erst im Jahr 2007 in [LPH07]. In diesem Beitrag wurde gezeigt, dass das Super-Lerner-Ensemble ein asymptotisch optimales System zum Lernen darstellt.

  Unter dem Gesichtspunkt der Erklärbarkeit ist dieser Ansatz allerdings deutlich schwerer realisierbar als die anderen bisher vorgestellten. Die Kombination der Vorhersagen mehrerer Modelle - selbst wenn diese erklärbar sind - über ein weiteres Modell, ist einem Laien wohl deutlich schwerer verständlich zu machen, als die bisher durchgeführten Schritte zur Performanceverbesserung. Aus diesem Grund wird der Ansatz in dieser Arbeit nicht weiter verfolgt.

Bevor im nächsten Teil der Arbeit eine empirische Untersuchung des Vorgehens anhand des Adult-Datensatzes erfolgt, noch eine Anmerkung zur Übertragung dieser Methodik auf Regressionsprobleme:

Grundsätzlich spricht nichts dagegen, auch für Regressionsprobleme Vorhersagedifferenzen zu berechnen. Auf Grund der Tatsache, dass die Zielgröße dann nicht mehr beschränkt und keine Evaluation der Performance in Bezug zu einer Baseline möglich ist, muss die Wahl des Schwellwerts, mittels dessen Beobachtungen in solche mit kleine bzw. große Differenz aufgeteilt werden, heuristisch erfolgen. Ist ein problemspezifischer Schwellwert bestimmt, übertragen sich alle weiteren Konzepte aus diesem Kapitel auch auf Regressionsprobleme.

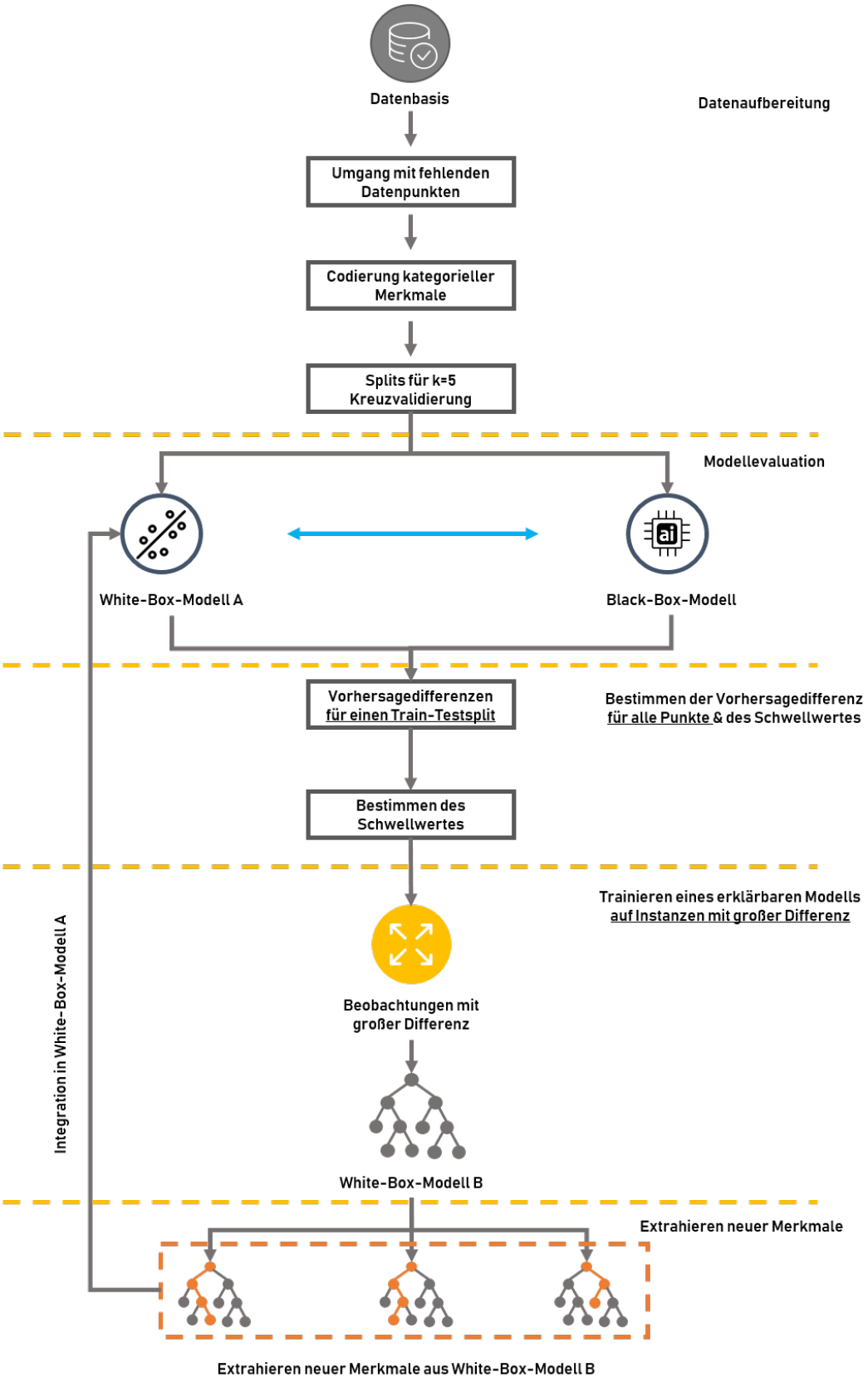


Abbildung 7.2: Vorgehen zur Analyse

Teil III

EMPIRISCHE UNTERSUCHUNG ANHAND DES ADULT-DATENSATZES

8

FEATURE ENGINEERING FÜR DEN ADULT-DATENSATZ

Der Adult-Datensatz [DG17] enthält Zensusdaten der USA aus dem Jahr 1994 zu ca. 32500 Personen. Zu jeder Beobachtung wurden 12 verschiedene Merkmale erhoben. Diese wurden 1994 von Barry Becker aus der Zensusdatenbank extrahiert [Koh97] und mit dem Ziel versehen, anhand dieser Merkmale vorherzusagen, ob das Einkommen der Person $>$ 50000$ oder $\leq$ 50000$ ist. Damit haben Ron Kohav und Barry Becker ein binäres Klassifikationsproblem definiert.

8.1 BESCHREIBUNG DES DATENSATZES

Der Datensatz enthält fünf stetige und sieben kategorielle Variablen (vgl. Tabelle 8.1), die im Folgenden genauer untersucht werden. Der Code zur Erzeugung der Modelle und Visualisierungen findet sich bei GitHub.

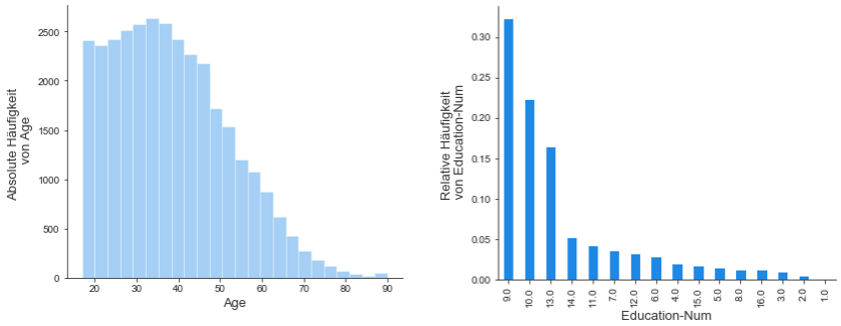


Abbildung 8.1: Adult-Datensatz: Verteilung *Age* und *Education-Num*

Age ist eine kontinuierliche Variable, die Werte zwischen 17 und 90 annimmt. Der Mittelwert beträgt 38.5 mit einer Standardabweichung von 13.6 Jahren. Anhand des Histogramms (vgl. Abbildung 8.1 Links) ist zu erkennen, dass die Verteilung leicht rechtsschief ist.

Education-Num ist eine kontinuierliche Variable, die Werte zwischen 1 und 16 annimmt. Der Mittelwert und Median beträgt 10 Schuljahre mit einer Standardabweichung von 2.5 Jahren. Das Histogramm (vgl. Abbildung 8.1 Rechts) zeigt, dass über 50% der Personen eine neun- oder zehnjährige Ausbildung (Schule und Universität) genossen haben.

Merkmal	Beschreibung	Ausprägungen
stetig		
Age	Alter in Jahren	$[17, 90]$
Education-Num	Jahre in Ausbildung	$[0, 16]$
Capital Gain	Kapitalgewinne	$[0, 100000]$
Capital Loss	Kapitalverlust	$[0, 4356]$
Hours per week	Arbeitszeit pro Woche	$[0, 99]$
kategoriell		
Workclass	Beschäftigungsverhältnis	Private, Self-not-inc, Self-inc, Fed.-gov, Loc.-gov, Sta.-gov, Without-pay, Never-worked
Marital Status	Familienstand	Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse
Occupation	Beruf der Person	Tech-support, Craft-repair, Other-service, Sales, Exec-manag., Prof-specialty, Handlers-cleaners, Machine-inspct, Adm-clerical, Farming, Transport, Priv-house-serv, Protective-serv, Armed-Forces
Relationship	Beziehung	Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried
Race	Ethnie	White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black
Sex	Geschlecht	male, female
Country	Herkunftsland	United-States, England, Canada, Germany [...] (41 Länder)

Tabelle 8.1: Merkmalsbeschreibung Adult Datensatz

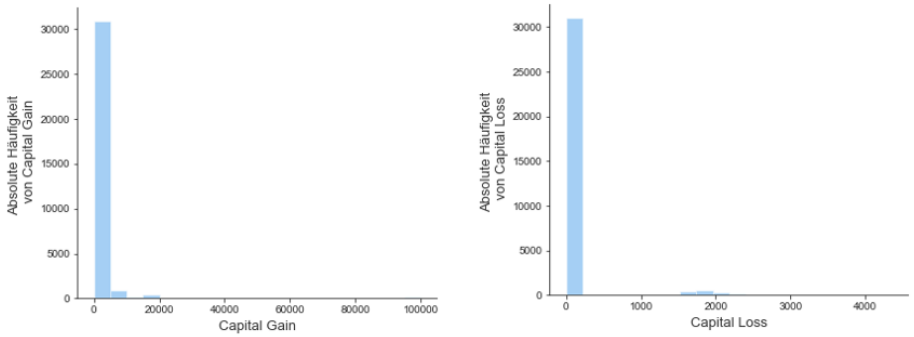


Abbildung 8.2: Adult-Datensatz: Verteilung *Capital Gain* und *Capital Loss*

Capital Gain ist eine kontinuierliche Variable, die Werte zwischen 0 und 100000 annimmt. Der Mittelwert ist 1097 und der Median liegt bei 0, was auf eine starke Rechtsschiefe der Daten hinweist. Anhand des Histogramms (vgl. Abbildung 8.2 Links) lässt sich ablesen, dass sehr viele Personen keine und einige wenige einen sehr große Gewinne an den Kapitalmärkten (zw. 10000 und 99999) erzielt haben.

Capital Loss ist eine kontinuierliche Variable, die Werte zwischen 0 und 4356 annimmt. Der Mittelwert ist 403 und der Median liegt bei 0, was erneut auf eine starke Rechtsschiefe hinweist. Das Histogramm (vgl. Abbildung 8.2 Rechts) zeigt einen ähnlichen Zusammenhang, wie er bereits bereits bei *Capital Gain* zu beobachten war.

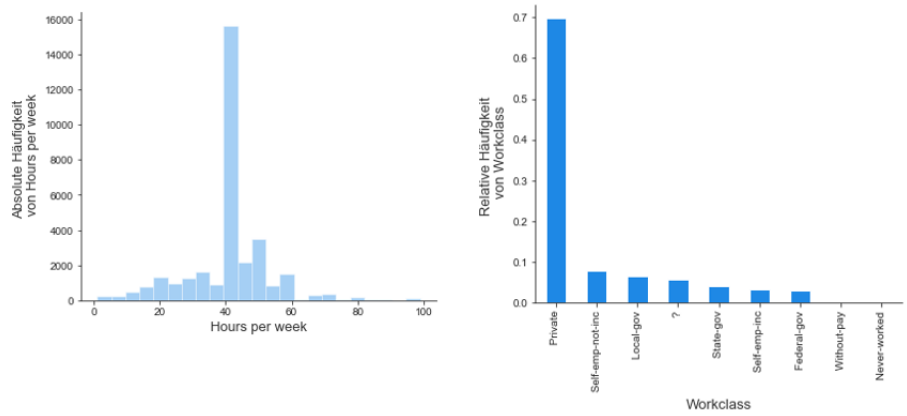


Abbildung 8.3: Adult-Datensatz: Verteilung *Hours per Week* und *Workclass*

Hours per week ist eine kontinuierliche Variable, die Werte zwischen 0 und 99 annimmt. Der Mittelwert ist 40,5 und der Median liegt bei 40. Anhand des Histogramms (vgl. Abbildung 8.3 Links) wird deutlich, dass über 50% der Personen zwischen 35 und 40 Stunden in der Woche arbeiten. Es gibt auch einige extreme Fälle, die über 60 bzw. weniger als 20 Stunden in der Woche arbeiten.

Workclass ist eine kategorielle Variable. Circa 70 Prozent der erhobenen Personen sind in der Privatwirtschaft angestellt. Darüber hinaus ist anhand von Abbildung 8.3 (Rechts) zu erkennen, dass bei 5% der Personen keine Informationen über das Beschäftigungsverhältnis vorliegen und die Kategorien *Never-worked* und *Without-pay* quasi keine Bedeutung haben.

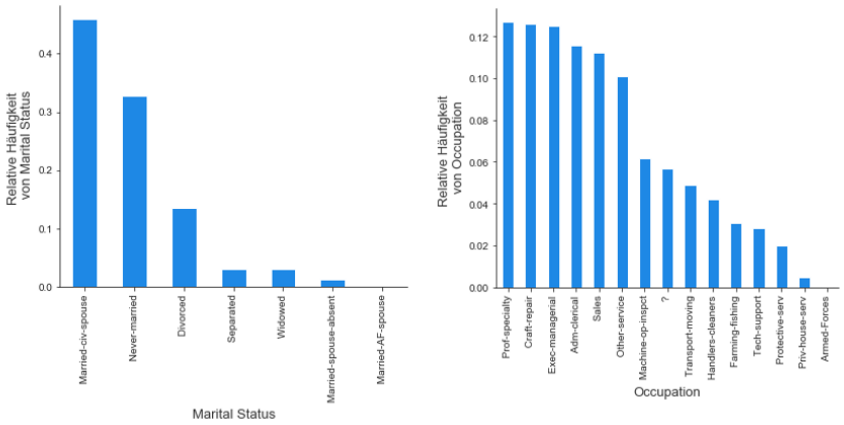


Abbildung 8.4: Adult-Datensatz: Verteilung *Marital Status* und *Occupation*

Marital Status ist ebenfalls eine kategorielle Variable. *Married-civ-spouse* und *Never-married* sind die beiden dominanten Kategorien mit mehr als Zweidrittel der Beobachtungen (vgl. Abbildung 8.4 Links). *Married-AF-spouse* ist die kleinste Kategorie.

Occupation ist die dritte kategorielle Variable. *Prof-specialty, Craft-repair* und *Exec-managerial* sind die drei größten Kategorien mit jeweils 12% der Personen (vgl. Abbildung 8.4 Rechts).

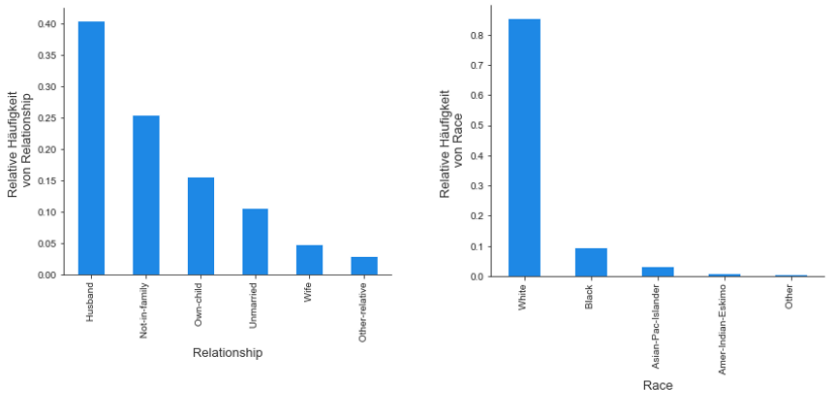


Abbildung 8.5: Adult-Datensatz: Verteilung *Relationship* und *Race*

Die kategorielle Variable ***Relationship*** hat sechs verschiedene Ausprägungen (Abbildung 8.5 Links) und *Husband* ist mit 40% die häufigste Kategorie.

Auffällig ist die Tatsache, dass es zwar sehr viele Ehemänner, allerdings nur vergleichsweise wenige Ehefrauen (ca. 5%) in den Daten gibt. Dies geht einher mit einer ungleichen Verteilung des Geschlechts über den Datensatzd.

Race ist eine kategorielle Variable, mit einer starken Konzentration auf der Ausprägung *White*. Die zweitgrößte Kategorie *Black* trifft nur für 10% der Beobachtungen zu (Abbildung 8.5 Rechts)

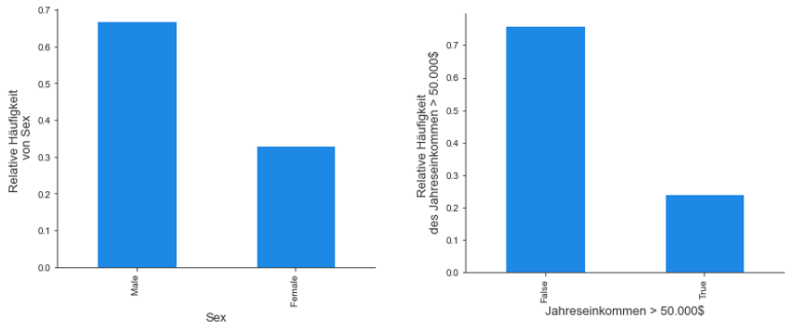


Abbildung 8.6: Adult-Datensatz: Verteilung von *Sex* und der Zielgröße

Die kategorielle Variable ***Sex*** ist nicht gleichverteilt. Es sind ungefähr doppelt so viele Männer wie Frauen befragt worden (vgl. Abbildung 8.6 Links).

Die Variable ***Country*** hat zwar sehr viele verschiedene Kategorien, allerdings entfallen auf die *USA* bereits 91.5% der Beobachtungen. Die anderen Kategorien spielen mit Ausnahme von *Mexico* keine große Rolle. Beides überrascht nicht, da die Datenbasis in den USA erhoben wurde.

Zu guter Letzt folgt die Betrachtung der Zielgröße in Abbildung 8.6 Rechts: das Einkommen einer Person ($>$ 50.000\$ (0) oder $\leq$ 50.000\$ (1)). 76% der Personen haben ein Einkommen kleiner als 50.000\$ und circa 24 Prozent ein Einkommen, das über 50.000 US-Dollar liegt.

8.2 PERFORMANCE DER BASELINE-MODELLE

Es werden drei verschiedene Modelle trainiert: Eine logistische Regression und ein Entscheidungsbaum der Tiefe Vier als erklärbare Modelle, sowie eine Gradient Boosting Machine (GBM), bestehend aus $n = 1000$ Entscheidungsbäumen mit maximal vier Blattknoten, als Black-Box-Modell. Zusätzlich zu diesen drei Modelltypen erfolgt eine Analyse auf zwei verschiedenen Merkmalsräumen: Einmal werden kategorielle Merkmale in Dummy-Variablen zerlegt (One-Hot-Encoding), und einmal lediglich in entsprechende numerische Ausprägungen übersetzt (Label-Encoding). Daraus resultierend entstehen sechs verschiedene Modelle, deren Vergleich anhand der mittleren Performance einer k=5-Kreuzvalidierung erfolgt. Aufgrund der

leichten Unbalanciertheit der Daten wird die ROC-AUC als Performancemaß verwendet. Zur Modellauswahl, Codierung und Evaluation vergleiche Kapitel 5.3.

Nach der Entfernung fehlender Beobachtungen und Codierung der kategoriellen Variablen, sind die beiden White-Box-Modelle und die GBM für jeden der fünf Train-Test-Splits trainierbar. Die gemittelten AUC-Werte der Kreuzvalidierungen finden sich in Tabelle 8.2, wobei die Werte in Klammer im weiteren Verlauf immer die geschätzte Standardabweichung angeben:

Log. Regression		Entscheidungsbaum		Grad. Boost. Machine	
One-Hot	Label	One-Hot	Label	One-Hot	Label
0.906		0.870	0.871	0.928	0.928
(0.004)		(0.004)	(0.004)	(0.002)	(0.002)

Tabelle 8.2: Performance Baseline-Modelle Adult

Das Black-Box-Modell hat die beste Performance - unabhängig von der Codierung der kategoriellen Variablen. Die logistische Regression ist das überlegene der beiden erklärbaren Modelle. Der Entscheidungsbaum ist auf dem Label-Encoded-Datensatz minimal besser. Die logistische Regression ist nur für One-Hot-Encoding der kategoriellen Merkmale sinnvoll berechenbar. Im Falle von Label-Encoding behandelt die logistische Regression kategorielle Variablen wie stetige Merkmale. Darüber hinaus ist die geringere Standardabweichung der GBM festzustellen, wobei die Standardabweichung über alle Verfahren und Codierungen hinweg eher gering ist.

Zusätzlich soll die statistische Signifikanz des Unterschieds zwischen der Performance des besten White-Box-Modells (Logistische Regression) und der Gradient Boosting Machine ermittelt werden. Dies erfolgt mittels der in Kapitel 5.3.4 vorgestellten 5x2CV. Die schließt das Ermitteln der Performance beider Modelle für fünf unabhängige k=2 Kreuzvalidierungen ein. Dadurch ergeben sich 10 AUC-Werte für jedes Modell. Anschließend wird ein verbundener t-Test berechnet, um zu untersuchen, ob sich die AUC-Werte der logistischen Regression signifikant von jenen der Gradient Boosting Machine unterscheiden.

In der anhängenden Tabelle A.1 finden sich die 10 AUC-Werte beider Modelle. Das Berechnen der verbundenen T-Statistik ergibt einen t-Wert von -40.16, was zu einem p-Wert von $1.8 \cdot 10^{-18}$ führt, so dass die Null-Hypothese einer Differenz von 0 zwischen den beiden Modellen, zu einem Fehlerniveau von 5 Prozent zu verwerfen ist. Daraus folgt, dass die Gradient Boosting Machine statistisch signifikant bessere Ergebnisse liefert als die logistische Regression. An dieser Stelle sei nochmals auf die Hinweise zu Einschränkungen dieser Testmethodik aus Kapitel 5.3.4 verwiesen. Im nächsten Kapi-

tel werden Erklärungen des Black-Box-Modells mittels SHAP erzeugt und anhand dieser die unwichtigen Merkmale ermittelt.

8.3 FEATURE ENGINEERING MITTELS ERKLÄRUNGEN

Zunächst wird die Methodik aus Kapitel 6 Anwendung finden und das Feature Engineering mittels der SHAP Erklärungen der Gradient Boosting Machine durchgeführt. Im ersten Schritt erfolgt das Entfernen irrelevanter Merkmale, gefolgt von nichtlinearen Transformationen, bevor Wechselwirkungen Eingang in das Modell finden. Die Erklärungen des Black-Box-Modells und damit auch die Transformationen des Merkmalsraums werden auf Basis der Trainingsdaten eines 80-20 Splits bestimmt, um eine Data Leakage zu vermeiden (vgl. dazu Kapitel 5.3.4).

8.3.1 *Entfernen unwichtiger Merkmale*

Die GBM wurde mittels des Python-Package *scikit-learn* implementiert. Das Paket berechnet die globale Feature Importance über die Gini Wichtigkeit (Kapitel 4.3.1). In Abbildung 8.7 findet sich die so ermittelte Wichtigkeit der einzelnen Merkmale. Dabei wurden alle Merkmale in Relation zum wichtigsten Feature *Relationship* normiert; d.h. die Wichtigkeit des Merkmals *Education-Num* ist ungefähr halb so groß wie die von *Relationship*.

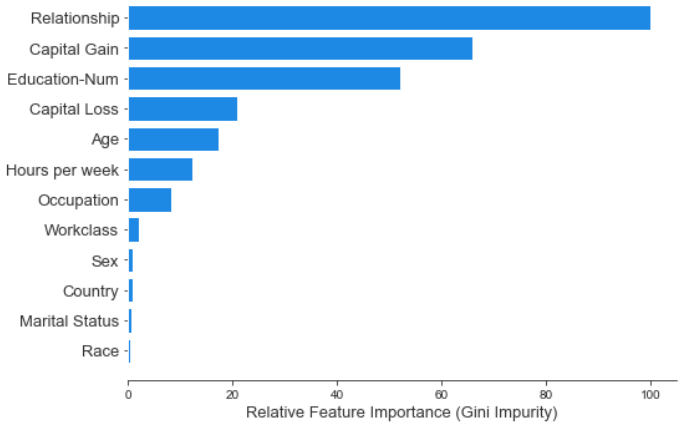


Abbildung 8.7: Feature Importance GBM (Adult-Datensatz)

Relationship, *Capital Gain* und *Education-Num* sind hier die mit Abstand wichtigsten Merkmale. Es folgen *Capital Loss*, *Age*, *Hours per week* und *Occupation*, wobei diese noch eine Relevanz von 10 bis 20 Prozent der Wichtigkeit von *Relationship* vorzuweisen haben. Die verbleibenden Merkmale sind alle vergleichsweise unbedeutend aus Sicht der Gini Wichtigkeit.

Zusätzlich wird die Merkmalsrelevanz mittels SHAP bestimmt (Kapitel 4.6.4):

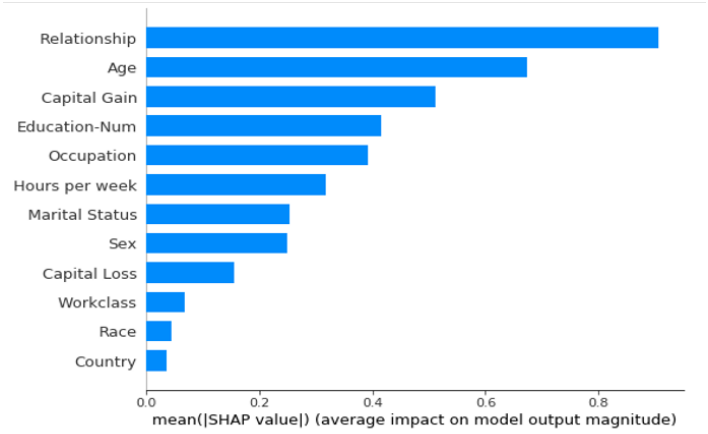


Abbildung 8.8: Feature Importance SHAP (Adult-Datensatz)

Die Relevanz einzelner Merkmale unterscheidet sich deutlich von der mittels Gini Importance ermittelten. *Relationship* ist erneut die wichtigste Variable des Modells. Mit *Age* folgt dann allerdings ein Merkmal, dass mittels SHAP einen deutlich stärkeren Einfluss zugewiesen bekommt.
Eine weitere auffällige Abweichung betrifft das Feature *Capital Loss*. Während SHAP dieses für vergleichsweise irrelevant hält, misst die Gini Wichtigkeit ihm eine große Relevanz zu. Wie kommen diese großen Unterschiede zustande, und was bedeutet dies für das eigentliche Problem - die Entfernung der irrelevanten Merkmale?

Der erste Teil der Frage hat eine vergleichsweise kurze und naheliegende Antwort: Die Gini Wichtigkeit neigt dazu, kategorielle Variablen mit vielen Ausprägungen und stetige Merkmale mit großem Wertebereich zu überschätzen (groß ist dabei immer in Relation zum Wertebereich der anderen Merkmale des Datenraums zu sehen) (vgl. Kapitel 4.3.1). Dieser Effekt ist vermutlich bei der Variable *Capital Loss* eingetreten. Die Variable hat einen vergleichsweise großen Wertebereich und wird daher durch die beiden Verfahren sehr unterschiedlich eingeschätzt.

Nun zum zweiten Teil der Frage: Welche Merkmale sind aus dem Datenraum zu entfernen, um die Erklärbarkeit des Modells zu erhöhen, ohne dessen Performance einzuschränken? Nach Abbildung 8.7 wären die Merkmale *Workclass*, *Sex*, *Country*, *Marital Status* und *Race* potenzielle Kandidaten, um aus dem Modell entfernt zu werden. SHAP kommt - wenn auch in anderer Reihenfolge - bei *Race*, *Country* und *Workclass* zu einem ähnlichen Urteil in Bezug auf die Wichtigkeit dieser erklärenden Variablen. Bei *Marital Status*, *Capital Loss* und *Sex* ist die Aussage nicht eindeutig, kommen die beiden Maße doch zu sehr unterschiedlichen Ergebnissen.
Für das Feature *Capital Loss* wurden bereits mögliche Ursachen erläutert. Die Wichtigkeit dieses Merkmals aus Sicht der Gini Wichtigkeit ist zu groß, um es zu entfernen. Bei *Marital Status* und *Sex* erfolgt die Entscheidung unter Betrachtung des *SHAP Summary Plots* (Abbildung 8.9), um ein differenzier-

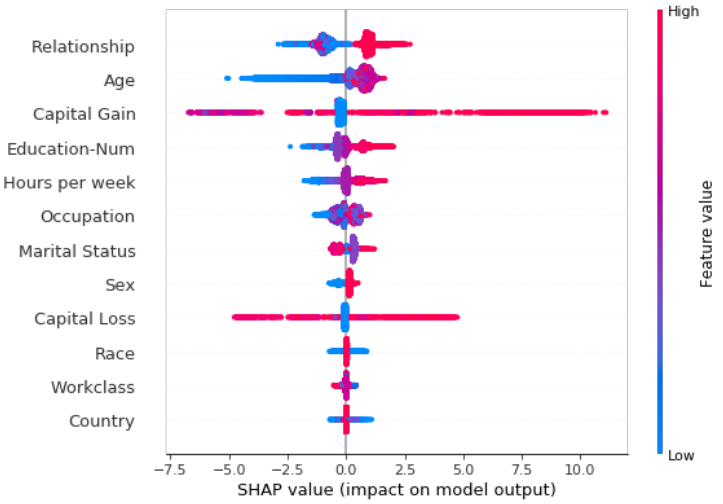


Abbildung 8.9: *SHAP Summary Plot* (Adult-Datensatz)

teres Bild von der prädiktiven Trennkraft der Merkmale zu erhalten:

Bevor die Implikationen des Plots für die Problemstellung dargelegt werden, nochmals eine kurze erklärende Beschreibung des Plots. Auf der y-Achse werden die verschiedenen Feature abgetragen, so dass jedes Merkmal eine eigene *Achse* bekommt. Die Reihenfolge der Variablen richtet sich dabei nach der globalen Wichtigkeit. Auf der x-Achse befindet sich der tatsächliche Shapley-Werte einer Beobachtung. Jede Beobachtung wird mit einem Punkt auf jeder der Merkmalsachsen repräsentiert. Die Farb-Codierung gibt die entsprechende Merkmalsausprägung an (von niedrig bis hoch). Treten SHAP Werte mehrfach auf, werden diese aufeinander gestapelt.

Besonders auffällig sind die Variablen *Capital Gain* und *Capital Loss*, haben diese doch eine sehr breite Verteilung der SHAP Werte. Bei einem Großteil der Beobachtungen - diejenigen ohne Einkünfte bzw. Verluste an den Kapitalmärkten - haben die beiden Feature quasi keinen Einfluss auf die Vorhersage. Nehmen sie stattdessen eine Ausprägung größer als Null an, ist deren Einfluss sehr groß und nicht eindeutig in Bezug auf die Richtung. Daraus lässt sich folgern, dass das Merkmal nichtmonoton in Hinblick auf seinen Einfluss ist. Diese Tatsache macht es den erklärbaren Modellen - insbesondere der logistischen Regression - schwer, den Einfluss dieser Merkmale entsprechend eingehen zu lassen. Dies wird später mittels der Methodik aus Kapitel 7 nochmals genauer analysiert.

Zurück zur Frage nach der Relevanz von *Sex* und *Marital Status*. *Sex* hat tendenziell eher geringe SHAP Werte mit geringer Streuung. Männer haben in diesem Modell alleine aufgrund der Tatsache ihres Geschlechtes eine größere Chance ein Einkommen über 50.000$ zu erwirtschaften. Bei Frauen ist der Einfluss hingegen negativ und streut ebenfalls nicht besonders stark.

Bei *Marital Status* ergibt sich ein ähnliches Bild. Im Vergleich zu den anderen Merkmalen ist die Streuung eher gering. Abhängig von der Ausprägung der Variablen lassen sich die Effekte auf die Vorhersagewahrscheinlichkeit in eher negativ, eher neutral und eher positiv unterteilen. Zusammenfassend ist zu sagen, dass sowohl das Geschlecht als auch der Beziehungsstatus eine prädiktive Trennkraft vorweisen können und daher nicht aus dem Datenraum entfernt werden.

Die Auswirkungen dieser Reduktion auf die Modellperformance sind in Tabelle 8.3 zusammengefasst. Dabei wurde - wie auch im weiteren Verlauf der Arbeit - das Black-Box-Modell auf dem transformierten Feature-Space ebenfalls neu trainiert. Die runde Klammer enthält erneut die Standardabweichung über die Kreuzvalidierung.

Merkmal	**Log. Regression**	**Entscheid'baum**		**Grad. Boost. M.**	
	One-Hot	One-Hot	Label	One-Hot	Label
Baseline	0.906 (0.004)	0.870 (0.004)	0.871 (0.004)	0.928 (0.002)	0.928 (0.002)
Merkmals-relevanz	0.905 (0.004)	0.870 (0.004)	0.871 (0.004)	0.927 (0.002)	0.927 (0.002)

Tabelle 8.3: Performance nach Entfernung irrelevanter Merkmale

Für die logistische Regression und die Gradient Boosting Machine führt die Entfernung der Merkmale zu einer minimalen Reduktion der Modellperformance um 0.01 AUC-Punkte. Auf den Entscheidungsbaum hat diese Transformation hingegen keine Auswirkungen. Dies liegt in der geringen Tiefe des Baumes begründet. Der Baum hat die entfernten Merkmale schlicht nicht als Split-Merkmale herangezogen.

Die Performance der logistischen Regression verschlechtertsich zwar minimal, durch die Entfernung der irrelevanten Merkmale wurde allerdings die Komplexität des Modells verringert und dadurch die Erklärbarkeit erhöht (vgl. hierzu auch Kapitel 6.2.1). In diesem Beispiel wurde die Dimensionalität des Merkmalsraums der logistischen Regression um 51 Dimensionen reduziert. So viele Dummy-Variablen waren notwendig, um die 41 Länder, die fünf verschiedenen Ethnien und acht Beschäftigungsverhältnisse zu repräsentieren.

8.3.2 *Integration von Nichtlinearitäten*

Es folgt für die verbleibenden Merkmale die Anwendung der Methodik aus Kapitel 6.2.2, um Merkmale zu transformieren und so nichtlineare Zusammenhänge in die erklärbaren Modelle zu integrieren. Die Reihenfolge, in der

die Transformationen durchgeführt orientiert sich an der SHAP Merkmalsrelevanz aus Abbildung 8.8. Wurde ein Merkmal transformiert verbleibt es im erklärbaren Modell; die Merkmalstransformationen werden nicht isoliert sondern schrittweise durchgeführt. Das Integrieren aller Nichtlinearitäten führt zu folgenden Veränderungen in Bezug auf die Performance der Modelle:

Merkmal	**Log. Regression**	**Entscheid'baum**		**Grad. Boost. M.**	
	One-Hot	One-Hot	Label	One-Hot	Label
Baseline	0.906	0.870	0.871	0.928	0.928
	(0.004)	(0.004)	(0.004)	(0.002)	(0.002)
Merkmals-	0.905	0.870	0.871	0.927	0.927
relevanz	(0.004)	(0.004)	(0.004)	(0.002)	(0.002)
Nichtlinearitäten	0.914	0.868	0.872	0.915	0.914
	(0.003)	(0.004)	(0.004)	(0.003)	(0.003)

Tabelle 8.4: Performance nach Integration von Nichtlinearitäten

Eine Tabelle, die die Auswirkung jeder einzelnen Transformationen auf die Area-under-the-Curve zeigt, befindet sich im Anhang (Tabelle A.4). Hier wird stattdessen Abbildung 8.10 betrachtet, in der diese Veränderungen über die verschiedenen Merkmale hinweg visualisiert sind. Anhand dieser lässt sich unmittelbar erkennen, welche Transformationen zu sehr großen Veränderungen der Leistungsfähigkeit der Modelle führen.

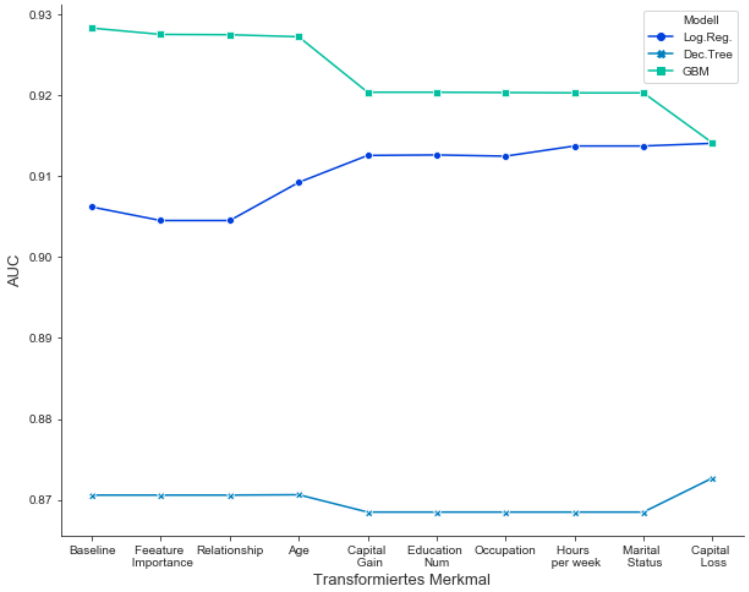


Abbildung 8.10: Entwicklung der Performance bei Transformation der Merkmale

Während sich die Performance der logistischen Regression deutlich verbessert, profitiert der Entscheidungsbaum nicht durch die Transformation

der Merkmale. Wenn sich die Gradient Boosting Machine mit den transformierten Merkmalsräumen konfrontiert sieht, sinkt deren Performance deutlich. Bevor die Gründe für die Auswirkungen der Integration von Nichtlinearitäten auf den Entscheidungsbaum und die GBM erläutert werden, folgt eine Untersuchung der Merkmale, die zu großen Verbesserung bzw. Verschlechterungen der Performance führen: *Age*, *Capital Gain* und *Capital Loss*.

Age

Das Merkmal *Age* war bereits als illustrierendes Beispiel in Kapitel 6.2.2 betrachtet worden. Der *SHAP Dependence Plot* in Abbildung 8.11 zeigt einen eindeutig nichtlinearen Einfluss des Alters in der GBM.

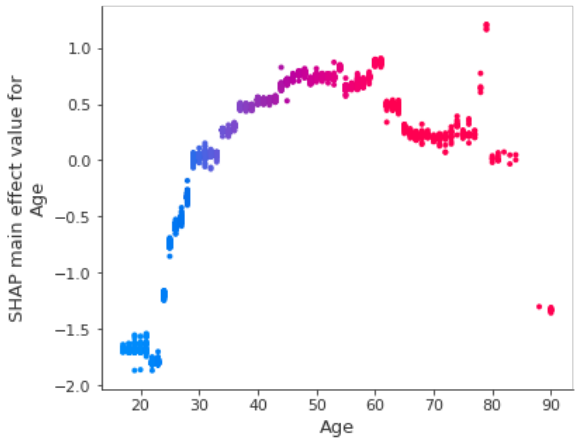


Abbildung 8.11: SHAP Dependence *Age* Haupteffekt (Adult-Datensatz)

Die Betrachtung des Bestimmtheitsmaßes R^2 bestätigt diesen visuellen Eindruck (vgl. Abbildung 8.12). Beim Versuch, Polynome vom Grad 1 bis 5 an den Einfluss des Features zu fitten, erzielt ein lineares Polynom lediglich ein R^2 von ungefähr 0.55. Eine quadratische Funktion erreicht stattdessen eine Güte der Approximation von über 0.90. Danach steigt die Anpassung nicht mehr durch das Erhöhen des Grads der Polynome.

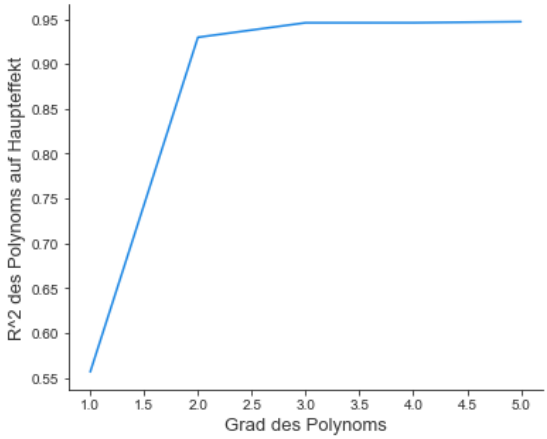


Abbildung 8.12: SHAP Dependence *Age* Polynom-Fit

Das heißt, als Transformationsfunktion des Alters wird das folgende quadratische Polynom verwendet:

$$t(\text{Age}) = -0.002 \cdot \text{Age}^2 + 0.236 \cdot \text{Age} - 5.37$$

Dass diese Transformation die Trennschärfe der logistischen Regression deutlich erhöht, überrascht wenig, wenn die schlechte Anpassung einer linearen Beziehung zwischen *Age* und dem Einfluss auf die Modellvorhersage bedacht wird.

Capital Gain

Das Transformieren des Merkmals *Capital Gain* führt ebenfalls zu einer deutlichen Verbesserung der Trennschärfe der logistischen Regression. Darüber hinaus verschlechtert sich die Performance des Black-Box-Modells dadurch erheblich. In Abbildung 8.13 ist der GBM-Einfluss über den gesamten Merkmalsraum dargestellt.

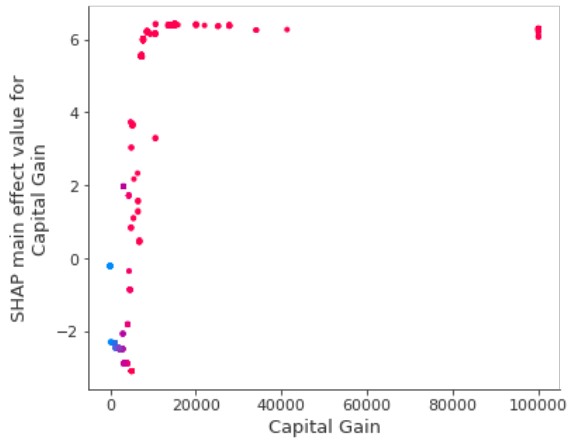


Abbildung 8.13: SHAP Dependence *Capital Gain* Haupteffekt (Adult-Datensatz)

Dabei fallen zwei Dinge besonders auf: Die Darstellung ist durch die Beobachtungen mit sehr großem *Capital Gain* verzerrt. Darüber hinaus ist kein - durch Polynome beschreibbarer - funktionaler Zusammenhang erkennbar. Daher wird das Merkmal in drei verschiedene Kategorien gesplittet:
Die Beobachtungen mit einem niedrigen *Capital Gain* von unter 4200 $, was negativen Einfluss auf die Vorhersage hat. Eine Klasse mit den besonders hohen Gewinnen am Kapitalmarkt ($\geq$ 7500$) - was stark positiven Einfluss auf die Vorhersage hat - und eine dritte Klasse mit den übrigen, mittleren Beobachtungen - deren Einfluss stark schwankt:

$$\text{Capital Gain}_{\text{trans.}} := \begin{cases} \text{niedrig} & \text{für Capital Gain} < 4200\$ \\ \text{hoch} & \text{für Capital Gain} \geq 7500\$ \\ \text{mittel} & \text{sonst.} \end{cases} \tag{8.1}$$

Capital Loss

Die Transformation des Merkmals *Capital Loss* bewirkt für die logistische Regression nur noch eine marginale Steigerung der Trennschärfe. Der Entscheidungsbaum profitiert von dieser Transformation, während die GBM einen erheblichen Performanceverlust erleidet. In Abbildung 8.14 ist erneut der Einfluss über den gesamten Merkmalsraum der GBM dargestellt.

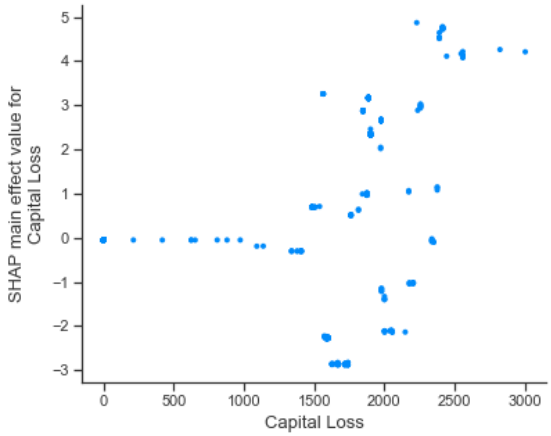


Abbildung 8.14: SHAP Dependence *Capital Loss* Haupteffekt (Adult-Datensatz)

Hier ist auf den ersten Blick ebenfalls kein funktionaler Zusammenhang erkennbar. Allerdings hat auch in diesem Beispiel ein *Capital Loss* von ungefähr kleiner als 1300 keinen Einfluss auf die Modellvorhersage. Für Werte größer 2400 ist der Einfluss hingegen sehr stark positiv. Der Bereich zwischen 1300 und 2400 lässt sich hingegen nur schwer charakterisieren. Abbildung 6.6 zeigt die Klassierung des Merkmals anhand dieser Splits:

$$\text{Capital Loss}_{\text{trans.}} := \begin{cases} \text{niedrig} & \text{für Capital Loss} < 1300\$ \\ \text{hoch} & \text{für Capital Loss} \geq 2400\$ \\ \text{mittel} & \text{sonst.} \end{cases} \tag{8.2}$$

Warum führt die Transformation von *Capital Gain* bzw. *Capital Loss* zu einer deutlichen Verbesserung der erklärbaren Modelle, verschlechtert aber die GBM so stark?

Der Einfluss der beiden Merkmale ist stark nichtlinear, d.h. mit den Mitteln der logistischen Regression nicht gut zu messen. Auf Grund der Tatsache, dass der Einfluss stark schwankend ist, fällt es schwer, mittels weniger Splits - die ein kurzer, erklärbarer Entscheidungsbaum zur Verfügung hat - die relevanten Bereiche des Merkmals zu identifizieren. Durch die Kategorisierung ist es den erklärbaren Modellen allerdings möglich, einen Teil der relevanten Dateninstanzen (solche mit sehr niedrigem bzw. sehr hohem *Capital Loss/Gain*) direkt zu identifizieren.

Aus Sicht der Gradient Boosting Machine erfolgte hingegen eine starke Komprimierung des Merkmals und viele relevante Informationen wurden entfernt. Durch die Tatsache, dass die GBM 1000 Bäume zur Verfügung hat, kann sie detailliert einzelne Teilbereiche der Daten identifizieren, in denen *Capital Gain* bzw. *Capital Loss* positiven oder negativen Einfluss haben. Die Vereinfachung bzw. implizite Aggregation von ähnlichen Beobachtungen, welche durch die Transformationen des Merkmalsraums entsteht, führt zu der Verschlechterung der GBM.

8.3.3 *Integration von Interaktionen*

Nachdem irrelevante Merkmale entfernt und nichtlineare (Haupt-)Effekte integriert wurden, widmet sich der letzte Schritt des Feature Engineerings mittels der Erklärungen von Black-Box-Modellen den Interaktionen zwischen zwei Merkmalen. Es werden die in Kapitel 6.2.3 vorgestellten Methoden auf die stärksten Interaktionen angewendet. Dabei erfolgt nur noch eine Analyse der für die jeweiligen Verfahren relevanten Codierungen kategorieller Merkmale: Also One-Hot-Encoding für die logistische Regression und Label-Codierung für die beiden baumbasierten Verfahren.

Um die starken Interaktionen zu erkennen, ist in Abbildung 8.15 die Heatmap der zweidimensionalen Interaktionseffekte aller Merkmale zu sehen. An dieser Stelle sei nochmals an die Bedeutung der Effektstärke im Zusammenhang mit dem *Signal-zu-Rausch-Verhältnis* aus Kapitel 6.2.3 erinnert.

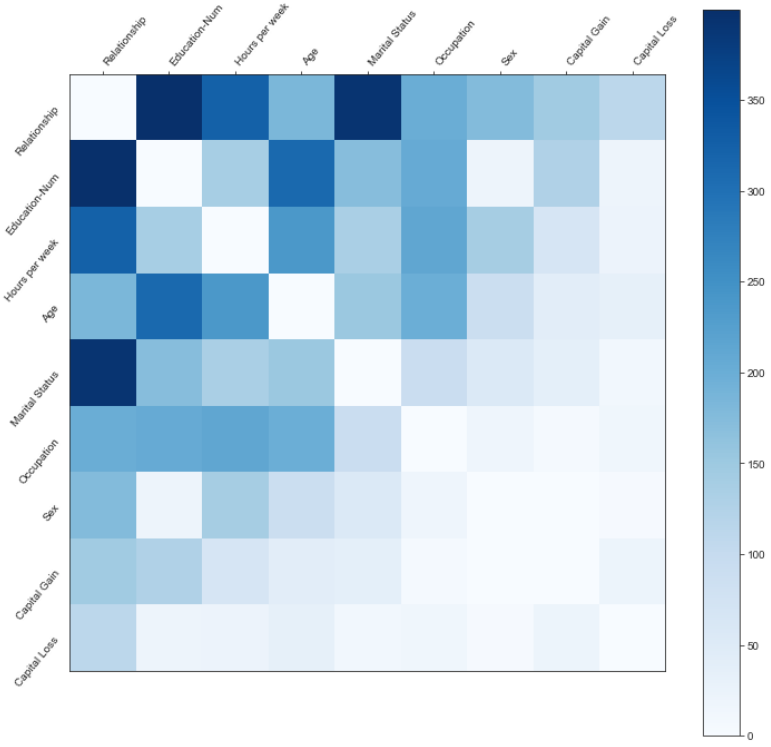


Abbildung 8.15: Heatmap der Interaktionseffekte (Adult-Datensatz)

Die Interaktionen werden in der Reihenfolge ihrer Effektstärke hinzugefügt. In der folgenden Tabelle finden sich die daraus resultierenden Veränderungen der Performance:

Merkmal	**Log. Regression** One-Hot	**Entscheid'baum** Label	**Grad. Boost. M.** Label
Baseline	0.906 (0.004)	0.871 (0.004)	0.928 (0.002)
Merkmalsrelevanz	0.905 (0.004)	0.871 (0.004)	0.927 (0.002)
Nichtlinearitäten	0.9140 (0.0029)	0.8727 (0.0036)	0.9141 (0.0026)
Education-Num x Relationship	0.9142 (0.0029)	0.8719 (0.0031)	0.9146 (0.0030)
Hours per Week x Relationship	0.9148 (0.0029)	0.8696 (0.0043)	0.9141 (0.0029)
Education-Num x Age	0.9149 (0.0028)	0.8677 (0.0065)	0.9138 (0.0029)
Hours per Week x Age	0.9150 (0.0028)	0.8677 (0.0065)	0.9138 (0.0028)

Tabelle 8.5: Performance nach Integration von Interaktionen

Die Performance des Entscheidungsbaums und auch der Gradient Boosting Machine verschlechtern sich. Für die GBM gilt dabei die gleiche Begründung wie schon zuvor bei der Integration von Nichtlinearitäten. Die Approximation über Polynome stellt eine starke Vereinfachung der durch das Black-Box-Modell ermittelten Merkmalseinflüsse dar. Diese Reduktion macht es der GBM quasi unmöglich, ihre Vorteile gegenüber den erklärbaren Modellen auszuspielen. Daher verschlechtert sich die Trennschärfe der GBM.

Bei der Integration von Interaktionen kommt noch ein weiterer Effekt in Bezug auf die Eigenschaften eines Entscheidungsbaums zum Tragen. Im Gegensatz zu Regressionsverfahren sind Bäume dazu imstande Interaktionen zwischen Merkmalen zu lernen. Vergleiche hierzu auch Tabelle 5.1. Händisches Integrieren dieser Wechselwirkungen in den Merkmalsraum erschwert das Training von Entscheidungsbäumen, und damit auch der GBM, da die stärksten Interaktionen bereits als eigenes Merkmal codiert sind.

Die Performance der logistischen Regression verbessert sich zwar, allerdings nur noch geringfügig. Dabei sind die Interaktionen eindeutig zu erkennen und lassen sich auch gut durch Polynome vom Grad kleiner 6 beschreiben. Zur Illustration dient die Wechselwirkung in Abbildung 8.16 zwischen

Education-Num und *Relationship*, welche bereits in Kapitel 6.2.3 als Beispiel verwendet wurde.

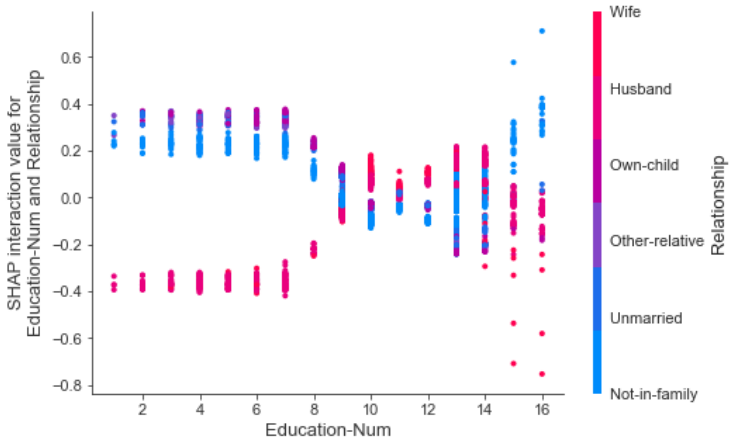


Abbildung 8.16: SHAP Interaktion *Education-Num* und *Relationship*

Allerdings wurde bereits bei der Vorstellung der Methodik zur Integration von Wechselwirkungen in Abschnitt 6.2.3 auf die Relevanz der Effektgröße einer Interaktion hingewiesen. Auch wenn hier bereits nur noch die besonders starken Wechselwirkungen Eingang in die Analyse finden, so sind diese in Relation zur Größe des Haupteffekts eher gering. Zur Illustration findet sich in Abbildung 8.17 die gleiche Heatmap wie zu Beginn dieses Abschnitts, allerdings mit den Haupteffekten auf der Hauptdiagonalen.

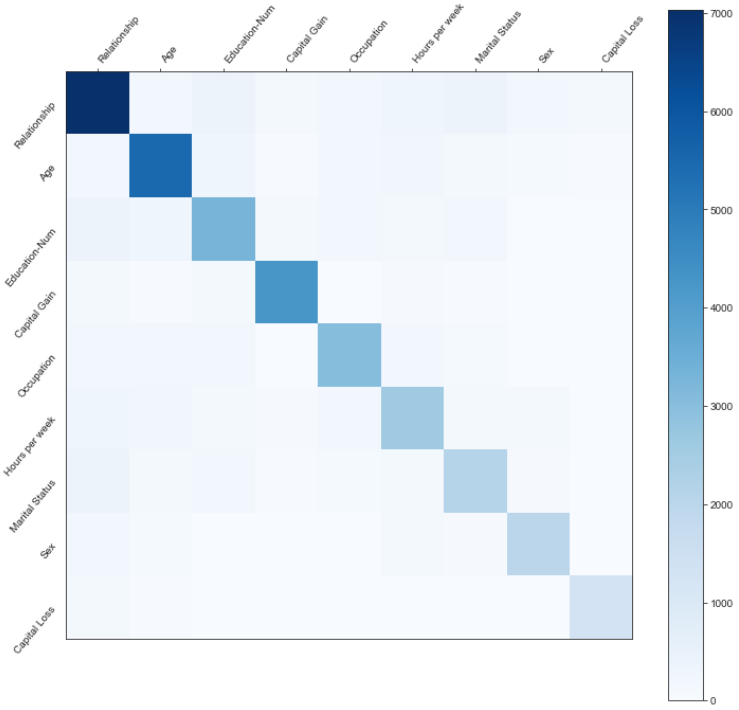


Abbildung 8.17: Heatmap der Interaktionseffekte und Haupteffekte (Adult-Datensatz)

Wie bereits bei der Einführung dieser Visualisierung in Kapitel 6.2.3 angedeutet, dominieren die Haupteffekte nun vollständig die Heatmap. Es findet durch diese Grafik quasi keine Differenzierung mehr zwischen der Stärke der Interaktionseffekte statt - alle erscheinen ungefähr gleich wichtig. Das Betrachten der Werte der Farbskala im Vergleich zu denen aus Abbildung 8.15 zeigt, dass selbst der geringste Haupteffekt immer noch deutlich stärker ist, als der der relevantesten Wechselwirkung. Aus diesem Grund verbessert sich die Performance der logistischen Regression nur noch geringfügig - die Effekte sind vergleichsweise gering.

8.3.4 *Evaluation der logistischen Regression (erweitert)*

Nachdem einmal das Vorgehen aus Kapitel 6 vollständig durchlaufen wurde, folgt eine erste Evaluation der Ergebnisse. Zunächst die AUC-Ergebnisse nochmals in Form einer Tabelle:

Modell	**Log. Regression**	**Entscheid'baum**	**Grad. Boost. M.**
Baseline	0.906	0.871	0.928
	(0.004)	(0.004)	(0.002)
Bestes Modell	0.915	0.872	0.928
	(0.003)	(0.004)	(0.002)

Tabelle 8.6: Performance der bislang besten Modelle

Die Zeile *Bestes Modell* repräsentiert das leistungsfähigste Modell jedes Modelltyps. Dabei wurden nur Modelle am Ende eines Verarbeitungsschritts in Betracht gezogen. Die logistische Regression (im weiteren Verlauf als logistische Regression erweitert bezeichnet) hatte nach Integration von Interaktionen die beste Performance, der Entscheidungsbaum nach der Einbeziehung von Nichtlinearitäten in das Modell, und die GBM performte auf dem Original-Merkmalsraum am besten.

Zu Beginn des Kapitels 8.2 zeigte sich die statistisch signifikant bessere Trennschärfe der GBM zum 5%-Niveau. Nachdem das Feature Engineering mittels der Erklärung von Black-Box-Modellen durchgeführt wurde, stellt sich die Frage ob die Performance der logistischen Regression verbessert werden konnte; und daran anschließend, ob das Black-Box-Modell nach wie vor eine überlegene Trennschärfe vorweisen kann. Für beide Fragen wird erneut die 5x2CV-Methodik (vgl. Kapitel 5.3.4) Anwendung finden. Die Werte der drei Modelle über die fünf Kreuzvalidierungen finden sich in Tabelle A.2 im Anhang.

Log. Regression vs. Log. Regression (erweitert)
Das Berechnen der verbundenen T-Statistik für die beiden logistischen Regressionsmodelle ergibt einen t-Wert von 18,857, was zu einem p-Wert von

$1.5 \cdot 10^{-8}$ führt, so dass die Null-Hypothese einer Differenz von 0 zwischen den beiden Modellen zum 5%-Niveau zu verwerfen ist. Daraus lässt sich folgern, dass das Feature Engineering zu einer statistisch signifikanten Verbesserung der Trennschärfe der logistischen Regression führt.

Gradient Boosting Machine vs. Log. Regression (erweitert)
Das Berechnen der verbundenen T-Statistik für den Vergleich zwischen Black- und verbessertem White-Box-Modell ergibt einen t-Wert von -43.9 und einem p-Wert von $1.3 \cdot 10^{-12}$, so dass nach wie vor ein zum 5%-Niveau signifikanter Unterschied zwischen der Performance von Black- und White-Box-Modell vorliegt.

8.4 GENERIERUNG NEUER MERKMALE ANHAND DER VORHERSAGEDIFFERENZ

Mittels der in Kapitel 6 vorgestellten Methodik konnte die Performance der logistischen Regression zum 5%-Niveau signifikant verbessert werden. Allerdings bleibt sie nach wie vor deutlich hinter der des Black-Box-Modell zurück. Es stellt sich die Frage, inwieweit die überlegene Trennschärfe der Gradient Boosting Machine nachvollziehbar ist. Dazu wird die in Kapitel 7 vorgestellte Methodik Anwendung finden, um die Differenz der Prognose zwischen GBM und logistischer Regression (erweitert) zu analysieren.
In der folgende Tabelle 8.7 finden sich die AUC-Werte für einen zufällig erzeugten 80-20 Train-Test-Split des Adult-Datensatzes:

Modell	Performance (Train)	Performance (Test)
Gradient Boosting M.	0.9402	0.9260
Log. Reg. (erweitert)	0.9159	0.9135

Tabelle 8.7: Performance Baseline-Modelle zur Analyse der Vorhersagedifferenz

8.4.1 *Bestimmung des Schwellwertes zur Differenzbildung*

Die Berechnung der Klassifikationsdifferenz erfolgt ausschließlich für Beobachtungen im Trainingsdatensatz, um ein Durchsickern von Informationen aus dem Testset in das Modelltraining zu verhindern. In Abbildung 8.18 ist die Verteilung der Prognosedifferenz zwischen dem logistischen Regressionsmodell und der Gradient Boosting Machine dargestellt. Die Differenz ist sehr symmetrisch verteilt und führt bei einem Großteil der Beobachtungen zu einer Differenz nahe 0, was in Anbetracht der guten Performance beider Modelle mit AUC-Werten von über 0.9 nicht überrascht.
Die Bestimmung des Schwellwertes erfolgt anhand der in Abschnitt 7.1.2 entwickelten Methodik. Dazu wird der Schwellwert sequentiell erhöht bis die Performance des White-Box-Modells bei einer AUC von ungefähr 0.5 liegt - der Performance eines zufälligen Modells. Das Visualisieren der Per-

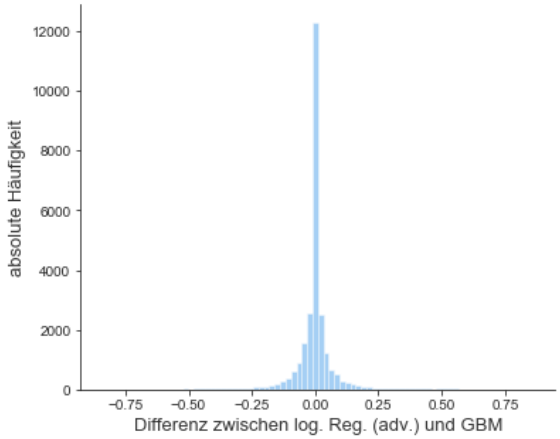


Abbildung 8.18: Histogramm der Vorhersagedifferenz zwischen Log. Reg. (erw.) und GBM

formanceentwicklung bei sequentieller Erhöhung des Cut-Off-Wertes ergibt Abbildung 8.19. Die Performance der logistischen Regression ist auf den

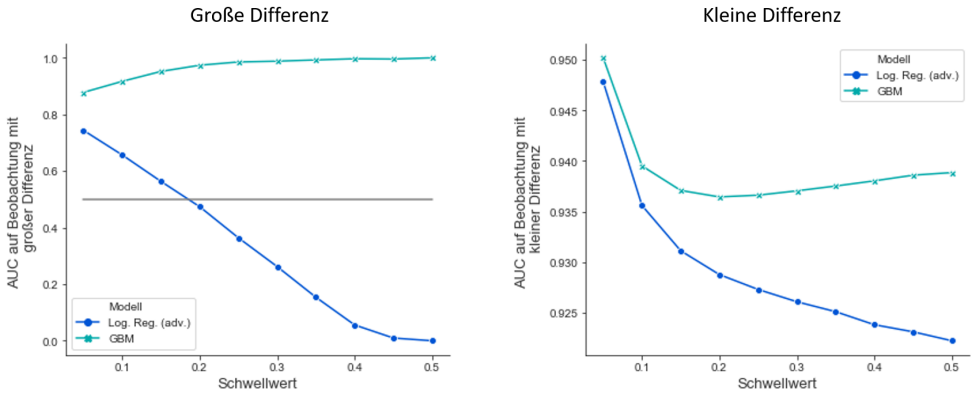


Abbildung 8.19: Entwicklung der Performance über verschiedene Schwellwerte

durch den Schwellwert determinierten Beobachtungen mit großer Vorhersagedifferenz monoton fallend. Bei einem Schwellwert von ungefähr 0.2 fällt die Performance unter die eines zufälligen Modells. Daher wird in diesem Beispiel mit einem Cut-Off von 0.2 weitergearbeitet.

Auffällig ist die Entwicklung der Performance der GBM, verhält sie sich doch konträr zu der des White-Box-Modells. Je höher der Cut-Off, desto besser wird die Performance des Black-Box-Modells. Das bedeutet, dass die GBM vor allem diejenigen Beobachtungen besonders gut prognostizieren kann, die stark von der Struktur der logistischen Regression abweichen und so zu einer betragsmäßig großen Differenz zwischen den prognostizierten Klassenwahrscheinlichkeiten führen. Dies unterstreicht nochmals die Eigenschaft gradientenbasierter Ensemble Methoden, sehr spezielle Teilbereiche der Daten gut wiedergeben zu können. In der folgenden Tabelle 8.8 finden

sich die AUC-Werte der Modelle für Beobachtungen mit großer bzw. kleiner Differenz.

Modell	Große Differenz	Kleine Differenz	Train
Gradient Boosting M.	0.9794	0.9365	0.9402
Log. Reg. (erweitert)	0.4896	0.9283	0.9159
Anzahl Datenpunkte	1254	24794	26048

Tabelle 8.8: Performance von logistischer Regression (erweitert) und GBM auf Teilmenge mit großer bzw. kleiner Vorhersagedifferenz

Dieses Vorgehen wurde in Kapitel 7 unter der Annahme der überlegenen Performance des Black-Box-Modells entwickelt. Eine direkte Implikation daraus war, dass es vor allem dann zu großen Differenzen in den Prognosewahrscheinlichkeiten kommt, wenn das Black-Box-Modell die Situation besser wiedergibt als das erklärbare Verfahren. Diese Annahme wird in Tabelle 8.8 bestätigt. Die Performance der GBM ist auf den Beobachtungen mit großer Differenz deutlich überlegen.

8.4.2 *Analyse der Vorhersagedifferenz*

Bevor neue Merkmale anhand der Klassifikationsdifferenz generiert werden, folgt eine Analyse der strukturellen Unterschiede zwischen den Beobachtungen, die zu einer kleinen bzw. großen Differenz führen. Zunächst wird eine Analyse der Merkmalsverteilung durchgeführt, wobei nur die Auffälligkeiten betrachtet werden.

8.4.2.1 *Analyse der Merkmalsverteilungen*

Beim Betrachten der Verteilungen der verschiedenen Merkmale, über die Datenpunkte mit kleiner und großer Differenz hinweg, fallen die Variablen *Relationship*, *Marital Status*, *Capital Gain* und *Capital Loss* besonders auf.

Bei Ersterem steigt der Anteil an Ehemännern (Husband) an den Beobachtungen von ungefähr 40 Prozent bei kleiner Prognosedifferenz auf über 68 Prozent. Beim Feature *Marital Status* lässt sich in Bezug auf die Ausprägung zivil-verheirateter Partner ein ähnlicher Effekt beobachten, steigt deren Anteil von circa 44 Prozent bei kleiner Vorhersagedifferenz auf gut 80 Prozent bei jenen Dateninstanzen, die eine große Abweichung zwischen den Prognosen von Black- und White-Box-Modell verursachen.

Zuletzt die Verteilung der Merkmale *Capital Gain* und *Capital Loss* (Abbildung 8.20). Der Anteil Beobachtungen mit einem *Capital Gain* bzw. einem *Capital Loss* von ungleich 0 ist bei den Dateninstanzen, die zu einer großen Vorhersagedifferenz (Abbildung 8.20 A und C) führen, deutlich erhöht. In Zahlen ausgedrückt steigt der Anteil der Datenpunkte mit einem *Capital Gain* ≥ 0 von ungefähr 7 Prozent bei kleiner Differenz (Abbildung 8.20 B)

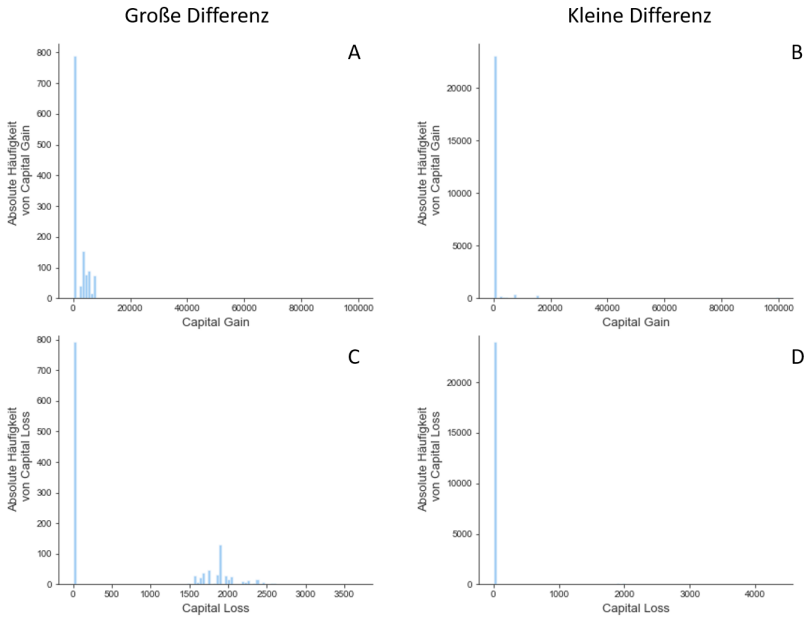


Abbildung 8.20: *Capital Gain/Loss* Merkmalsverteilung bei großer und kleiner Differenz

auf über 37 Prozent (Abbildung 8.20 A). Beim Merkmal *Capital Loss* steigt dieser von 3 auf über 36 Prozent.

Offenbar sind Beobachtungen mit großer Prognosedifferenz deutlich häufiger zivil-verheiratete Ehemänner mit einem vergleichsweise hohen Gewinn bzw. Verlust an den Kapitalmärkten als Instanzen mit kleiner Differenz. Dies sollte sich auch in der globalen Relevanz der Merkmale aus Sicht der beiden ML-Modelle widerspiegeln, die als nächstes betrachtet wird.

8.4.2.2 *Analyse des Merkmaleinflusses*

In Abbildung 8.21 findet sich die Merkmalswichtigkeit der Gradient Boosting Machine für große (links) und kleine (rechts) Vorhersagedifferenzen. Während sich die Feature Importance bei Beobachtungen mit einer kleiner Differenz zwischen Black- und White-Box-Modell stark der bereits in Kapitel 8.3.1 berechneten gleicht, ergibt sich für Instanzen mit großer Differenz ein ganz anderes Bild. Dort haben die Merkmale *Capital Gain* und *Capital Loss* deutlich größeren Einfluss auf die Prognose, was angesichts der stark veränderten Verteilung der beiden Merkmale bereits zu vermuten war.

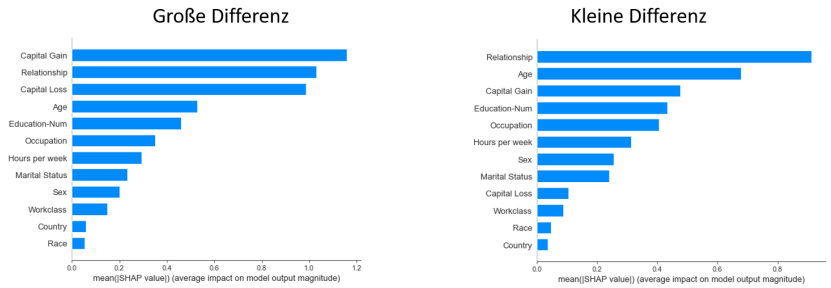


Abbildung 8.21: SHAP Feature Importance GBM für kleine und große Prognosedifferenz

Auch die Merkmalsrelevanz der logistischen Regression kommt zu einem ähnlichen Resultat, was die Veränderung der Wichtigkeit in Bezug auf *Capital Gain* und *Capital Loss* angeht. Auch dort steigt deren Wichtigkeit von der kleinen zur großen Prognosedifferenz deutlich. Die in Abbildung 8.22 dargestellte Merkmalswichtigkeit ist nur approximativ bestimmt. Auf Grund der Transformationen und des One-Hot-Encodings kategorieller Merkmale kann die Wichtigkeit in Relation zu den Merkmalen der GBM nur approximativ wiedergegeben werden. Dazu erfolgt eine Aggregation der SHAP Werte aller Variablen, die durch Transformationen bzw. Encoding eines Merkmals hervorgegangen sind (mit Ausnahme der Interaktionseffekte). Da die Interaktionseffekte aber nur einen kleinen Einfluss auf die Performance des Modells hatten (vgl. Kapitel 8.3.3), stellt dieses Vorgehen eine gute Approximation der Feature Importance innerhalb der logistischen Regression (erweitert) dar.

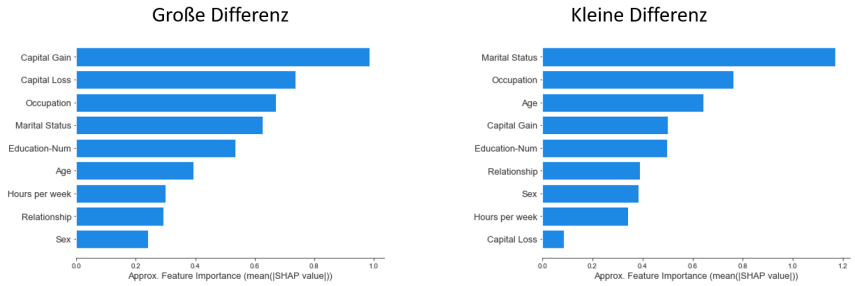


Abbildung 8.22: SHAP Feature Importance der logistischen Regression (erweitert) für kleine und große Prognosedifferenz

Sowohl die GBM als auch das logistische Regressionsmodell kommen auf den Beobachtungen, die zu einer großen Differenz führen, zu einer ähnlichen Einschätzung in Bezug auf die globale Wichtigkeit der Merkmale. Wieso hat dann die logistische Regression eine im Vergleich zur GBM so schlechte Performance auf den Daten mit großer Prognosedifferenz? Dazu muss der *SHAP Summary Plot* des Black-Box-Modells in Abbildung 8.23 betrachtet werden. Dort wird sichtbar, dass der Einfluss von *Capital Gain* und

Capital Loss nicht eindeutig, geschweige denn monoton ist. Selbst durch die in Kapitel 8.3.2 durchgeführte Klassifikation der Merkmale *Capital Gain* und *Capital Loss*, ist die logistische Regression nicht imstande, diesen komplexen Zusammenhang darzustellen.

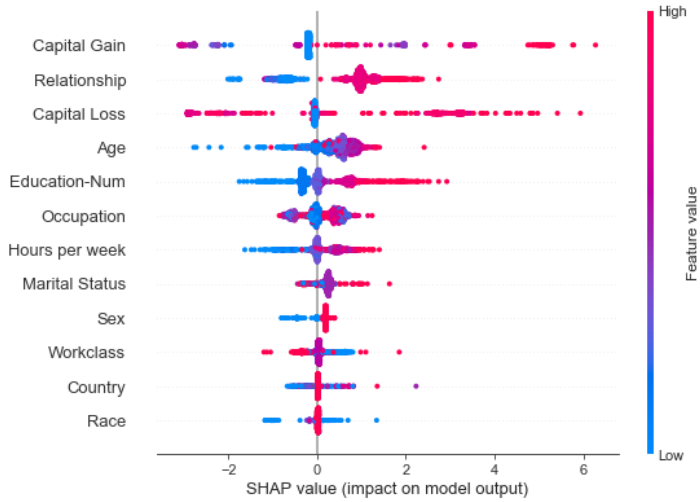


Abbildung 8.23: *SHAP Summary Plot* der GBM für große Prognosedifferenzen

8.4.3 *Merkmalsgenerierung anhand der Vorhersagedifferenz*

Am *SHAP Summary Plots* in Abbildung 8.23 lässt sich bereits ablesen, dass eine logistische Regression nicht fähig sein wird, den Zusammenhang zwischen den wichtigsten Merkmalen aus Sicht der GBM (aber auch der logistischen Regression erweitert) und der Zielgröße darzustellen, da dieser nichtmonoton ist. Eine logistische Regression, die auf 80 Prozent der Beobachtungen mit großer Vorhersagedifferenz trainiert wird, erzielt auf den übrigen 20 Prozent lediglich eine AUC von 0.55 - ist also nur minimal besser als eine rein zufällige Entscheidung über die Klassenzugehörigkeit.

Neben der logistischen Regression steht noch ein weiteres erklärbares ML-Verfahren zur Analyse von Klassifikationsproblemen zur Verfügung - kurze Entscheidungsbäume. Auf Grund ihrer Struktur - binäre Splits eines Merkmals - sollten diese deutlich besser imstande sein, den nichtlinearen, nichtmonotonen Einfluss der Merkmale *Capital Gain* und *Capital Loss* zu beschreiben. Ein Entscheidungsbaum der Tiefe Vier, welcher auf den gleichen 80 Prozent der Beobachtungen mit großer Differenz trainiert wurde wie zuvor die logistische Regression, erzielt eine deutlich bessere AUC von ungefähr 0.86. In Abbildung 8.24 ist dieser Entscheidungsbaum dargestellt. Wie zu erwarten, werden vor allem *Capital Gain* und *Capital Loss* als Splitmerkmale herangezogen.

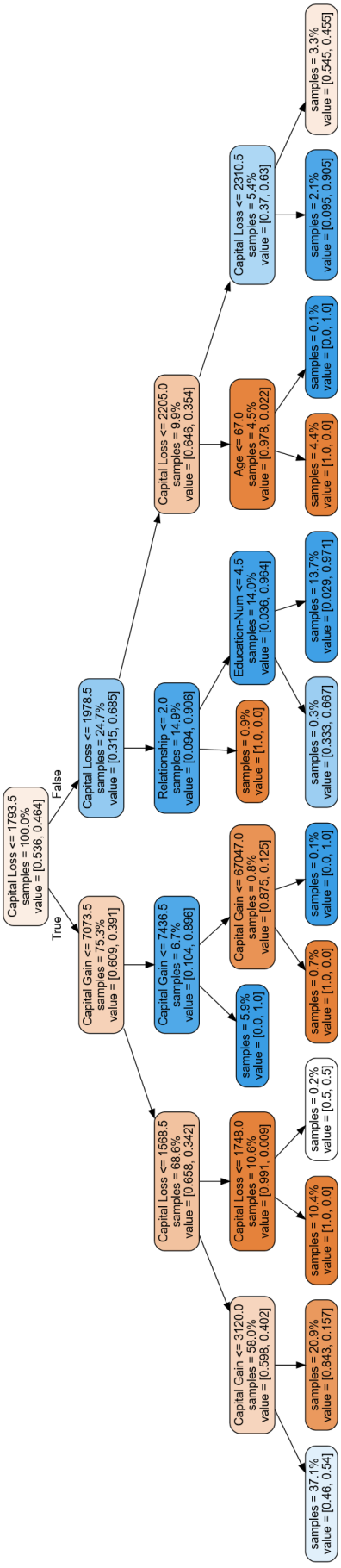


Abbildung 8.24: Entscheidungsbaum auf großer Vorhersagedifferenz

Nun stellt sich die Frage, wie die Regeln des Entscheidungsbaums zu nutzen sind, um die Vorhersagen der logistischen Regression (erweitert) zu verbessern. Dazu wird die folgende Interpretation des Pfades eines Entscheidungsbaums herangezogen: Jeder Pfad eines Baums zu einem Knoten kann als ein neues, komplexes Merkmal angesehen werden. Dabei handelt es sich um ein Dummy-Merkmal, das wiedergibt, ob eine Beobachtung die Eigenschaften des Pfades aufweist oder nicht. Wenn aus dem Entscheidungsbaum in Abbildung 8.24 diejenigen Pfade mit möglichst großen - groß im Sinne von vielen Beobachtungen - und reinen Knoten - rein im Sinne von einer möglichst eindeutigen Zuordnung zu einer Zielklasse - extrahiert werden, ergibt sich ein sehr trennscharfes 0-1-Merkmal. Mittels diesen Vorgehens folgt die Selektion der folgenden fünf Merkmale aus dem Entscheidungsbaum:

- $(\text{Capital Loss} \leq 1793.5) \wedge (\text{Capital Gain} \in (7073.5, 7436.5])$
- $(\text{Capital Gain} \leq 7073.5) \wedge (\text{Capital Loss} \in (1568.5, 1793.5])$
- $(\text{Capital Gain} \leq 1978.5) \wedge (\text{Capital Loss} > 1793.5)$
- $(\text{Capital Gain} \leq 2205.5) \wedge (\text{Capital Loss} > 1978.5)$
- $(\text{Capital Gain} \leq 2568.5) \wedge (\text{Capital Loss} > 2205.5)$

Bei den letzten drei Merkmalen aus dem Entscheidungsbaum ist eine ähnliche Differenz zwischen *Capital Gain* und *Capital Loss* festzustellen. Dieses Phänomen, kombiniert mit der Tatsache, dass entweder ein Gewinn oder ein Verlust an den Kapitalmärkten erzielt werden kann, legt Nahe, dass die beiden Merkmale auch zu einer Variable zusammengefasst werden könnten.

Das Integrieren dieser Merkmale in die logistische Regression (erweitert) - im Weiteren als logistische Regression (erweitert ++) bezeichnet - führt auf dem Datensplit, der zur Berechnung der Vorhersagedifferenz herangezogen wurde, zu einer deutlichen Verbesserung der Performance auf den Beobachtungen mit großer Vorhersagedifferenz. Dabei verschlechtert sich die Trennschärfe auf den verbleibenden Dateninstanzen (mit kleiner Differenz) nur geringfügig, so dass sich die Leistungsfähigkeit auf den Trainingsdaten insgesamt deutlich erhöht. Die Werte finden sich in Tabelle 8.9. :

Modell	**Große Differenz**	**Kleine Differenz**	**Train**
Gradient Boosting M.	0.9794	0.9365	0.9402
Log. Reg. (erweitert)	0.4896	0.9283	0.9159
Log. Reg. (erw.++)	0.7681	0.9280	0.9227
Anzahl Datenpunkte	1254	24794	26048

Tabelle 8.9: Performanceverbesserung der logistischen Regression (erweitert++) durch Feature Engineering mittels Entscheidungsbaum auf großer Vorhersagedifferenz

Auch auf den Testdaten kann eine deutliche Verbesserung der AUC-Werte der logistischen Regression (erweitert++) festgestellt werden. In der folgenden Tabelle 8.10 finden sich nochmals die AUC-Werte für den gleichen zufällig erzeugten 80-20 Train-Test-Split wie zu Beginn des Kapitels:

Modell	Performance (Train)	Performance (Test)
Gradient Boosting M.	0.9402	0.9260
Log. Reg. (erweitert)	0.9159	0.9135
Log. Reg. (erw.++)	0.9225	0.9188

Tabelle 8.10: Performance der logistischen Regression erweitert++

8.4.4 *Evaluation der logistischen Regression (erweitert ++)*

Die Zusammenfassung der AUC-Ergebnisse findet sich in Tabelle 8.11. Dabei handelt es sich erneut um den mittleren AUC-Wert nach einer k=5 Kreuzvalidierung.

Modell	Performance	Standardabweichung
Gradient Boosting Machine	0.928	0.0020
Log. Regression (erweitert)	0.915	0.0029
Log. Regression (erweitert ++)	0.921	0.0026

Tabelle 8.11: Performance der logistischen Regression nach Ende des Feature Engineerings

Das Black-Box-Modell hat die beste Performance. Allerdings ist es durch die Analyse der Vorhersagedifferenz gelungen, die Performance des logistischen Regressionsmodells nochmals deutlich zu verbessern.

Mittels eines statistischen Tests folgt der Vergleich der Performance zwischen logistischer Regression (erweitert) und der logistischen Regression nach Analyse der Vorhersagedifferenz. Für beide Fragen wird erneut die 5x2CV-Methodik (vgl. Kapitel 5.3.4) angewendet. Die Werte der drei Modelle über die fünf Kreuzvalidierungen finden sich in Tabelle A.3 im Anhang.

Log. Regression (erweitert++) vs. Log. Regression (erweitert)
Das Berechnen der verbundenen T-Statistik für die beiden logistischen Regressionsmodelle ergibt einen t-Wert von 51.840, was zu einem p-Wert von $1.9 \cdot 10^{-12}$ führt, so dass die Null-Hypothese einer Differenz von 0 zwischen den beiden Modellen zum 5%-Niveau zu verwerfen ist. Daraus lässt sich schließen, dass das Feature Engineering zu einer statistisch signifikanten Verbesserung der Trennschärfe der logistischen Regression führt.

Gradient Boosting Machine vs. Log. Regression (erweitert++)
Das Berechnen der verbundenen T-Statistik für den Vergleich zwischen Black-

und verbessertem White-Box-Modell ergibt einen t-Wert von -23.5, was zu einem p-Wert von $2.2 \cdot 10^{-9}$ führt, so dass nach wie vor ein zum 5%-Niveau signifikanter Unterschied zwischen der Performance von Black- und White-Box-Modell vorliegt.

8.5 EVALUATION DES FEATURE ENGINEERINGS MITTELS BLACK-BOX-MODELLEN

Nachdem das Vorgehen aus Kapitel 6 und Kapitel 7 durchgeführt wurden, folgt die Evaluation der Ergebnisse. Dabei werden die Ergebnisse auf dem Adult-Datensatz einmal in Bezug auf die Performance evaluiert und daran anschließend die Erklärbarkeit des durch das Feature Engineering entstandenen White-Box-Modells rekapituliert.

8.5.1 *Evaluation aus Sicht der Performance*

In Tabelle 8.12 finden sich die mittleren AUC-Werte und deren Standardabweichung nach einer k=5 Kreuzvalidierung.

Modell	**Performance**	**Standardabweichung**
Log. Regression (Baseline)	0.906	0.0042
Log. Regression (erweitert)	0.915	0.0029
Log. Regression (erweitert ++)	0.921	0.0026
Gradient Boosting Machine	0.928	0.0020

Tabelle 8.12: Performanceentwicklung der logistischen Regression durch das Feature Engineering

In Abbildung 8.25 finden sich die Boxplots eines jeden Modells für eine fünfmal durchgeführte k=5 Kreuzvalidierung mit unterschiedlichen Splits in den einzelnen Kreuzvalidierungssamples. Sowohl in Tabelle 8.12 als auch im Boxplot ist die deutliche Verbesserung der Performance der logistischen Regression durch das Feature Engineering zu erkennen. Allerdings bleibt die Gradient Boosting Machine nach wie vor das performanteste Modell.

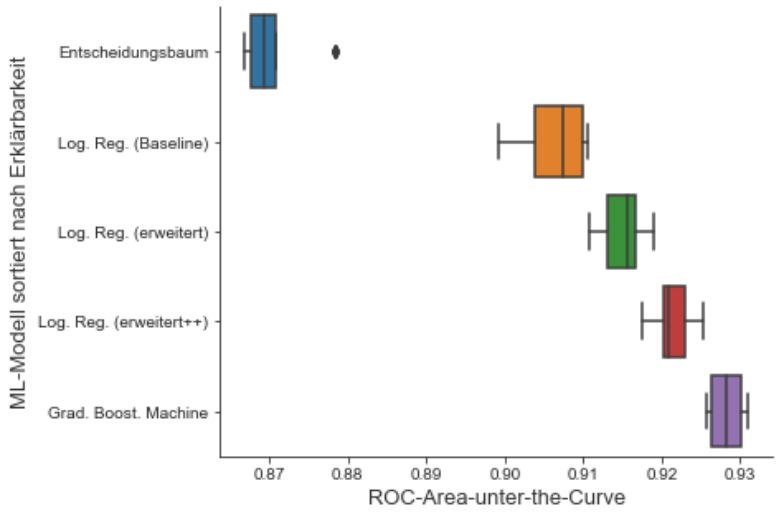


Abbildung 8.25: Boxplot der AUC-Werte über 5 x k=5 CV's

Darüber hinaus lässt sich eine verringerte Breite der Box - und damit eine geringere Streuung der Trennschärfe über die Samples hinweg - durch das Feature Engineering feststellen. Während des Prozesses der Transformation bzw. Generierung von Merkmalen war eine (minimale) Reduzierung der Standardabweichung für die logistische Regression zu beobachten.

Durch das Integrieren von funktionalen Nichtlinearitäten und Wechselwirkungen, sowie den Merkmalen aus dem Entscheidungsbaum erhöht sich anscheinend die Stabilität der Vorhersage über die Kreuzvalidierung hinweg, was auf eine bessere Generalisierung des Modells über verschiedene Beobachtungen schließen lässt. Diese naheliegende Interpretation muss allerdings aus mehreren Gründen relativiert werden:

1. Die Transformation wurde auf Basis der Informationen eines Train-Test-Splits (80-20) bestimmt. Das heißt, sowohl das logistische Regressionsmodell (erweitert) als auch das Modell (erweitert++), wurden mit Informationen über die gleichen 80 Prozent der Beobachtungen engineert. Dies hat Auswirkungen auf die Kreuzvalidierung:
   Es ist dadurch sehr wahrscheinlich, dass jedes Test-Sample der Kreuzvalidierung auch Beobachtungen enthält, anhand derer die Transformationen bestimmt worden sind. Dadurch sind gewissermaßen Information aus den Testdaten in das Modelltraining geleakt, was zwangsläufig zu einer Reduktion der Standardabweichung im Vergleich zu einem Modell ohne Feature Engineering führt.

2. Darüber hinaus sind die Standardabweichungen in diesem Beispiel bei allen Verfahren - egal ob Black- oder White-Box-Modell - eher gering, was die Interpretation ebenso relativiert.

Zusammenfassend lässt sich eine deutliche Steigerung der Leistungsfähigkeit der logistischen Regression feststellen. Mittels der 5x2CV-Methode nach

[Die98] sind statistisch siginifikante Verbesserungen nachweisbar. Ziel der Arbeit ist es aber nicht nur, die Performance intrinsisch erklärbarer Modelle zu steigern, sondern dabei deren intrinsisch erklärbaren Charakter zu erhalten.

8.5.2 *Evaluation aus Sicht der Erklärbarkeit*

Bereits in Kapitel 5.1 wurde auf den Zusammenhang zwischen Transformationen und Erklärbarkeit eines Modells hingewiesen. Der Originalmerkmalsraum ist immer die erklärbarste Repräsentation der Daten. Mit jeder Transformation des Datenraums wird das Modell weniger erklärbar und damit insbesondere für Laien schwerer nachzuvollziehen.
Im Folgenden werden sowohl die Methodik aus Kapitel 6 als auch das Feature Engineering anhand der Prognosedifferenz unter diesem Gesichtspunkt betrachtet.

Entfernen von irrelevanten Merkmalen
Die Entfernung der Merkmale *Country*, *Race* und *Workclass* aus dem Merkmalsraum hat die Interpretierbarkeit des Modells verbessert; denn je weniger Feature ein Modell zu seiner Entscheidung heranzieht, desto leichter lässt sich dieses interpretieren.

Transformationen von Merkmalen
Der Einfluss von Merkmalstransformationen mit dem Ziel, nichtlinearen Einflüssen besser Rechnung zu tragen, hängt stark vom Merkmalstyp und der verwendeten Transformation ab:

- Das Zusammenfassen von Kategorien einer kategoriellen Variable hat eher positiven Einfluss auf die Transparenz eines White-Box-Modells, da sich dadurch die Dimensionalität des Problems reduziert.
- Ähnliches gilt für das Diskretisieren einer kontinuierlichen Variable. Die einfache Kategorisierung im Falle von *Capital Gain* und *Capital Loss* in niedrig, mittel und hoch ist bezüglich der Erklärbarkeit sicherlich mit der des kontinuerlichen Äquivalents vergleichbar.
- Die Interpretierbarkeit des Transformation eines stetigen Merkmals hängt sicherlich stark von der gewählten Transformation ab. Quadratische Zusammenhänge sind deutlich leichter nachvollziehbar, als Polynome vom Grad Fünf oder noch komplexere Transformationen. Daher sollte immer eine möglichst einfache Transformationsabbildung gewählt werden, um die Beeinflussung der Erklärbarkeit zu minimieren.

Graphische Darstellungen der durch die Transformation entstandenen Einflüsse eines Merkmals können das Verständnis deutlich verbessern. In Kapitel 6.2.2 finden sich für alle Merkmalstypen und Transformationen beispielhafte Visualisierungen solcher veränderter Merkmalseinflüsse.

Integration von Wechselwirkungen
Die Integration von Wechselwirkungen ist im Hinblick auf die Interpretierbarkeit der problematischste Schritt des Feature Engineerings mittels Erklärungen. Hierbei bilden immer zwei Merkmale gleichzeitig die Grundlage der Analyse. Die Visualisierungsmöglichkeit mittels der *SHAP Dependence Plots* sind sicherlich hilfreich, um einen möglichst hohen Grad an Transparenz herzustellen, deren Interpretation bleibt aber herausfordernd.

In Falle des Adult-Datensatzes muss angesichts der geringen Verbesserung der Trennschärfe der logistischen Regression die Relevanz der Transformation für die Modellperformance betrachtet werden. Es stellt sich die Frage, ob der Verlust an Erklärbarkeit durch die minimale Steigerung der Performance aufgewogen wird. Diese Entscheidung ist immer abhängig vom Anwendungsfall und obliegt den Entwicklern der Modelle. So lange es keine Möglichkeiten gibt, die Erklärbarkeit eines Modells mit einfachen Mitteln vergleichbar und verlässlich zu messen [Lip18], bleibt dies immer eine subjektive Ermessensentscheidung.

Analyse der Vorhersagedifferenz
Auch das Analysierern der Vorhersagedifferenz zwischen logistischer Regression (erweitert) und der GBM, führte zu einer signifikanten Verbesserung der Performance des White-Box-Modells. Da die logistische Regression (erweitert) Grundlage des resultierenden Modells ist, übertragen sich die gerade getätigten Aussagen, auch auf das Modell (erweitert++).

Die Bedeutung graphischer Visualisierungen bei der Manipulation und Generierung von Merkmalen spielt auch für die anhand des Entscheidungsbaums generierten neuen Merkmale eine große Rolle. Denn auch diese lassen sich graphisch darstellen. Dazu muss lediglich der Entscheidungsbaum visualisiert werden. [VAB07], [Mar+08] oder [RSG16a] konnten in ihren experimentellen Studien zeigen, dass auch Laien imstande sind, die Schlussfolgerungen eines Entscheidungsbaums nachzuvollziehen.

Im Falle des Adult-Datensatzes sind die Entscheidungsregeln des Baums und damit auch die extrahierten Merkmale allerdings wenig intuitiv. Es sund nur schwer rationale Begründungen zu finden, warum ein *Capital Gain* zwischen 7073.5 und 7436.5 die Wahrscheinlichkeit auf ein Einkommen von über 50000 $ deutlich erhöht, ein *Capital Gain* unter 3120 die Wahrscheinlichkeit hingegen deutlich verringert. An dieser Stelle spielt der in Kapitel 6.1 bereits angesprochene *Confirmation Bias* eine große Rolle: Erklärungen bzw. Zusammenhänge, die nicht den menschlichen Vorurteilen entsprechen, bzw. den rationalen Denkmustern widersprechen, werden sehr skeptisch bis ablehnend wahrgenommen oder schlicht ignoriert [Nic98].

9 ERGEBNISSE UND SCHLUSSFOLGERUNGEN

Durch die Anwendung auf den Adult-Datensatz konnte dem Leser der Arbeit ein Eindruck von den Möglichkeiten und Herausforderungen des Feature Engineerings mittels der Erklärungen eines Black-Box-Modells vermittelt werden. In diesem Kapitel werden die wesentlichen Erkenntnisse der Arbeit nochmals zusammengefasst und diskutiert. Daran anschließend folgt ein Ausblick auf weitere offene Fragen, die sich im Rahmen der Arbeit ergeben haben.

9.1 ZUSAMMENFASSUNG

Ziel dieser Arbeit war es aufzuzeigen, dass sich die Performanz von White-Box-Modellen mittels Erklärungen von Black-Box-Modellen verbessern lässt, ohne dabei deren intrinsische Erklärbarkeit zu stark einzuschränken.

Intrinsisch erklärbare Verfahren spielen in der Praxis vieler Branchen nach wie vor eine große Rolle. Insbesondere in regulierten Branchen erfreuen sich lineare bzw. logistische Regressionsmodelle nach wie vor einer großen Beliebtheit, da sie im Einsatz erprobt und die Aufsichtsbehörden mit deren Interpretation vertraut sind. Trotz jahrelanger Erfahrung im Umgang mit diesen Modellen sind die Entwickler häufig nicht imstande die Performance moderner maschineller Lernverfahren aus den Bereichen der Ensemble Modelle oder Neuronalen Netze zu erzielen.

Es existieren mittlerweile zahlreiche Werkzeuge zur Erklärung der Entscheidungen dieser Black-Box-Verfahren, von denen einige in Kapitel 4 erläutert wurden. Jedoch besitzen die Unternehmen keinerlei Erfahrung im Umgang mit diesen Methoden und der Akzeptanz dieser auf Seiten der Aufsichtsbehörden.

Ziel muss es sein, ein möglichst erklärbares Modell mit einer möglichst hohen Vorhersagequalität zu entwickeln. In dieser Arbeit wurden zwei Ansätze präsentiert, die bessere Trennschärfe der modernen, hochgradig komplexen Verfahren mit dem erklärbaren Charakter einer logistischen Regression zu kombinieren. Diese führen ein performance-orientiertes Feature Engineering für White-Box-Modelle durch und greifen dabei auf die Erklärungen eines leistungsfähigeren Black-Box-Ansatzes zurück. Dazu wurden zwei verschiedene Methodiken entwickelt, die unabhängig voneinander, aber auch gemeinsam angewendet werden können. Je nach Methodik und Problem, erhöht sich entweder die Performance des erklärbaren Modells, oder dessen

Interpretierbarkeit. In seltenen Fällen ist sogar beides möglich.

Die erste Methodik bedient sich der strukturellen Unterschiede zwischen White- und Black-Box-Verfahren in Bezug auf die funktionale Form der daraus resultierenden Modelle. In Kapitel 5.3.1 wurden vier wesentliche Unterschiede zwischen erklärbaren und intransparenten ML-Algorithmen dargelegt, die einerseits die oft unterlegene Performance der klassischen Verfahren, gleichzeitig aber auch deren hohe Transparenz begründen: Die Linearität und Monotonie der Verfahren in Kombination mit einer einfachen Merkmalsauswahl, ohne das Berechnen komplexer Interaktionen.
Black-Box-Verfahren enthalten keine dieser vier Restriktionen und sind imstande beliebig komplexe, nichtlineare Zusammenhänge und Interaktionen zu beschreiben. Hinzu kommt eine in diesen Verfahren implizit enthaltene Merkmalsselektion.

Ausgehend von dieser Analyse wurde in Kapitel 6.2 ein Vorgehen entwickelt, das es ermöglicht durch geschicktes Feature Engineering die angesprochenen Einschränkungen zu überwinden. Die drei wesentlichen Ansatzpunkte der Methodik aus Kapitel 6.2 sind:

- Eine geschickte **Merkmalsauswahl**. Die Performance eines ML-Modells ist in hohem Maße von den verwendeten Merkmalen abhängig. Um besonders trennscharfe Variablen zu bestimmen, kamen zum einen die Methoden aus Abschnitt 4.3, aber auch die Aggregation von SHAP Werten (vgl. Kapitel 4.6.4) zum Einsatz. Das in Kapitel 6.2.1 vorgestellte Vorgehen ist zweistufig und besteht aus der automatisierten Berechnung der Relevanz einzelner Merkmale via Gini Wichtigkeit oder SHAP. Anschließend kann die Auswahl der Variablen manuell erfolgen, sowohl unter den Gesichtspunkten der Leistungsfähigkeit als auch der Erklärbarkeit des Modells. Der Vorteil der Regularisierung mittels Erklärungen eines Black-Box-Modells besteht in dem halbautomatischen Vorgehen, was es ermöglicht auch Aspekte in Bezug auf die Interpretierbarkeit des finalen Modells zu berücksichtigen.

- **Nichtlineare Transformationen** und Klassierung. Kein erklärbares Verfahren ist imstande selbst nichtlineare Effekte adäquat zu repräsentieren. Eine manuelle Integration von einfachen nichtlinearen Effekten in erklärbare Modelle ist möglich, aber aufwendig (vgl. Kapitel 5.3.1). Daher wurde in Kapitel 6.2.2 ein Vorgehen entwickelt, das es ermöglicht, durch Black-Box-Modelle detektierte nichtlineare Abbildungen von Input- auf Outputwerte zu approximieren und anschließend in das Modell zu integrieren.

  Dabei kamen abhängig vom Merkmalstyp zwei verschiedene Ansätze zum Einsatz:

  – Im Falle stetiger Merkmale war der erste Ansatz die Zusammenhänge mittels Polynomen vom Grad kleiner-gleich Fünf zu be-

schreiben. Ist der Zusammenhang nicht durch eine solche Funktion beschreibbar, erfolgt eine Klassierung des Merkmals anhand des Einflusses im Black-Box-Modell. Dabei stand die Verbesserung der Performance im Vordergrund. Die Erklärbarkeit wird durch die Verwendung von einfachen Transformationen bzw. Klassierungen gewährleistet.

  - Der Umgang mit kategoriellen Features hingegen zielt eher auf die Verbesserung der Interpretierbarkeit eines Modells ab. Dazu wurde ebenfalls eine Klassierung durchgeführt und Datenbereiche mit ähnlichem Einfluss auf die Vorhersage des Black-Box-Modells zusammengefasst.

- Die **Integration von zweidimensionalen Interaktionen**. Sowohl die lineare als auch die logistische Regression müssen manuell um Informationen über Wechselwirkungen zwischen Merkmalen ergänzt werden. Das in Abschnitt 6.2.3 entwickelte Vorgehen beschreibt eine Möglichkeit, zweidimensionale Interaktionen in ein bestehendes White-Box-Modell zu integrieren. Dazu wurde der Datenraum zunächst anhand eines interagierenden Merkmals partitioniert. Anschließend wurde für jede dieser Partitionen eine Transformation des anderen interagierenden Merkmals durchgeführt. Die Methodik ist sowohl für kategorielle als auch für stetige Merkmale anwendbar.

Die zweite Methodik wählt einen anderen Ansatz, Merkmale anhand des Black-Box-Modells zu generieren. Anstatt das Black-Box-Modell in seiner Gesamtheit zu erklären und daraus Erkenntnisse über sinnvolle Transformationen bzw. neue Feature abzuleiten, betrachtet diese ausschließlich die Beobachtungen, für die das Black-Box-Modell zu einer anderen Einschätzung als das White-Box-Modell gelangt. Dazu wurden in Kapitel 7 zunächst die Konzepte der **Vorhersagedifferenz** und des damit verbundenen Schwellwerts eingeführt.

Anschließend wurden zwei Ansätze betrachtet, um auf Basis der Analyse von Vorhersagedifferenzen neue Merkmale für erklärbare Modelle zu generieren:

1. Der Ansatz aus Kapitel 7.2.1 verfolgt eine Analyse der Daten mit großer Vorhersagedifferenz mittels Erklärungen der verwendeten White- bzw. Black-Box-Modelle. Tauchen dabei neue, bislang nicht detektierte nichtlineare Effekte oder Wechselwirkungen auf, lassen sich diese mittels einer der in Kapitel 6.2.2 bzw. 6.2.3 vorgestellten Methoden in das Modell integrieren. Dabei ist es von besonders großer Bedeutung, die Performance des Modells auf dem gesamten Datensatz im Auge zu behalten.

2. Darüber hinaus besteht die Möglichkeit, auf den Daten mit großer Vorhersagedifferenz zwischen White- und Black-Box-Modell ein neues erklärbares Modell zu trainieren. Dieser Ansatz wurde in Kapitel 7.2.2

vorgestellt und betrachtet das Extrahieren von Merkmalen aus diesem neuen erklärbaren Modell. Diese lassen sich in das ursprüngliche, erklärbare Verfahren integrieren.

In Tabelle 9.1 findet sich eine zusammenfassende Darstellung der vier verschiedenen Ansätze, die zum Feature Engineering herangezogen werden können, in Bezug auf die Performance bzw. Erklärbarkeit des resultierenden Modells.

Methode	Verbesserung der	
	Performance	Erklärbarkeit
Merkmalsrelevanz		✓
Transformation durch Polynome	✓	
Transformation durch Klassierung (stetig)	✓	
Transformation durch Klassierung (kateg.)		✓
Integration von Interaktionen	✓	
Analyse der Klassifikationsdifferenz	✓	

Tabelle 9.1: Auswirkungen der Ansätze des Feature Engineerings

In Kapitel 8 wurde die Anwendung der beiden Methoden anhand des Adult-Datensatzes illustriert und das Vorgehen aus den Perspektiven der Leistungsfähigkeit und Erklärbarkeit evaluiert. Zur Bewertung der Performance wurde die 5x2CV-Prozedur verwendet, wobei bereits bei deren Herleitung in Kapitel 5.3.3 auf die Herausforderungen bzgl. des Vergleichs der Leistungsfähigkeit verschiedener maschineller Lernverfahren eingegangen wurde.
Beide Methoden wurden auf dem Adult-Datensatz kombiniert angewendet und führten zu einer Verbesserung der Trennschärfe (in AUC evaluiert) um 1.5 Prozentpunkte von 0.906 auf 0.921. Die Differenz zwischen den AUC-Werten der logistischen Regression und der Gradient Boosting Machine (0.928) verringerte sich dabei um 68 Prozent.

Die Erklärbarkeit der aus dem Feature Engineering resultierenden logistischen Regression zu bewerten, ist momentan nicht nach objektiven Maßstäben möglich, da in der Forschung bislang keine Möglichkeiten bekannt sind, die Erklärbarkeit maschineller Lernverfahren zu messen [Lip18]. Nichtsdestoweniger wurde das Vorgehen in Kapitel 8.5.2 nochmals unter dem Gesichtspunkt möglichst erklärbare Modelle zu erhalten, rekapituliert.

9.2 DISKUSSION UND AUSBLICK

Ziel der Arbeit war es, zu untersuchen, inwieweit Erklärungen eines performanten, komplexen maschinellen Black-Box-Verfahrens das Feature Engineering für erklärbare Modelle unterstützen können, um performante und

gleichzeitig transparente Modelle zu erhalten. Dazu wurde ein Vorgehen entwickelt, das durch systematische Analyse der SHAP Erklärungen eines Black-Box-Verfahrens die Selektion, Transformation und auch Generierung von Merkmalen für erklärbare Modelle ermöglicht.

In den zahlreichen Publikationen zur Erklärbarkeit maschineller Lernverfahren wird zwar betont, dass sich diese auch zur Verbesserung traditioneller, statistischer Verfahren nutzen lassen - so beispielsweise bei [HG18], [LL17] oder [Wel+16] - Empfehlungen oder Methoden, wie diese Integration der Erkenntnisse aus Black-Box-Modellen erfolgen kann, wenn die funktionale Form des Zusammenhangs bzw. die Interaktion nicht offensichtlich ist, waren bislang allerdings nicht Gegenstand der Literatur.

Das hier präsentierte Vorgehen unterbreitet einen ersten Vorschlag, wie eine solche Integration problemunabhängig und reproduzierbar durchführbar ist. Dabei ermöglicht es nicht nur die Verbesserung der Performance von White-Box-Modellen, sondern hält dem Entwickler stets die Möglichkeit offen, Transformationen auf Grund von Bedenken in Bezug auf der Interpretierbarkeit zu unterlassen. Im Laufe der Arbeit konnte anhand mehrerer Beispiele illustriert werden, dass verschiedene Anpassungen auf Basis eines Black-Box-Modells zu deutlichen Verbesserungen der Performance von White-Box-Modellen führen können.

Das Vorgehen beschreibt eine Kombination der Vorhersagepower moderner maschineller Black-Box-Verfahren mit dem intrinsisch erklärbaren Charakter linearer Modelle. Das Black-Box-Modell und dessen Erklärungen werden genutzt, um besonders trennscharfe Merkmale zu detektieren und für diese bestmögliche Transformationen zu bestimmen. Darüber hinaus werden durch die Verfahren Interaktionen zwischen Merkmalen erkennbar. Das Vorgehen überführt diese Erkenntnisse in Transformationen des Merkmalsraums der erklärbaren Modelle, was zu einem leistungsfähigeren White-Box-Modell führt, das seinen intrinsisch erklärbaren Charakter nicht verliert, so lange die Transformationen nicht zu komplex werden.

Aus Sicht des Praktikers bietet sich dadurch - gerade im regulativen Umfeld - eine dritte Möglichkeit - Patrick Hall bezeichnet dies als *dritten Weg* [HG18] - im Umgang mit Black-Box-Verfahren, neben dem Verzicht auf deren Einsatz oder aber der Akzeptanz der intransparenten Gestalt.

Damit liefert die Methodik zugleich eine Orientierung, wie das Feature Engineering für White-Box-Modelle systematischer gestaltet werden kann. Momentan verbringen die Entwickler solcher Modelle einen Großteil ihrer Zeit mit der Vorbereitung der Datenbasis für das Modelltraining [Cro16]. Das in dieser Arbeit entwickelte Vorgehen kann diesen Prozess durch ein systematisches Feature Engineering unterstützen. Dadurch wird der Aufwand für ein manuelles Engineering, das meist auf Korrelationen, Verteilungen, und im

Kontext der logistischen Regression zusätzlich auf den sogenannten *Weight of Evidenz* [JG60] basiert, deutlich reduziert.

Auch wenn die Methodik das Feature Engineering beschleunigen kann, erfordert das hier präsentierte Vorgehen immer noch einen hohen manuellen Aufwand. Ziel zukünftiger Arbeiten könnte eine teilweise Automatisierung des Vorgehens sein. Die Berechnung der nichtlinearen Transformationspolynome für stetige Merkmale in Kapitel 6.2.2 liefert bereits erste Ideen, wie eine solche automatische Generierung von Feature Engineering Schritten aussehen könnte:

Die Menge der in Frage kommenden Polynome ist durch die Anforderungen an die Erklärbarkeit der Modelle beschränkt. Darüber hinaus besteht mit der Anpassungsgüte R^2 des Polynoms die Möglichkeit der Evaluation einer Transformation. Ein durch den Entwickler zu wählender Parameter bestimmt, welcher Gewinn an Anpassungsgüte notwendig ist, um den Anstieg des Polynomgrades, und damit den Verlust an Erklärbarkeit, zu rechtfertigen.

Zur automatischen Klassierung von stetigen bzw. kategoriellen Merkmalen könnten sich L1-basierte Trend Schätzungen [Kim+09] aus der Analyse von Zeitreihen als nützlich erweisen, ermöglichen diese doch das Schätzen konstanter Trends in den Daten. Im Falle des Einflusses von Merkmalen auf die Vorhersagen eines Black-Box-Modells entspricht ein konstanter Trend in den Daten dann gerade Bereichen mit einem identischen Einfluss auf die Zielgröße.

Das Ziel einer solchen Automatisierung ist eine Liste mit Vorschlägen für mögliche Transformationen anhand der gelernten Zusammenhänge des Black-Box-Modells. So wird dem Entwickler die Möglichkeit gegeben, selbst zu entscheiden, welche Transformationen aus Sicht der zu erhaltenden Erklärbarkeit des Modells vertretbar sind. Dies ist immer eine problemspezifische, schwierige Entscheidung, da bislang keine objektive Bewertung der Interpretierbarkeit eines Modells existiert.

In zukünftigen Arbeiten könnte eine systematische Untersuchung des hier entwickelten Vorgehens, oder späterer Erweiterungen dessen, auf einer Vielzahl von Datensätzen erfolgen. Einzige Einschränkung ist die überlegene Performance des verwendeten Black-Box-Modells gegenüber dem erklärbaren Ansatz. Neben dieser Fokussierung auf verschiedene Datensätze scheint eine Analyse der Methodik auf verschiedenen zugrunde liegenden Black-Box-Modellen vielversprechend, da das entwickelte Vorgehen modell-agnostisch ist. Diese Arbeit fokussiert sich ausschließlich auf Gradient Boosting Machines, da bislang eine effiziente Berechnung der SHAP Erklärungen nur für baumbasierte Ensembles möglich ist.

Von besonderem Interesse ist der Vergleich der Transformationen verschiedener Black-Box-Modelle für ein und dasselbe Merkmal. Unterscheiden sich diese stark, ist dies ein Hinweis darauf, dass der Einfluss eines Merkmals nicht eindeutig ist. Daraus könnten sich Anknüpfungspunkte ergeben, dass Training von Black-Box-Verfahren so zu gestalten, das der Einfluss eines Merkmals auf die Zielgröße wünschenswerte Eigenschaften vorzuweisen hat.

Darüber hinaus existieren zahlreiche modell-spezifische Erklärbarkeitsanssätze - insbesondere für Neuronale Netze - deren Erkenntnisse abhängig vom verwendeten Black-Box-Verfahren Berücksichtigung finden könnten.

Des Weiteren existieren vor allem für lineare und logistische Regressionsverfahren eine Reihe an (aufwendigen) Erweiterungen, mittels derer eine Auswahl von Merkmalen und die Integration von Nichtlinearitäten bzw. Interaktionen durchgeführt werden kann. Ein Vergleich der Transformationen, die sich durch diese Erweiterungen ergeben, mit dem Feature Engineering, das in dieser Arbeit entwickelte wurde, wäre ebenfalls von Interesse.

Teil IV

ANHANG

A

TABELLEN

Log. Regression	Grad. Boost. M.
0.902	0.923
0.907	0.930
0.905	0.925
0.905	0.927
0.901	0.924
0.909	0.931
0.902	0.923
0.908	0.929
0.902	0.929
0.906	0.927
∅ 9.05	∅ 0.927

Tabelle A.1: 2x5CV - AUC Werte der logistischen Regression und GBM

Log. Regression	Log. Regression (erweitert)	Grad. Boost. M.
0.902	0.912	0.923
0.907	0.917	0.930
0.905	0.914	0.925
0.905	0.916	0.927
0.901	0.911	0.924
0.909	0.919	0.931
0.902	0.912	0.923
0.908	0.918	0.929
0.902	0.916	0.929
0.906	0.913	0.927
∅ 9.05	∅ 0.915	∅ 0.927

Tabelle A.2: 2x5CV - AUC Werte der logistischen Regression (erweitert) und GBM

Log. Regression (erweitert++)	Log. Regression (erweitert)	Grad. Boost. M.
0.918	0.912	0.923
0.924	0.917	0.930
0.920	0.914	0.925
0.922	0.916	0.927
0.918	0.911	0.924
0.924	0.919	0.931
0.918	0.912	0.923
0.923	0.918	0.929
0.922	0.916	0.929
0.919	0.913	0.927
∅ 9.20	∅ 0.915	∅ 0.927

Tabelle A.3: 2x5CV - AUC Werte der logistischen Regression (erweitert++) und GBM

Merkmal	**Log. Regression**		**Entscheid'baum**		**Grad. Boost. M.**	
	One-Hot	Label	One-Hot	Label	One-Hot	Label
Baseline	0.906 (0.004)	0.888 (0.005)	0.870 (0.004)	0.871 (0.004)	0.928 (0.002)	0.928 (0.002)
Merkmals-relevanz	0.905 (0.004)	0.887 (0.007)	0.870 (0.004)	0.871 (0.004)	0.927 (0.002)	0.927 (0.002)
Relationship	0.905 (0.003)	0.889 (0.005)	0.870 (0.004)	0.871 (0.004)	0.928 (0.002)	0.927 (0.002)
Age	0.909 (0.003)	0.896 (0.004)	0.870 (0.004)	0.871 (0.004)	0.928 (0.002)	0.927 (0.002)
Capital Gain	0.913 (0.003)	0.899 (0.004)	0.868 (0.004)	0.868 (0.005)	0.921 (0.003)	0.920 (0.002)
Education-Num	0.913 (0.003)	0.899 (0.003)	0.868 (0.004)	0.868 (0.005)	0.921 (0.003)	0.920 (0.002)
Occupation	0.913 (0.003)	0.899 (0.004)	0.868 (0.004)	0.868 (0.005)	0.921 (0.003)	0.920 (0.002)
Hours per week	0.914 (0.003)	0.901 (0.004)	0.868 (0.004)	0.869 (0.005)	0.921 (0.002)	0.920 (0.002)
Marital Status	0.914 (0.003)	0.901 (0.004)	0.868 (0.004)	0.868 (0.005)	0.921 (0.002)	0.920 (0.002)
Capital Loss	0.914 (0.003)	0.901 (0.004)	0.868 (0.004)	0.872 (0.004)	0.915 (0.003)	0.914 (0.003)

Tabelle A.4: Performance nach Integration von Nichtlinearitäten (detailliert)

B

ALGORITHMEN

Data : Menge an N Beobachtungen $x = \{x_n\}$ für $n = 1, \ldots, N$ und ein Ensemble aus M verschiedenen Modellen $y_m(x)$ für $m = 1, \ldots, M$.

1 Initialisiere die Gewichte der Beobachtungen $\{w_n\}$ mit $w_n^{(1)} = \frac{1}{N}$ für $n = 1, \ldots, N$.

2 **for** $m = 1, \ldots, M$ **do**

3 Trainiere einen Klassifizierer $y_m(x)$ für die Trainingsdaten, der die gewichtete Fehlerfunktion

$$J_m = \sum_{n=1}^{N} w_n^{(m)} \cdot \mathcal{I}(y_m(x_n) \neq t_n)$$

minimiert. Dabei sei $\mathcal{I}(y_m(x_n) \neq t_n)$ eine Indikatorfunktion, die 1 annimmt, wenn $y_m(x_n) \neq t_n$ und 0 anderenfalls.

4 Ermittle den gewichteten Anteil der falsch prognostizierten Beobachtungen (vgl. (2.6)):

$$\epsilon_m = \frac{\sum_{n=1}^{N} w_n^{(m)} \cdot \mathcal{I}(y_m(x_n) \neq t_n)}{\sum_{n=1}^{N} w_n^{(m)}}$$

und bestimme damit das Gewicht des m-ten Klassifizierers über

$$\alpha_m = \ln \frac{1 - \epsilon_m}{\epsilon_m}.$$

5 Aktualisiere die Gewichte der Beobachtungen

$$w_n^{m+1} := w_n^m \cdot \exp\left(\alpha_m \cdot \mathcal{I}(y_m(x_n) \neq t_n\right).$$

6 **end**

7 Ermittle die Vorhersage des Ensembles als

$$Y_M(x) = sign\left(\sum_{m=1}^{M} \alpha_m \cdot y_m(x)\right)$$

Algorithmus 3 : Adaptive Boosting Algorithmus

Initialisiere $f_0(x) = \underset{\phi}{\operatorname{argmin}} \sum_{i=1}^{N} L(y_i, \phi)$.

for $m = 1, \ldots, M$ **do**

  for $i = 1, \ldots N$ **do**

    Berechne

$$\tilde{y}_i = -g_m(x_i) = -\left[\frac{\partial L(y, f(x_i))}{\partial f(x_i)}\right]_{f(x_i)=f_{m-1}(x_i)}$$

  end

  Trainiere einen schwachen Lerner $h(x, a)$ derart, dass

$$\phi'_m = \underset{w,\phi}{\operatorname{argmin}} \sum_{i=1}^{N} \left(-g_m(x_i) + w \cdot b(x_i, \phi)\right)^2$$

  und berechne anschießend

$$\rho_m = \underset{\rho}{\operatorname{argmin}} \sum_{i=1}^{N} L\left(y_i, f_{m-1}(x_i) + \rho \cdot b(x_i, \phi')\right)$$

  Setze $f_m(x) = f_{m-1}(x) + \rho_m \cdot b(x; \phi_m)$

end

Ermittle die Vorhersage des Ensembles als

$$F_M(x) = f_M(x)$$

Algorithmus 4 : Gradient Boosting Algorithmus

LITERATUR

[AJ18] David Alvarez-Melis und Tommi S. Jaakkola. "On the Robustness of Interpretability Methods". In: *Workshop on Human Interpretability in Machine Learning (@ICML 2018).* (2018).

[Bre04] L Breiman. "Technical Report 670". In: *Statistics Department University of California, Berkeley* (2004).

[Bre+84] L. Breiman, J. H. Friedman, R. A. Olshen und C. J. Stone. *Classification and Regression Trees*. Monterey, CA: Wadsworth und Brooks, 1984.

[Bre96] Leo Breiman. "Stacked regressions". In: *Machine Learning* 24.1 (1996), S. 49–64.

[Bre01] Leo Breiman. "Random Forests". In: *Mach. Learn.* 45.1 (Okt. 2001), S. 5–32. ISSN: 0885-6125.

[CGT09] Javier Castro, Daniel Gomez und Juan Tejada. "Polynomial calculation of the Shapley value based on sampling". In: *Computers & Operations Research* 36 (Mai 2009), S. 1726–1730.

[Cro16] CrowdFlower. "Data Science Report 2016". In: (2016). URL: https://visit.figure-eight.com/rs/416-ZBE-142/images/CrowdFlower_DataScienceReport_2016.pdf (besucht am 28. 08. 2019).

[Die98] Thomas G. Dietterich. "Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms". In: *Neural Computation* 10.7 (1998), S. 1895–1923.

[Dom12] Pedro Domingos. "A Few Useful Things to Know About Machine Learning". In: *Commun. ACM* 55.10 (Okt. 2012), S. 78–87. ISSN: 0001-0782.

[DG17] Dheeru Dua und Casey Graff. *UCI Machine Learning Repository*. 2017. URL: http://archive.ics.uci.edu/ml.

[Dö+18] Inga Döbel, Miriam Leis, Manuel Vogelsang und Henning Petzka. *Maschinelles Lernen - eine Analyse zu Kompetenzen, Forschung und Anwendung*. Fraunhofer Gesellschaft, 2018.

[Est+17] Andre Esteva, Brett Kuprel, Roberto Novoa, Justin Ko, Susan M Swetter, Helen M Blau und Sebastian Thrun. "Dermatologist-level classification of skin cancer with deep neural networks". In: *Nature* 542 (Jan. 2017).

[FK92] U. Faigle und W. Kern. "The Shapley value for cooperative games under precedence constraints". In: *International Journal of Game Theory* 21.3 (1992), S. 249–266.

[Fis25] R.A. Fisher. *Statistical methods for research workers*. Edinburgh Oliver & Boyd, 1925.

[Fri01] Jerome H. Friedman. "Greedy function approximation: A gradient boosting machine." In: *Ann. Statist.* 29.5 (Okt. 2001), S. 1189–1232.

[Fri02] Jerome H. Friedman. "Stochastic gradient boosting". In: *Computational Statistics & Data Analysis* 38.4 (2002). Nonlinear Methods and Data Mining, S. 367 –378. ISSN: 0167-9473. URL: http://www.sciencedirect.com/science/article/pii/S0167947301000652.

[FKM06] Katsushige Fujimoto, Ivan Kojadinovic und Jean-Luc Marichal. "Axiomatic characterizations of probabilistic and cardinal-probabilistic interaction indices". In: *Games and Economic Behavior* 55 (Feb. 2006), S. 72–99.

[Gil+18] Leilani H. Gilpin, David Bau, Ben Z. Yuan, Ayesha Bajwa, Michael Specter und Lalana Kagal. "Explaining Explanations: An Approach to Evaluating Interpretability of Machine Learning". In: *The 5th IEEE International Conference on Data Science and Advanced Analytics* (2018). arXiv: 1806.00069. URL: http://arxiv.org/abs/1806.00069.

[Gla18] Shirin Glander. ""Künstliche Intelligenz und Erklärbarkeit". In: *Informatik Aktuell* (2018). URL: https://www.informatik-aktuell.de/betrieb/kuenstliche-intelligenz/kuenstliche-intelligenz-und-erklaerbarkeit.html#c24715 (besucht am 29. 06. 2019).

[Gol+13] Alex Goldstein, Adam Kapelner, Justin Bleich und Emil Pitkin. "Peeking Inside the Black Box: Visualizing Statistical Learning With Plots of Individual Conditional Expectation". In: *Journal of Computational and Graphical Statistics* 24 (Sep. 2013).

[GF17] Bryce Goodman und Seth Flaxman. "European Union regulations on algorithmic decision-making and a "right to explanation"". In: *AI Magazine* 38 (2017). URL: https://arxiv.org/abs/1606.08813.

[HG18] Patrick Hall und Navdeep Gill. *An Introduction to Machine Learning Interpretability-Dataiku Version*. O'Reilly Media, Incorporated, 2018.

[HTF09] Trevor Hastie, Robert Tibshirani und Jerome Friedman. *The elements of statistical learning: data mining, inference and prediction*. 2. Aufl. Springer, 2009. URL: http://www-stat.stanford.edu/~tibs/ElemStatLearn/.

[He+15] Kaiming He, Xiangyu Zhang, Shaoqing Ren und Jian Sun. "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification". In: *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*. ICCV '15. Washington, DC, USA: IEEE Computer Society, 2015, S. 1026–1034. ISBN: 978-1-4673-8391-2.

[IMW18] Institut für Management-& Wirtschaftsforschung IMWF. "Künstliche Intelligenz am Arbeitsplatz 2018". In: (2018).

[JG60] I J. Good. "Weight of Evidence, Corroboration, Explanatory Power, Information and the Utility of Experiments". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 22 (Juli 1960), S. 319–331.

[Jam+14] Gareth James, Daniela Witten, Trevor Hastie und Robert Tibshirani. *An Introduction to Statistical Learning: With Applications in R*. Springer Publishing Company, Incorporated, 2014. ISBN: 1461471370, 9781461471370.

[JAK16] Surya Mattu Julia Angwin Jeff Larson und Lauren Kirchner. "Machine Bias: There's software used across the country to predict future criminals. And it's biased against blacks." In: *ProPublica* (Mai 2016). URL: https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing (besucht am 28. 06. 2019).

[KRP11] Shachar Kaufman, Saharon Rosset und Claudia Perlich. "Leakage in Data Mining: Formulation, Detection, and Avoidance". In: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* 6 (Jan. 2011), S. 556–563.

[Kim+09] S. Kim, K. Koh, S. Boyd und D. Gorinevsky. "L-1 Trend Filtering". In: *SIAM Review* 51.2 (2009), S. 339–360.

[Koh97] Ron Kohavi. "Scaling Up the Accuracy of Naive-Bayes Classifiers: a Decision-Tree Hybrid". In: *KDD* (Sep. 1997).

[Kul+15] Anthony Kulesa, Martin Krzywinski, Paul Blainey und Naomi S. Altman. "Points of Significance: Sampling distributions and the bootstrap". English (US). In: *Nature Methods* 12.6 (Mai 2015), S. 477–478. ISSN: 1548-7091.

[LPH07] Mark J. van der Laan, Eric C. Polley und Alan E. Hubbard. "Super learner." In: *Statistical applications in genetics and molecular biology* 6 (2007), Article25.

[Lip18] Zachary C. Lipton. "The Mythos of Model Interpretability". In: *Queue* 16.3 (Juni 2018), 30:31–30:57. ISSN: 1542-7730.

[LEL18] Scott M Lundberg, Gabriel G Erion und Su-In Lee. "Consistent individualized feature attribution for tree ensembles". In: *arXiv preprint arXiv:1802.03888* (2018).

[LL17] Scott M Lundberg und Su-In Lee. "A Unified Approach to Interpreting Model Predictions". In: *Advances in Neural Information Processing Systems 30*. Hrsg. von I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan und R. Garnett. Curran Associates, Inc., 2017, S. 4765–4774. URL: http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf.

[Lun+18] Scott M Lundberg, Bala Nair, Monica S Vavilala, Mayumi Horibe, Michael J Eisses, Trevor Adams, David E Liston, Daniel King-Wai Low, Shu-Fang Newman, Jerry Kim u. a. "Explainable machine-learning predictions for the prevention of hypoxaemia during surgery". In: *Nature Biomedical Engineering* 2.10 (2018), S. 749.

[Mar+08] David Martens, Johan Huysmans, Rudy Setiono, Jan Vanthienen und Bart Baesens. "Rule Extraction from Support Vector Machines: An Overview of Issues and Application in Credit Scoring". In: *Rule Extraction from Support Vector Machines*. Hrsg. von Joachim Diederich. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, S. 33–63.

[Mar+11] David Martens, Jan Vanthienen, Wouter Verbeke und Bart Baesens. "Performance of classification models from a user perspective". In: *Decision Support Systems* 51.4 (2011). Recent Advances in Data, Text, and Media Mining & Information Issues in Supply Chain and in Service System Design, S. 782 –793.

[McN47] Quinn McNemar. "Note on the sampling error of the difference between correlated proportions or percentages". In: *Psychometrika* 12.2 (1947), S. 153–157.

[Mil17] Tim Miller. "Explanation in Artificial Intelligence: Insights from the Social Sciences". In: *CoRR* abs/1706.07269 (2017). arXiv: 1706.07269. URL: http://arxiv.org/abs/1706.07269.

[Molsu] Christoph Molnar. *Interpretable Machine Learning. A Guide for Making Black Box Models Explainable*. https://christophm.github.io/interpretable-ml-book/. besucht am 13.09.2019.

[Mon06] Douglas C. Montgomery. *Design and Analysis of Experiments*. USA: John Wiley & Sons, Inc., 2006. ISBN: 0470088109.

[Nic98] Raymond Nickerson. "Confirmation Bias: A Ubiquitous Phenomenon in Many Guises". In: *Review of General Psychology* 2 (Juni 1998), S. 175–220.

[NW00] J. Nocedal und S. Wright. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer New York, 2000. ISBN: 9780387987934.

[O'N16] Cathy O'Neil. *Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy*. New York, NY, USA: Crown Publishing Group, 2016. ISBN: 0553418815, 9780553418811.

[Onl19] Heise Online. "OpenAI Five: Die KI, die den Dota-2-Weltmeister besiegt hat". In: (2019). URL: https://www.heise.de/newsticker/meldung/OpenAI-Five-Die-KI-die-den-Dota-2-Weltmeister-besiegt-hat-4400773.html (besucht am 08. 08. 2019).

[Pia18] Gregory Piatetsky. *Will GDPR Make Machine Learning Illegal?* 2018. URL: https://www.kdnuggets.com/2018/03/gdpr-machine-learning-illegal.html (besucht am 05. 08. 2019).

[Pow07] David Powers. "Evaluation: From precision, recall and fmeasure to roc, informedness, markedness and correlation". In: *Journal of Machine Learning Technologies* 2 (Jan. 2007), S. 37–63.

[RSG16a] Marco Túlio Ribeiro, Sameer Singh und Carlos Guestrin. "Model-Agnostic Interpretability of Machine Learning". In: *ICML Workshop on Human Interpretability in Machine Learning* (2016).

[RSG16b] Marco Tulio Ribeiro, Sameer Singh und Carlos Guestrin. ""Why Should I Trust You?": Explaining the Predictions of Any Classifier". In: *Knowledge Discovery and Data Mining (KDD)* (2016).

[RSB18] Marko Robnik-Sikonja und Marko Bohanec. "Perturbation-Based Explanations of Prediction Models". In: *Human and Machine Learning* (2018), S. 159–175.

[Sal94] Steven L. Salzberg. "C4.5: Programs for Machine Learning by J. Ross Quinlan. Morgan Kaufmann Publishers, Inc., 1993". In: *Machine Learning* 16.3 (1994), S. 235–240. ISSN: 1573-0565.

[Sha53] Lloyd S. Shapley. "A value for n-person games". In: *Contributions to the Theory of Games* 2 (1953), S. 307–317.

[SGK17] Avanti Shrikumar, Peyton Greenside und Anshul Kundaje. "Learning Important Features Through Propagating Activation Differences". In: *International Conference on Machine Learning*. Proceedings of Machine Learning Research 70 (2017). Hrsg. von Doina Precup und Yee Whye Teh, S. 3145–3153. URL: http://proceedings.mlr.press/v70/shrikumar17a.html.

[Sil+17] David Silver u. a. "Mastering the game of Go without human knowledge". In: *Nature* 550 (Okt. 2017), S. 354–359.

[Str+08] Carolin Strobl, Anne-Laure Boulesteix, Thomas Kneib, Thomas Augustin und Achim Zeileis. "Conditional variable importance for random forests". In: *BMC Bioinformatics* 9.1 (2008), S. 307. ISSN: 1471-2105. URL: http://www.biomedcentral.com.proxy.lib.uiowa.edu/1471-2105/9/307/abstract (besucht am 28. 06. 2013).

[ŠK11] Erik Štrumbelj und Igor Kononenko. "A General Method for Visualizing and Explaining Black-Box Regression Models". In: *Adaptive and Natural Computing Algorithms*. Hrsg. von Andrej Dobnikar, Uroš Lotrič und Branko Šter. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, S. 21–30. ISBN: 978-3-642-20267-4.

[ŠK13] Erik Štrumbelj und Igor Kononenko. "Explaining prediction models and individual predictions with feature contributions". In: *Knowledge and Information Systems* 41 (Dez. 2013), S. 647–665.

[ŠK14] Erik Štrumbelj und Igor Kononenko. "Explaining prediction models and individual predictions with feature contributions". In: *Knowledge and Information Systems* 41.3 (2014), S. 647–665.

[VAB07] Anneleen Van Assche und Hendrik Blockeel. "Seeing the Forest Through the Trees: Learning a Comprehensible Model from an Ensemble". In: *Machine Learning: ECML 2007*. Hrsg. von Joost N. Kok, Jacek Koronacki, Raomon Lopez de Mantaras, Stan Matwin, Dunja Mladenič und Andrzej Skowron. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, S. 418–429. ISBN: 978-3-540-74958-5.

[Wel+16] Søren Welling, Hanne Refsgaard, Per Brockhoff und Line Clemmensen. "Forest Floor Visualizations of Random Forests". In: *arXiv:1605.09196* (Mai 2016).

[Wil92] Frank Wilcoxon. "Individual Comparisons by Ranking Methods". In: *Breakthroughs in Statistics: Methodology and Distribution*. Hrsg. von Samuel Kotz und Norman L. Johnson. New York, NY: Springer New York, 1992, S. 196–202.

[You85] Peyton Young. "Monotonic solutions of cooperative games". In: *International Journal of Game Theory* 14.2 (1985), S. 65–72.

[ZM12] Cha Zhang und Yunqian Ma. *Ensemble Machine Learning: Methods and Applications*. Springer Publishing Company, Incorporated, 2012. ISBN: 1441993258, 9781441993250.

[ZH19] Qingyuan Zhao und Trevor Hastie. "Causal Interpretations of Black-Box Models". In: *Journal of Business & Economic Statistics* (Juni 2019), S. 1–19.

[Zho12] Zhi-Hua Zhou. *Ensemble Methods: Foundations and Algorithms*. 1st. Chapman Hall/CRC, 2012. ISBN: 1439830037, 9781439830031.