Universität des Saarlandes

Wissenschaftliche Arbeit im Rahmen des Studiums für das Lehramt Sekundarstufe I+II im Fach Informatik in der Fachrichtung Informatik der Fakultät MI Mathematik und Informatik

Entscheidungsbäume im Maschinellen Lernen - Anwendung auf Studierendendaten und Entwurf eines Unterrichtsmoduls für SchülerInnen

Eingereicht von:

Franz Walgenbach

Eingereicht am:

02.01.2020

Erstgutachterin:

Univ.-Prof. Dr. Verena Wolf

Zweitgutachterin:

Univ.-Prof. Dr. Vera Demberg

Eidesstattliche Erklärung

Hiermit versichere ich, dass ich die vorliegende Wissenschaftliche Arbeit selbstständig verfasst und keine anderen als die angegebenen Hilfsmittel benutzt habe. Die Stellen der Arbeit, die anderen Werken dem Wortlaut oder dem Sinn nach entnommen sind, sind unter Angabe der Quellen als Entlehnung kenntlich gemacht. Bei Zeichnungen, Skizzen oder Plänen sowie bildlichen und grafischen Darstellungen ist angegeben, wenn sie nach eigenen Angaben durch andere ausgeführt oder übernommen worden sind. Sollte ich Teile dieser Arbeit bereits für andere Prüfungen eingereicht haben, habe ich dies ebenfalls kenntlich gemacht. Die eingereichte elektronische Version der Arbeit stimmt mit der vorliegenden schriftlichen überein.

Saarbrücken, 02.01.2020

Franz Walgenbach

Abstract

Ein großer Teil neuer Technik kommt heute nicht mehr ohne den Einsatz Künstlicher Intelligenz aus. Umso wichtiger ist es für die digitale Mündigkeit der Schülerinnen und Schüler, sich bereits in der Schule als Teil der heutigen Allgemeinbildung mit Künstlicher Intelligenz (KI) auseinanderzusetzen und ihre Grundprinzipien, Chancen und Risiken zu verstehen.
Neben einer Anwendung von Techniken des Maschinellen Lernens auf Studierendendaten aus den Informatikstudiengängen der Universität des Saarlandes, stellt die Konzeption eines Unterrichtsmoduls für Schülerinnen und Schüler der Oberstufe den Hauptteil dieser Arbeit dar. Das beschriebene Modul soll Ideen für eine Unterrichtsreihe geben, welche das bisher meist nicht beachtete Thema der Künstlichen Intelligenz in den schulischen Informatikunterricht integriert. Mit Hilfe von Entscheidungsbäumen als Teil des Maschinellen Lernens wird eine Einführung in dieses komplexe Teilgebiet der Künstlichen Intelligenz vermittelt.

Danksagung

Einen großen Dank möchte ich Prof. Verena Wolf dafür aussprechen, mir die Möglichkeit geboten zu haben in der Informatik ein Thema mit fachdidaktischem Schwerpunkt zu bearbeiten. Sowohl bei ihr als auch bei Prof. Vera Demberg bedanke ich mich für die Bereitschaft, die Gutachten für die vorliegende Arbeit zu erstellen.

Des Weiteren gilt mein Dank Gerrit Großmann für die Hilfe beim Studienerfolgsprojekt, sowie Lukas Wachter und Pascal Schmidt für ihre Anregungen zum Unterrichtsmodul.

Außerdem möchte ich mich bei meiner Familie und meinen Freunden Timo Gros, Christina Eimer und Sebastian Wack bedanken, die diese Arbeit Korrektur gelesen haben.

Abschließend danke ich meiner Familie für die großartige Unterstützung während meines Studiums. Ohne Eure Hilfe wäre diese Arbeit nicht möglich gewesen. Danke für Eure Geduld.

Inhaltsverzeichnis

1. Einleitung und Motivation **1**

1.1. Allgemeinbildung 1
1.2. Kompetenzmodell der Gesellschaft für Informatik e.V. 5
1.3. Aufbau der Arbeit 7

2. Grundlagen **9**

2.1. Was ist Maschinelles Lernen? 9
2.2. Arten Maschinellen Lernens 10
2.2.1. Überwachtes vs. unüberwachtes vs. bestärkendes Lernen 10
2.2.2. Klassifikation vs. Regression 11
2.3. Bewertung der Qualität eines Modells 11
2.3.1. Klassifikation 12
2.3.2. Regression 16
2.4. Mögliche Probleme und ihre Lösungen 17
2.4.1. Bias vs. Variance 17
2.4.2. Overfitting 18
2.4.3. Underfitting 19
2.5. Verfahren des überwachten Lernens 21
2.5.1. Entscheidungsbäume 21
2.5.2. Ensemble-Methoden 29
2.6. Attributauswahl 30
2.7. Pruning 31
2.7.1. Reduced Error Pruning 32
2.7.2. Critical Value Pruning 32

3. Anwendung auf Studierendendaten **33**

3.1. Fragestellung und Zielsetzung 33
3.2. Analyse der Daten 34
3.2.1. Struktur der Daten 34
3.2.2. Korrelationsanalyse 35
3.2.3. Ergebnisse 38
3.3. Einsatz Maschinellen Lernens 43
3.3.1. Basisbäume 44

3.3.2. Feature Engineering . . . 45
3.3.3. Anpassen der Hyperparameter . . . 50
3.3.4. Pruning . . . 54
3.3.5. Confidence . . . 62
3.3.6. Künstliche Daten . . . 64
3.4. Zusammenfassung . . . 68

4. Bestehende Angebote zu Maschinellem Lernen für Schülerinnen und Schüler 71
4.1. Schulische Entwicklung . . . 71
4.2. Schülerlabore . . . 71

5. Beschreibung des Unterrichtsmoduls 73
5.1. Zielgruppe . . . 73
5.2. Lernziele . . . 73
5.3. Ablauf . . . 74
5.4. Webseite . . . 79

6. Didaktische Begründung 83
6.1. Aufbau . . . 83
6.2. Arbeitsformen . . . 85
6.3. Aufgaben . . . 86

7. Auswertung des Moduls 87
7.1. Feedback der Schülerinnen und Schüler . . . 87
7.2. Reflektion . . . 88

8. Zusammenfassung 89
8.1. Ausblick . . . 90

Anhang 101
B. Codebook zur Befragung . . . 102
C. CV-Scores der Recursive Feature Elimination . . . 107
C.1. Szenario 1 . . . 107
C.2. Szenario 1a . . . 107
C.3. Szenario 2 . . . 108
C.4. Szenario 3 . . . 109
C.5. Szenario 4 . . . 110
C.6. Szenario 5 . . . 111
C.7. Szenario 6 . . . 112

C.8. Szenario 7 . . . 112
D. gewählte Hyperparameter . . . 113
D.1. Szenario 1 . . . 113
D.2. Szenario 1a . . . 113
D.3. Szenario 2 . . . 114
D.4. Szenario 3 . . . 114
D.5. Szenario 4 . . . 115
D.6. Szenario 5 . . . 115
D.7. Szenario 6 . . . 116
D.8. Szenario 7 . . . 116
E. Notenverteilungen . . . 117
F. Ergebnisse Künstliche Daten . . . 118
G. Verlaufsplan . . . 120
H. Fragebogen zu Vorstellungen der Studierenden . . . 123
I. Arbeitsblätter . . . 125
I.1. Arbeitsblatt Trainings- und Testdaten . . . 125
I.2. Arbeitsblatt Decision Trees . . . 127
I.3. Arbeitsblatt Hyperparameter . . . 129
I.4. Arbeitsblatt Brustkrebsdiagnose . . . 133
J. Mögliche Entscheidungsbäume . . . 134
J.1. Entscheidungsbaum zu Beispiel 5.1 . . . 134
J.2. Entscheidungsbaum zu Arbeitsblatt I.2 a) Version A . . . 134
J.3. Entscheidungsbaum zu Arbeitsblatt I.2 a) Version B . . . 134
K. Sonstige Materialien . . . 135
K.1. Beispiel Decision Tree . . . 135

1. Einleitung und Motivation

Beispiele für Maschinelles Lernen lassen sich in vielen Bereichen finden. Von bekannten und prestigeträchtigen Projekten wie autonom fahrenden Autos, über die Analyse medizinischer Daten inklusive personalisierter Diagnosen und Therapievorschläge, sowie die Automatisierung in der Logistik bis hin zu alltäglichen Dingen wie die Bild-, Handschrift- und Spracherkennung digitaler Assistenten in unseren Smartphones, findet man in nahezu allen Lebensbereichen Einsatzgebiete Künstlicher Intelligenz. So gut wie jeder ist schon mit einem Teilbereich der KI in Kontakt getreten.
Um dieser allgegenwärtigen Präsenz des Maschinellen Lernens gerecht zu werden, ist es wichtig die grundlegenden Konzepte in der Schule zu vermitteln. Im Widerspruch hierzu findet sich das Thema des Maschinellen Lernens in den Lehrplänen des Informatikunterrichts im Saarland [LPG10, LPN19] und den Anforderungen an die Informatikabiturprüfungen der KMK [Kul04] jedoch nicht wieder.

1.1. Allgemeinbildung

Diese Diskrepanz zwischen den vorgesehenen Inhalten des Schulfachs Informatik und der Realität im Bereich der Künstlichen Intelligenz wirft die Frage auf, inwiefern grundlegendes Wissen zu Künstlicher Intelligenz und im Speziellen Grundkenntnisse zu Maschinellem Lernen zur Allgemeinbildung der Schülerinnen und Schüler gehört.
Um dieser Frage nachzugehen soll im Folgenden kurz die Theorie zur Allgemeinbildung von Heymann dargestellt werden. Heymann [Hey96] hat sieben Aufgaben beschrieben, die die Allgemeinbildung in der Schule unabhängig vom Fach zu erfüllen hat. Er nennt hier:

- *Lebensvorbereitung*: Die Schülerinnen und Schüler sollen auf das Leben als Erwachsene vorbereitet werden. Daher soll durch die Schule die „lebenspraktische Nützlichkeit“ [Hey96, S. 51] des Lernstoffs gewährleistet sein. Die Gefahr dieses Ansatzes ist jedoch, eine zu enge Lebensvorbereitung zu erzielen, bei der lediglich konkrete, verkürzende Kenntnisse und Fertigkeiten vermittelt werden. Daher sieht Heymann auch die „Lebensvorbereitung im weiteren Sinne“ [Hey96, S. 60] als es-

senziell an. Hierbei soll frei von jeglichem praktischem Zweck Bildung verwirklicht werden. Wird einzig dieser Ansatz verfolgt, kann die hieraus abgeleitete Bildung jedoch weltfremd werden. Dementsprechend befürwortet Heymann prinzipiell eine ausgewogene Balance der beiden Ansätze zur Lebensvorbereitung. Durch die folgenden Aufgaben der Allgemeinbildung werden bereits viele Aspekte der Lebensvorbereitung im weiteren Sinne aufgegriffen, weshalb sich Heymann auf die Lebensvorbereitung im engeren Sinne der Nützlichkeit beschränkt.

- *Stiftung kultureller Kohärenz*: Schule soll zum einen zur Tradierung der bestehenden Kultur beitragen und somit kulturelle Kontinuität erzeugen und zum anderen die unterschiedlichen und parallel existierenden Teilkulturen, beispielsweise die Kulturen verschiedener Generationen, vereinen und somit kulturelle Kohärenz erzielen. Die kulturelle Kontinuität darf dabei nicht durch einseitige Konzentration auf die Vergangenheit bestimmt sein, sondern muss im Einklang mit dem gesellschaftlichen Fortschritt stehen.

- *Weltorientierung*: Schule soll den Lernenden materiales Wissen über die Welt vermitteln und ihnen einen größeren Erfahrungshorizont ermöglichen. Neben der Wissenschaftsorientierung werden durch die Weltorientierung auch die „lebensweltlichen und nicht primär wissenschaftlich repräsentierbaren Akte des Weltverstehens“ [Hey96, S. 80] impliziert. Als wichtig erachtet Heymann die „fachliche Entgrenzung“ [Hey96, S. 84], also das Darstellen der Bedeutung der vermittelten fachlichen Inhalte für außerfachliche Themen.

- *Anleitung zum kritischen Vernunftgebrauch*: Zum kritischen Vernunftgebrauch gehören die Mündigkeit, Emanzipation und Aufklärung der Schülerinnen und Schüler. Sie sind bereits im Grunde zur vernünftigen Selbstbestimmung fähig, müssen jedoch durch die Bildung hierin unterstützt und angeleitet werden. In der Schule kann dies beispielsweise durch die Auseinandersetzung mit dem formalen Aspekt der Wissenschaftsorientierung, der Neugier nach wissenschaftlicher Erkenntnis, erreicht werden.

- *Entfaltung von Verantwortungsbereitschaft*: Durch diesen Aspekt wird der sozialethische Auftrag an die Allgemeinbildung verdeutlicht. Es soll das verantwortliche Handeln im begrenzteren Feld des schulischen Miteinanders gelernt werden. Auch hinsichtlich des eigenen Lernfortschritts gilt es in der Schule eigenverantwortlich zu handeln. Durch die Forderung nach verantwortlichem Denken zieht Heymann

eine Verbindung zum Aspekt der Weltorientierung und löst sich von den Grenzen der Verantwortungsbereitschaft im schulischen Umfeld.

- *Einübung in Verständigung und Kooperation*: Verständigung und Kooperation sind für Heymann die beiden sozialen Umgangsformen, die für „eine zeitgemäße Allgemeinbildung in einem demokratischen Gemeinwesen unverzichtbar“ [Hey96, S. 110] sind. Verständigung meint dabei sowohl die Einsicht in das Denken der Anderen, als auch das Kommunizieren der eigenen Vorstellungen und ist Voraussetzung für eine gelungene Kooperation. Kooperatives Verhalten soll in der Schule dadurch gefördert werden, dass Situationen geschaffen werden, in denen Kooperation lohnenswert erscheint.

- *Stärkung des Schüler-Ichs*: Dieser Aspekt zielt im Gegensatz zu den meisten anderen nicht auf gesellschaftliche Allgemeinbildung ab, sondern stellt den einzelnen Lernenden in den Vordergrund. Zentrale Idee der Stärkung des Schüler-Ichs ist es, dass sich die Schülerinnen und Schüler „als Subjekt begreifende, bewusst handelnde, Zivilcourage entwickelnde Persönlichkeit“ [Hey96, S. 117] erkennen müssen um alle anderen Aspekte der Allgemeinbildung überhaupt erst entfalten zu können. Diese lassen sich nicht von außen aufzwingen.

Zudem gibt Heymann Kriterien vor, die ein Thema in der Mathematik erfüllen sollte, um als allgemeinbildend zu gelten. Auf die Informatik wurde dies von Witten [Wit03] übertragen.
Die von ihm angeführte *Weltorientierung* soll „die Informationstechnik in den alltäglichen Anwendungen sichtbar und verstehbar [...] machen“ [Wit03, S. 62]. Ohne Zweifel gehören für viele Menschen die Nutzung digitaler Assistenten sowie - in Zukunft häufiger - (teil)autonom fahrender Autos zu den erwähnten alltäglichen Anwendungen deren Funktion sichtbar und verstehbar gemacht werden muss. Witten erwähnt die Künstliche Intelligenz als eine Möglichkeit fächerübergreifenden Unterrichts, welcher ein wichtiger Beitrag zur Weltorientierung ist und nennt explizit die Philosophie und Biologie als Beispiel für andere Fächer [Wit03]. Das Unterrichtsmodul beinhaltet durch die Anwendung der vermittelten Theorie auf medizinische Daten zur Brustkrebsdiagnose und die damit einhergehenden ethischen Fragestellungen diese beiden Fächer. Darüber hinaus wird durch das im Vordergrund stehende Beispiel der Regenvorhersage ein Bezug zum Schulfach Erdkunde aufgebaut.
Als weitere Aufgabe allgemeinbildenden Unterrichts führt Witten die *Anleitung zum kritischen Vernunftgebrauch* an [Wit03]. Ein wichtiger Punkt in der kritischen Ausein-

andersetzung mit IT-Systemen ist es „die Perspektive der Entwickler einzunehmen“ [Wit03, S. 64]. Die medizinische Nutzung Maschinellen Lernens stellt an die Ärzte als Anwender einen hohen Anspruch an eine korrekte Interpretation der Ergebnisse, die ihnen durch das eingesetzte Computersystem gegeben werden. Insbesondere das Wissen, dass beispielsweise die Diagnose einer Krankheit mit Hilfe Maschinellen Lernens zwar eine Hilfestellung und Anhaltspunkte für den behandelnden Arzt geben, jedoch nicht fehlerfrei sein kann, ist sowohl für die behandelnden Ärzte als auch die betroffenen Patienten immens wichtig bei der Planung der weiteren Behandlung. Für diese Einsicht werden im Unterrichtsmodul die nötigen theoretischen Grundlagen durch die Übertragung der Begriffe Sensitiviät, Spezifität und Genauigkeit auf medizinische Daten zur Brustkrebsdiagnose gelegt.
Neben diesen Kriterien der Allgemeinbildung hat Goorhuis bereits 1990 postuliert, dass Künstliche Intelligenz „eine zentrale Stellung im allgemeinbildenden Informatik-Unterricht einnehmen [sollte]“ [Goo90, S. 113]. Seiner Ansicht nach soll eine Diskussion über Gefahren des Einsatzes von Computersystemen erfolgen, welche sich gut am Beispiel der Künstlichen Intelligenz durchführen lässt. Seine beiden Forderungen, die Gefahren des Einsatzes von Computersystemen sowie die Verantwortung im Umgang mit solchen Systemen zu thematisieren, werden in der bereits erwähnten ethischen Diskussion am Ende des Moduls erfüllt. Um die Verantwortung bei der Nutzung technischer Systeme zu verdeutlichen, werden Entscheidungsbäume, von Goorhuis „induktive Werkzeuge zur Konstruktion von Expertensystemen“ [Goo90, S. 116] genannt, genutzt, welche hierzu als besonders geeignet bezeichnet werden [Goo90]. Beim Einsatz dieser Expertensysteme im Unterricht sieht Goorhuis das Problem, dass einerseits die reine Benutzung nur wenig Einsichten für die Schülerinnen und Schüler zulässt und andererseits die selbstständige Programmierung zu aufwendig wird. Durch die Betrachtung der Ergebnisse bei der Benutzung eines Entscheidungsbaums auf der einen Seite, sowie den fachlichen Hintergründen in der Erstellungsphase des Baums ohne Programmierung auf der anderen Seite, wird versucht dieser Problematik zu begegnen. Das händische Erstellen mehrerer Bäume und die Arbeit auf der für das Modul entworfenen Webseite als Tool für die Verdeutlichung des Einflusses von Hyperparametern bieten hier die Möglichkeit den Aufwand zu reduzieren.
Klafki [Kla93] hat „Schlüsselprobleme der Menschheit“ definiert, die er als Grundlage für allgemeinbildenden Unterricht ansieht. Dazu zählen neben der Auseinandersetzung mit dem Nationalitätsprinzip und der häufig verbundenen Frage nach Krieg und Frieden, der Umweltfrage, dem Bevölkerungswachstum und dem Phänomen der Ich-Du-Beziehung insbesondere auch die beiden Schlüsselprobleme der *gesellschaftlichen Ungleichheit* und der Frage nach den *Folgen des Technikeinsatzes*. Diese beiden Schlüsselprobleme werden

durch die Diskussion der Möglichkeit ein diskriminierendes System, beispielsweise bei der Bewerberauswahl, zu trainieren miteinander in Beziehung gesetzt und die hiermit verbundenen Folgen in der Abschlussdiskussion des Moduls aufgegriffen. Auch die Diskussion der medizinisch-ethischen Fragestellung der sinnvollen Nutzung maschineller Diagnosen trägt zur gedanklichen Auseinandersetzung mit diesen Schlüsselproblemen bei. Klafkis Idee der gesellschaftlichen Schlüsselprobleme ist dabei vergleichbar mit dem Aspekt der Weltorientierung in Heymanns Theorie zur Allgemeinbildung und kann in diesen eingegliedert werden.
Obwohl also die Bedeutung der Künstlichen Intelligenz für die Allgemeinbildung gegeben ist, findet das Thema kaum Einzug in die schulische Bildung. Das entwickelte Unterrichtsmodul soll diesem Problem entgegenwirken und den Schülerinnen und Schülern einen ersten Einblick in die Funktionsweise des Maschinellen Lernens und den mit dessen Einsatz verbundenen Fragestellungen geben.

1.2. Kompetenzmodell der Gesellschaft für Informatik e.V.

Das Kompetenzmodell der *Gesellschaft für Informatik e.V.* (GI) nennt verschiedene Prozess- und Inhaltsbereiche [Ges16a].
Die in diesem Unterrichtsmodul vorkommenden Prozessbereiche sind:

- *Modellieren und Implementieren*: Durch die Übertragung realer Probleme, wie zum Beispiel der Regenvorhersage und Brustkrebsdiagnose auf Entscheidungsbäume, wird „das Abbilden eines Realitätsausschnitts“[Ges16a, S. 5] und die „Realisierung mit einem Informatiksystem“[Ges16a, S. 5] durchgeführt.

- *Begründen und Bewerten*: Insbesondere das Bewerten der hypothetischen und bereits bestehenden Systeme in der Medizin wird im Modul thematisiert. Die Schülerinnen und Schüler begründen ihre Erkenntnisse sowohl bei der Wahl von Hyperparametern, als auch anhand mehrerer Quizfragen.

- *Strukturieren und Vernetzen*: Die Wahl binärer Entscheidungsbäume stellt eine strukturierte Informationsrepräsentation dar. Des Weiteren findet durch die fächerübergreifenden Anteile eine Vernetzung von informatischen Methoden mit Inhalten außerhalb der Informatik statt.

- *Darstellen und Interpretieren*: Die Schülerinnen und Schüler erstellen selbst Entscheidungsbäume und vergleichen verschiedene Entscheidungsbäume miteinander.

Im Hinblick auf die Inhaltsbereiche sind für das Modul insbesondere die folgenden Bereiche relevant:

- *Informatik, Mensch und Gesellschaft*: Die Schülerinnen und Schüler setzen sich mit den Folgen, Chancen und Risiken des Einsatzes Maschinellen Lernens auseinander.

- *Information und Daten*: Dieser Bereich ist essenzieller Bestandteil der Beschäftigung mit Maschinellem Lernen. Hierbei geht es vor allem um Daten als „Darstellung von Information in formalisierter Art“ [Ges16a, S. 9]. Durch beispielsweise die Darstellung von Wetterinformationen als für die weitere Verarbeitung geeignete Zahlen, sowie die spätere Darstellung mit Hilfe von Binärbäumen ist auch dieser Inhaltsbereich für das Modul von Bedeutung.

- *Algorithmen*: Auch dieser Inhaltsbereich findet sich im Unterrichtsmodul wieder, da den Schülerinnen und Schülern die Schritte beim Erstellen eines Entscheidungsbaums erläutert werden und diese Schritte in der Folge am Beispiel durchgeführt werden. Die im Kompetenzmodell der GI erwähnten Tests und Überarbeitungen die für die Implementierung eines Algorithmus notwendig sind werden im erweiterten Sinne durch das Testen und die Auswahl der Hyperparameter von den Schülerinnen und Schülern realisiert.

Neben diesen fachbezogenen Kompetenzen wird auch die Methodenkompetenz der Schülerinnen und Schüler gefördert, indem sie neue Prinzipien der Problemlösung durch Techniken des Maschinellen Lernens kennenlernen. Die Methodenkompetenz ist dabei ein „Bestandteil [...] von Fachkompetenz“ [Kul07, S. 11] und sollte in den Fachunterricht integriert werden.
Die Leitideen des Informatikunterrichts der GI können also auf den Themenbereich des Maschinellen Lernens übertragen werden. Dies unterstreicht erneut die Wichtigkeit des Themas für den Informatikunterricht und seine Eignung zur Förderung der beschriebenen Kompetenzen.

1.3. Aufbau der Arbeit

Zunächst werden die für diese Arbeit benötigten Grundlagen des Maschinellen Lernens in Kapitel 2 betrachtet. Kapitel 3 wird eine Anwendung dieser Grundlagen auf ein Studienerfolgsprojekt in Informatik an der Universität des Saarlandes beinhalten. Kapitel 4 wird die momentane schulische Entwicklung zu Maschinellem Lernen, sowie bereits bestehende Angebote erläutern. Nach dem Überblick über das entworfene Unterrichtsmodul zu Maschinellem Lernen in Kapitel 5, wird Kapitel 6 die zugrundeliegenden didaktischen Überlegungen darlegen. In Kapitel 7 wird eine Auswertung der Durchführungen des Unterrichtsmoduls vorgenommen. Abgeschlossen wird die Arbeit dann mit einer Zusammenfassung in Kapitel 8.

2. Grundlagen

In diesem Kapitel sollen die für diese Arbeit relevanten theoretischen Grundlagen zu Maschinellem Lernen gelegt werden.

2.1. Was ist Maschinelles Lernen?

Maschinelles Lernen ist ein Teilgebiet der Künstlichen Intelligenz. Betrachtet man das weite Feld der Einsatzmöglichkeiten des Maschinellen Lernens, stellt sich die Frage, was diese Anwendungen verbindet. Allen gemein ist die Tatsache, dass der Computer selbstständig durch *lernende Algorithmen* gelernt hat. Im Gegensatz zu nicht-lernenden Algorithmen werden diese nicht vollständig durch den Programmierer vorgeschrieben, sondern die Regeln und Vorschriften werden vom Computer selbst erstellt. Eine bekannte Definition von Tom Mitchell lautet im Original

Definition 2.1 (Maschinelles Lernen). *[Mit97, S. 2]*
*A computer program is said to learn from experience **E** with respect to some class of tasks **T** and performance measure **P**, if its performance at tasks in **T**, as measured by **P**, improves with experience **E**.*

Diese Definition gibt den Grundgedanken des Maschinellen Lernens wieder. Wenn wir eine Aufgabe **T** mit Hilfe Maschinellen Lernens lösen wollen, müssen wir dem Computer die Möglichkeit geben, Erfahrung **E** zu sammeln. Wir bewerten die Qualität der Lösung anhand eines von uns gewählten Kriteriums **P** zur Leistungsmessung.
Der *MNIST*-Datensatz [LCB] ist ein bekanntes Beispiel für den Einsatz des Maschinellen Lernens. Aufgabe **T** ist es, die handgeschriebenen Ziffern korrekt zu bestimmen. Unser Kriterium zur Leistungsmessung **P** kann dann die Gesamtgenauigkeit sein, also der Anteil an korrekt klassifizierten Ziffern an allen klassifizierten Ziffern. Um den Computer überhaupt in die Lage zu versetzen diese Aufgabe zu erfüllen, müssen wir ihm die Möglichkeit bieten Erfahrung **E** zu sammeln.

2.2. Arten Maschinellen Lernens

2.2.1. Überwachtes vs. unüberwachtes vs. bestärkendes Lernen

Neben weiteren Möglichkeiten der Unterscheidung der verschiedenen Typen des Maschinellen Lernens wird häufig anhand des Grads der Überwachung während der Lernphase eines Algorithmus unterschieden.
Beim *überwachten Lernen* werden dem Computer möglichst viele Beispieldatensätze, die sogennanten *Trainingsdaten*, bereitgestellt um Erfahrung **E** zu sammeln. Die Datensätze der Trainingsdaten enthalten, neben den eigentlichen Merkmalen, die zur Unterscheidung und Vorhersage dienen, auch *Labels*. Diese Labels sind die gewünschten Ergebnisse für die einzelnen Datensätze. Bei einer Aufgabe wie der Ziffernerkennung wird den Bildern der handgeschriebenen Ziffern, aus denen der Computer lernen soll, auch die korrekte Antwort mitgegeben. So kann die Maschine während des Lernens die Muster analysieren, die zu einer bestimmten Ziffer gehören und dieses Wissen später beim produktiven Einsatz nutzen, um auch ohne die Angabe der Labels Vorhersagen zu treffen. Aus dieser Überwachung und Steuerung des Lernprozesses leitet sich der Name überwachtes Lernen ab.
Im Gegensatz hierzu finden sich in den Trainingsdaten beim *unüberwachten Lernen* keine Labels. Der Computer analysiert auch hier die zur Verfügung stehenden Trainingsdaten, bildet jedoch selbstständig die Vorhersagen. Unüberwachtes Lernen kann zum *Clustering* [Gé17] eingesetzt werden.

Beispiel 2.1 (Clustering).
Ein Online-Shop Betreiber möchte seine Kunden mit Hilfe von unüberwachtem Lernen analysieren. Der eingesetzte Algorithmus findet Gemeinsamkeiten zwischen den einzelnen Kunden und gruppiert diese entsprechend in Clustern. So könnte er erkennen, dass 30% der männlichen Kunden abends elektronische Geräte kaufen, während 10% der weiblichen Kunden sich morgens für Bücher interessieren. Mit Hilfe dieses Wissens können dann zu bestimmten Tageszeiten den verschiedenen Kundengruppen unterschiedliche Produkte angeboten werden und so der Umsatz gesteigert werden.

Neben halbüberwachten Algorithmen, bei denen nur ein Teil der Daten Label hat und die somit eine Kombination aus überwachtem und unüberwachtem Lernen sind, existieren Algorithmen des *bestärkenden Lernens*. Ein bekanntes Beispiel für ein Programm welches bestärkendes Lernen nutzt, ist AlphaGo von DeepMind [SHM+16]. Beim bestärkenden Lernen entscheidet ein *Agent* welche Aktion als nächstes ausgeführt werden soll. Ist die Aktion gewünscht, so erhält er Belohnungen, andernfalls eine Bestrafung. So kann die beste Strategie, auch *Policy* genannt, erlernt werden [Gé17].

2.2.2. Klassifikation vs. Regression

Die Unterscheidung zwischen Klassifikation und Regression spielt hauptsächlich im Bereich des *überwachten Lernens* eine Rolle. Beim oben angegebenen Beispiel der Ziffernerkennung ist das Ziel die korrekte Einordnung der Daten in verschiedene Klassen. Diese Klassen können binär sein, wie bei einem Spam-Filter, der nur zwischen *Spam* und *nicht Spam* unterscheidet, oder aus mehreren Kategorien bestehen. Sobald durch den Computer eine Entscheidung für eine bestimmte Klasse getroffen werden soll, handelt es sich um eine *Klassifikationsaufgabe*.
Im Gegensatz hierzu gibt es Aufgaben, die nicht durch eine einfache Zuordnung zu Klassen gelöst werden können. Während die Gehaltsvorhersage einer Person, bei der zwischen niedrigem, mittlerem und hohem Gehalt unterschieden wird, zwar schon eine grobe Richtung vorgibt, ist es wesentlich zielführender und genauer, einen konkreten Wert vorherzusagen. Da es theoretisch unbegrenzt viele Möglichkeiten für das Gehalt gibt, liegen hier keine einzelnen abgegrenzten Klassen, sondern kontinuierliche Ergebnisse vor, weshalb man von einer *Regressionsaufgabe* spricht.

2.3. Bewertung der Qualität eines Modells

Um die Qualität eines trainierten Modells zu bewerten gibt es verschiedene Möglichkeiten. Für die in dieser Arbeit betrachteten Algorithmen des überwachten Lernens gilt, dass zwischen *Trainings- und Testdaten* unterschieden werden muss. Anhand der Trainingsdaten lernt das Modell und wird trainiert. Würde man nun die Qualität des Modells auf diesen Daten überprüfen, wäre die Wahrscheinlichkeit recht hoch, dass es sehr performant ist. Schließlich wurde es hierauf trainiert. Allerdings kann so keine Aussage getroffen werden wie gut das Modell mit neuen und unbekannten Daten umgeht.

Daher wird das Modell auf einem Teil der gesamten zur Verfügung stehenden Daten getestet und beurteilt, die nicht in den Lernprozess eingeflossen sind. Diese werden Testdaten genannt.

2.3.1. Klassifikation

Die intuitivste Idee um die Qualität einer Klassifikation zu beurteilen ist die Berechnung der Genauigkeit.

Definition 2.2 (Genauigkeit).
Die Genauigkeit ist der Anteil der korrekten Vorhersagen an allen Vorhersagen.

$$Genauigkeit = \frac{korrekte\ \ Vorhersagen}{alle\ \ getätigten\ \ Vorhersagen}$$

Die Genauigkeit ist einfach zu interpretieren, hat aber auch Nachteile. Sofern die Anzahl an Datensätzen innerhalb der Klassen nicht ungefähr gleich groß ist, können ungewollte Effekte auftreten. Im Fall eines Spam-Filters sind im Normalfall die meisten untersuchten E-Mails kein Spam. Erreicht ein Modell eine Genauigkeit von 93%, so sieht dies auf den ersten Blick sehr gut aus. Wenn allerdings lediglich 10% der E-Mails Spam sind, so würde bereits ein Modell, welches für jede E-Mail die Vorhersage *kein Spam* trifft eine Genauigkeit von 90% erreichen. Dies relativiert die erreichten 93% sehr stark.
Zielführender ist hier die Betrachtung des Fehlers 1. und 2. Art. Hierzu benötigen wir die folgenden beiden Definitionen.

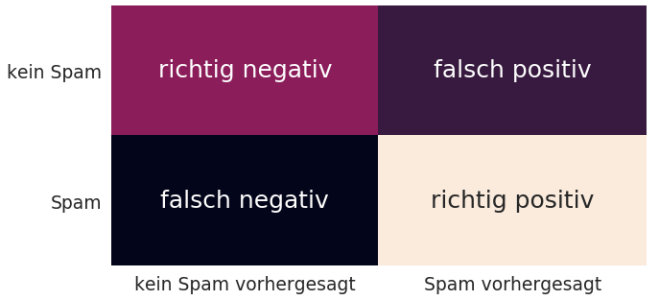


Abbildung 2.1.: Vierfeldertafel

Definition 2.3 (Sensitivität).
Sensitivität ist der Anteil der korrekt als positiv klassifizierten Datensätze an allen positiven Datensätzen.

$$Sensitivität = \frac{richtig\ positiv}{falsch\ negativ + richtig\ positiv}$$

Definition 2.4 (Spezifität).
Spezifität ist der Anteil der korrekt als negativ klassifizierten Datensätze an allen negativen Datensätzen.

$$Spezifität = \frac{richtig\ negativ}{falsch\ positiv + richtig\ negativ}$$

Die Sensitivität und Spezifität kann über die Anpassung eines Schwellenwerts beeinflusst werden. Insbesondere für Klassifikationsaufgaben, bei denen mehr als zwei Klassen existieren, ist dieser Schwellenwert von Bedeutung. Mit ihm wird festgelegt, ab wann ein Ergebnis als positiv klassifiziert wird. Im Falle des Spam-Beispiels bedeutet die Verringerung des Schwellenwerts, dass mehr E-Mails als Spam eingestuft werden. Somit steigt zum einen die Anzahl der richtig positiv klassifizierten E-Mails, zum anderen aber auch die Anzahl der falsch positiv klassifizierten E-Mails. Dies hat eine Steigerung der Sensitivität und eine Senkung der Spezifität zur Folge. Hieraus ergibt sich, dass man je nach Anwendung entscheiden muss, ob mehr Fokus auf eine hohe Sensitivität oder Spezifität gelegt wird.
Um trotzdem anhand eines einzelnen Wertes die Güte des Modells zu bestimmen, betrachten wir die sogenannte *Receiver Operating Characteristic* (ROC) Kurve. Bei dieser wird die Sensitivität gegen 1−Spezifität in einem Diagramm für verschiedene Schwellenwerte aufgetragen [Lan11]. Im Anschluss wird die *Area Under Curve* (AUC) als bestimmtes Integral berechnet. Je näher die AUC an 1 liegt, desto besser sind die Vorhersagen des Modells. Ein rein zufälliges Modell, in Abbildung 2.2 durch die rote gestrichelte Linie dargestellt, hätte unabhängig von den gegebenen Proportionen der Klassen eine AUC von 0,5 [Gé17].

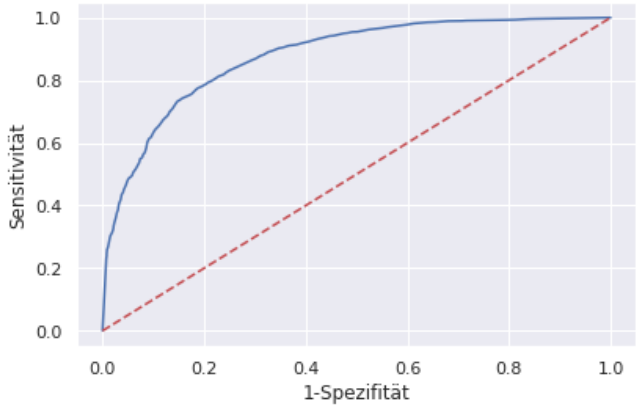


Abbildung 2.2.: ROC-Kurve (blau) mit einer AUC von 0.88

Bei einer Wahrscheinlichkeit von p für die Vorhersage von Klasse A und $1-p$ für die Vorhersage von Klasse B, sowie einem Anteil von $\frac{x}{x+y}$ für Klasse A und $\frac{y}{x+y}$ für Klasse B an allen Datensätzen, gilt, sofern Klasse A als positive Klasse betrachtet wird, mit obigen Formeln

$$\begin{aligned} \textit{Sensitivität} &= \frac{p \cdot \frac{x}{x+y}}{(1-p) \cdot \frac{x}{x+y} + p \cdot \frac{x}{x+y}} \\ &= p = 1 - (1-p) \\ &= 1 - \frac{(1-p) \cdot \frac{y}{x+y}}{p \cdot \frac{y}{x+y} + (1-p) \cdot \frac{y}{x+y}} \\ &= 1 - \textit{Spezifität} \end{aligned}$$

Eine weitere Möglichkeit mehrere Werte zu kombinieren ist der *F1-Score*. Der F1-Score ist ein Maß für die Genauigkeit. Er ist das harmonische Mittel des positiven Vorhersagewerts und der Sensitivität.

Definition 2.5 (Positiver Vorhersagewert).
Der positive Vorhersagewert (Precision) ist der Anteil der richtig positiv klassifizierten Datensätze an allen als positiv klassifizierten Datensätzen.

$$\textit{Precision} = \frac{\textit{richtig positiv}}{\textit{falsch positiv} + \textit{richtig positiv}}$$

Mit dem positiven Vorhersagewert kann dann der F1-Score definiert werden als:

Definition 2.6 (F1-Score).

$$F1 - Score = 2 \cdot \frac{Precision \cdot Sensitivität}{Precision + Sensitivität}$$

Wie zu erkennen ist, werden die Sensitivität und der positive Vorhersagewert gleich gewichtet. Je nach Kontext ist jedoch teilweise eine andere Gewichtung erwünscht, da beispielsweise das Vorhandensein von vielen falsch negativ klassifizierten Datensätzen schwerwiegender ist und somit die Sensitivität einen größeren Einfluss hat. Aus diesem Grund existiert die allgemeine Form des $F\beta$-Scores, bei der über den Parameter β das Gewicht von Sensitivität und positivem Vorhersagewert beeinflusst werden kann. Je höher β ist, desto wichtiger wird die Sensitivität.

Definition 2.7 ($F\beta$-Score).

$$F\beta - Score = \left(1 + \beta^2\right) \cdot \frac{Precision \cdot Sensitivität}{\beta^2 \cdot Precision + Sensitivität}$$

Analog zur Area Under Curve der ROC, können auch der positive Vorhersagewert und die Sensitivität in einem Diagramm für verschiedene Schwellenwerte dargestellt werden. Auch hier lässt sich die Area Under Curve berechnen. In Abbildung 2.3 stellt die gestrichelte Linie wiederum ein rein zufälliges Modell dar.

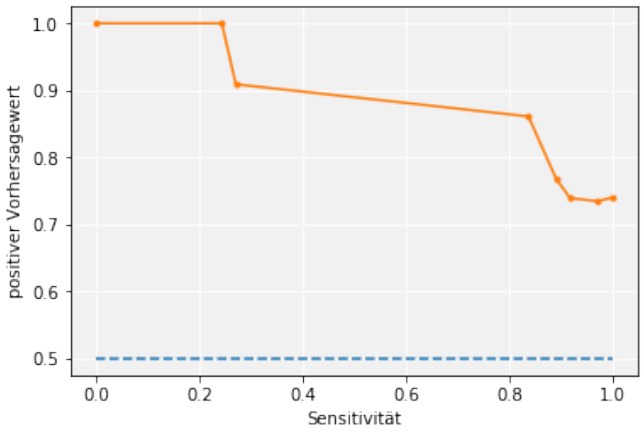


Abbildung 2.3.: Precision-Sensitivität-Kurve mit einer AUC von 0.896

2.3.2. Regression

Im Falle der Regression sind aufgrund der nicht vorhandenen Klassen andere Kriterien nötig um die Leistung des Modells zu beurteilen. Ein häufig genutzter Wert ist die Wurzel des mittleren quadratischen Fehlers (RMSE).

Definition 2.8 (Wurzel des Mittleren Quadratischen Fehlers).

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(p(\vec{x}^{(i)}) - y^{(i)})^2}$$

Dabei gilt:

- n*: Anzahl an Datensätzen*
- $\vec{x}^{(i)}$*: Vektor mit allen Attributen die zu Datensatz i gehören*
- $p(\vec{x}^{(i)})$*: Funktion zum Vorhersagen des Wertes, berechnet aus dem Attributsvektor* $\vec{x}^{(i)}$
- $y^{(i)}$*: Tatsächlicher Wert des i-ten Datensatzes, der durch das Label angegeben wird*

Dabei wird die Wurzel des mittleren quadratischen Fehlers dem mittleren quadratischen Fehler (MSE) normalerweise vorgezogen, um eine bessere Interpretation der Ergebnisse zuzulassen. Durch das Ziehen der Wurzel stimmen die Einheit des Fehlers und des Labels überein. Da die Abweichungen $p(\vec{x}^{(i)}) - y^{(i)}$ quadriert werden, gehen große Fehler stärker in die Berechnung des RMSE ein als geringe Abweichungen. Somit kann der RMSE durch Ausreißer beeinflusst werden.
Um diesen manchmal unerwünschten Einfluss von Ausreißern zu verhindern, kann der mittlere absolute Fehler (MAE) genutzt werden, da hier die Abweichungen linear einfließen.

Definition 2.9 (Mittlerer Absoluter Fehler).

$$MAE = \frac{1}{n} \sum_{i=1}^{n} \left| p(\vec{x}^{(i)}) - y^{(i)} \right|$$

Sofern kaum Ausreißer vorhanden sind oder diese bewusst stärker gewichtet werden sollen, ist der RMSE zu bevorzugen [Gé17, CD14].

2.4. Mögliche Probleme und ihre Lösungen

2.4.1. Bias vs. Variance

Betrachtet man die Ergebnisse eines trainierten Modells des überwachten Lernens auf den Trainingsdaten, können speziell zwei Probleme auftreten.
Entweder sind die Ergebnisse sehr schlecht (hoher Bias bzw. Verzerrung) oder sehr gut (hohe Variance bzw. Varianz). Verständlicherweise sind sehr schlechte Ergebnisse nicht gewollt, aber auch augenscheinlich sehr gute Ergebnisse auf den Trainingsdaten sind nicht wünschenswert.
Ist die Vorhersage auf den Trainingsdaten sehr gut, so besteht die Gefahr, dass das Modell die Daten bis in kleinste Feinheiten analysiert hat. Bereits leichte Änderungen in den Daten würden also zu großen Abweichungen in der Vorhersage führen. Somit ist das Modell an die vorhandenen Trainingsdaten überangepasst (overfitting). Dies verringert die Fähigkeit des Modells zur Generalisierung und damit neue, nicht zum Training verwandte Daten korrekt vorherzusagen.
Ist die Vorhersage trotz ausreichendem Training auf den Trainingsdaten schlecht, spricht man von einer Unteranpassung (underfitting) des Modells auf die Trainingsdaten. Das heißt, das verwendete Modell war nicht in der Lage die Zusammenhänge innerhalb der Trainingsdaten korrekt zu analysieren und hat zu schwache Annahmen verwendet.
Um eine hohe Verzerrung zu erkennen reicht es aus, die Qualität der Vorhersagen auf den Trainingsdaten zu betrachten. Ist diese schlecht, liegt eine hohe Verzerrung vor. Sind die Vorhersagen auf den Trainingsdaten sehr gut, aber auf den Testdaten deutlich schlechter, so liegt eine hohe Varianz vor.
Abbildung 2.4 verdeutlicht die Über- und Unteranpassung durch ein Beispiel der linearen und polynomiellen Regression der Cosinus-Funktion. Je höher der Grad der Regressionsfunktion, desto mehr Einzelheiten werden im Modell berücksichtigt. In diesem Fall ist

die lineare Regression unterangepasst, während das Polynom mit Grad 20 überangepasst ist. In der Folge verallgemeinern beide Modelle nur unzureichend.

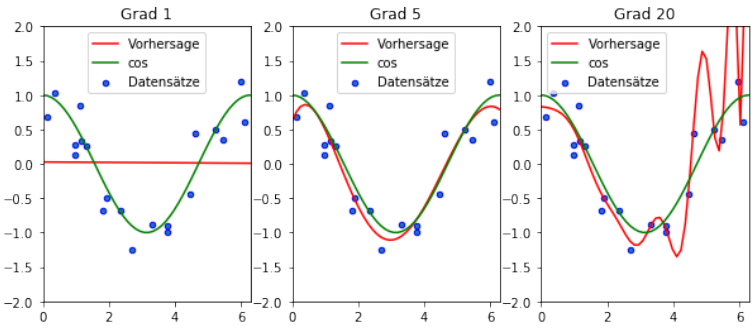


Abbildung 2.4.: Over- und Underfitting

2.4.2. Overfitting

Um Overfitting zu vermeiden gibt es mehrere Möglichkeiten [Gé17].

1.) einfacheres Modell wählen:
Wenn ein Modell, wie die polynomielle Regression mit Grad 20, für das zugrundeliegende Problem zu komplex ist, werden durch das Modell Zusammenhänge innerhalb der Trainingsdaten gefunden, die es nicht gibt. Somit fließen falsche Erkenntnisse in die Berechnung mit ein. Ein Polynom von Grad 20 hat 21 Freiheitsgrade. Hierdurch gibt es 21 Parameter, die bei der Berechnung der besten Anpassung an die Trainingsdaten verändert werden können. Folglich ist es möglich, mehr Informationen zu berücksichtigen als bei einem linearen Modell, bei dem lediglich 2 Parameter (y-Achsenabschnitt und Steigung) angepasst werden können. Dementsprechend ist ein Modell mit vielen Freiheitsgraden häufig anfälliger für Overfitting.

2.) vorhandenes Modell einschränken (siehe Abschnitt 2.5.1)

3.) weniger Attribute nutzen (siehe Abschnitt 2.6)

4.) mehr Daten sammeln:
Mehr Daten bedeuten im Normalfall mehr Variabilität innerhalb der Daten. Somit

wird es für den Algorithmus schwerer, Detailinformationen zu finden und zu berücksichtigen, die nur auf wenige Spezialfälle zutreffen. Leider ist es in der Praxis nicht immer möglich, mehr Daten zu sammeln.

5.) Rauschen innerhalb der Daten verringern:
Ist es offensichtlich, dass einige Datensätze Ausreißer sind oder Fehler enthalten, so können diese in der Trainingsphase bewusst ausgeschlossen werden, damit diese Spezialfälle keine zu hohe Bedeutung im fertigen Modell erhalten.

2.4.3. Underfitting

Analog zum Overfitting, können bei einem Modell mit Underfitting folgende Punkte zum Erfolg führen [Gé17]:

- komplexeres Modell wählen, welches mehr Freiheitsgrade besitzt um die Informationen ausreichend abzubilden

- weniger Einschränkungen verwenden

- bessere und relevantere Attribute bereitstellen:
Mit Hilfe von *Feature Engineering* können die vorhandenen Attribute (Features) optimiert werden. Es gibt verschiedene Möglichkeiten des Feature Engineering. Zum einen können die vorhandenen Attribute untersucht und lediglich die relevantesten ausgewählt werden um das Modell zu trainieren (dies wird in Abschnitt 2.6 genauer behandelt). Zum anderen können verschiedene Attribute kombiniert werden um bessere Ergebnisse mit relevanteren Informationen zu erzielen. Insbesondere bei Modellen wie Entscheidungsbäumen (Abschnitt 2.5.1) wird häufig die *Principal Component Analysis* (PCA) zur Erzeugung neuer Attribute verwendet [Gé17]. Dazu werden die Korrelationen und Kovarianzen zwischen den einzelnen Attributen genutzt um die Hauptkomponenten der Attribute zu erhalten. Die i-te Hauptkomponente kann durch eine lineare Funktion

$$f_i = \vec{c_i} \cdot \vec{a} = \sum_{k=1}^{n} c_{1,k} \cdot a_k$$

berechnet werden. Die einzelnen Variablen haben dabei die folgenden Bedeutungen:

- $\vec{a}$: Vektor mit allen Attributen

- $\vec{c_i}$: Vektor mit konstanten Faktoren für die i-te Hauptkomponente. $\vec{c_i}$ ist der i-te Eigenvektor der Kovarianzmatrix der Attribute.

- n: Anzahl der Attribute

Die Funktion f_1 wird dabei so gewählt, dass sie die höchste Varianz in den Komponenten von $\vec{a}$ hat. f_i ist die von $f_1, f_2, f_3, ..., f_{i-1}$ unabhängige lineare Funktion mit der höchsten Varianz in $\vec{a}$ [Jol02].
Dieses Vorgehen wird so lange wiederholt, bis eine ausreichend hohe Varianz von $\vec{a}$ durch die Hauptkomponenten erklärt wird. Für eine erwünschte Varianz von 90% bedeutet dies also, dass die Hauptkomponenten f_1 bis f_i 90% der gesamten Varianz aller Attribute abbilden. Die Korrelation der Attribute wurde minimiert und die Dimension des Attributraums entsprechend von n auf i reduziert. Damit können aus den vorhandenen Attributen neue Attribute generiert werden, die, obwohl sie in ihrer Anzahl reduziert wurden, trotzdem einen großen Teil der in den Daten vorhandenen Information wiedergeben. Das Ziel hierbei ist es, aussagekräftigere Attribute zu erzeugen und gleichzeitig die Zahl der Attribute zu verringern.

2.5. Verfahren des überwachten Lernens

Im Folgenden werden die für diese Arbeit relevanten Entscheidungsbäume und die mit ihnen verwandten Random Forests vorgestellt.

2.5.1. Entscheidungsbäume

Entscheidungsbäume sind sehr intuitive Modelle des Maschinellen Lernens. Sie geben die Antwort auf genau eine zu beantwortende Frage, für die sie erstellt wurden.

Definition 2.10 (Entscheidungsbaum). *[MR05]*
Ein Entscheidungsbaum ist ein geordneter und gerichteter Baum. Er teilt die gesamte Menge der Trainingsdaten in jedem inneren Knoten anhand einer diskreten Funktion der Attributwerte in zwei oder mehr Teilmengen auf.

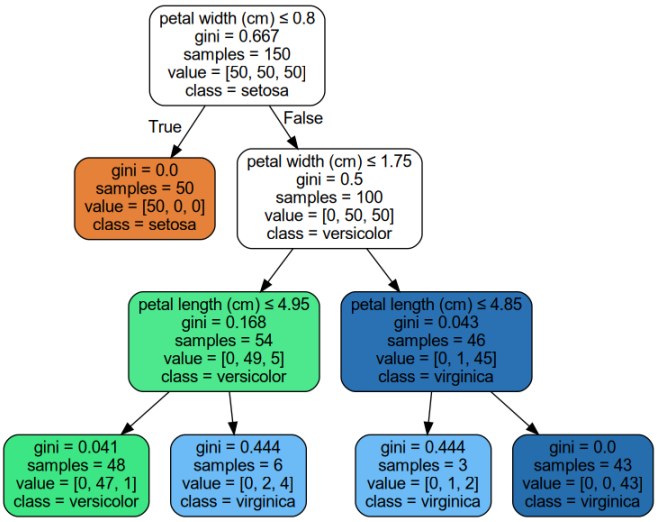


Abbildung 2.5.: Entscheidungsbaum

Abbildung 2.5 zeigt einen einfachen binären Entscheidungsbaum, welcher Vorhersagen auf dem bekannten Iris-Flower Datensatz [Fis88] trifft. Auf jede Frage gibt es in diesem Beispiel nur die Antworten *True/Ja* oder *False/Nein*, woraus die binäre Struktur des Entscheidungsbaums folgt.

In diesem Fall wird der Entscheidungsbaum für eine Klassifikationsaufgabe genutzt. Es soll mit Hilfe der Länge und Breite der Kelch- und Blütenblätter die Frage beantwortet werden, um welche Art der Schwertlilie es sich handelt. Dabei werden drei Arten betrachtet, woraus drei Klassen resultieren.
Es ist jedoch auch möglich Entscheidungsbäume in Regressionsaufgaben einzusetzen. Beispielsweise könnte die Gehaltsvorhersage aus Abschnitt 2.2.2 mit einem Entscheidungsbaum gelöst werden. Hierbei würde dann nicht eine Klasse in den Blättern des Entscheidungsbaums angegeben werden, sondern ein konkreter Wert.

Erstellen und Nutzen eines Entscheidungsbaums

Wenn wir mit Hilfe des fertigen Entscheidungsbaums aus Abbildung 2.5 eine neue Schwertlilie einordnen wollen, so können wir mit Hilfe des Baums sehr einfach eine Entscheidung treffen. Angenommen die Schwertlilie hat die folgenden Eigenschaften:

Breite des Blütenblatts	1,1cm
Breite des Kelchblatts	2,8cm
Länge des Blütenblatts	3,2cm
Länge des Kelchblatts	5,3cm

Zum Einordnen dieser Schwertlilie fangen wir in der Wurzel des Entscheidungsbaums an und sehen, dass die Bedingung „petal width (cm) $\leq 0,8$“ nicht erfüllt ist, da die Breite des Blütenblatts $1,1$ cm beträgt und somit der Schwellenwert von $0,8$ überschritten wird. Wir folgen also der rechten Kante, da die Antwort *False* ist, und stellen die nächste Frage nach „petal width (cm) $\leq 1,75$“. Dies trifft zu und somit ist die nächste Frage „petal length (cm) $\leq 4,95$“, worauf die Antwort positiv ist. Wir sind also im linken grünen Blatt des Entscheidungsbaums angekommen und die Entscheidung, dass es sich höchstwahrscheinlich um eine `Iris versicolor` handelt, ist getroffen.
Man kann sich einen Entscheidungsbaum also als eine Art Entscheidungskette nach Art einer Abfolge von `if`-Abfragen beziehungsweise einer Konjunktion von Bedingungen vorstellen.

```
if !(petal width (cm) <= 0.8) then
    if (petal length (cm) <= 4.95) then \nopagebreak
        Iris versicolor
```

Dies verdeutlicht einen der größten Vorteile von Entscheidungsbäumen: Sie sind leicht zu interpretieren [Kub15]. Im Gegensatz zu vielen anderen Modellen des Maschinellen Lernens kann jede Entscheidung anhand der abgefragten Eigenschaften einfach nachvollzogen werden. Ist der Entscheidungsbaum erstellt, so kann er auch vollkommen unabhängig von einem Computer per Hand genutzt werden [Gé17].
Zur Erstellung des Baums gibt es verschiedene Algorithmen. Die bekanntesten sind ID3 [Qui86], C4.5 als dessen Weiterentwicklung [Qui93] , sowie CART (*Classification And Regression Trees*) [BFSO84]. Die für diese Arbeit verwendete Implementierung aus der Python-Bibliothek *scikit-learn* [dev19] nutzt den CART Algorithmus, weshalb dieser Algorithmus im Folgenden genauer betrachtet wird.
Der CART Algorithmus folgt dabei einer recht einfachen rekursiven Struktur:

1. Finde den besten Split unter allen zur Verfügung stehenden Attributen

2. Teile den Datensatz anhand des besten Splits in zwei Teilmengen

3. Wiederhole Schritte 1. und 2. auf den beiden Teilmengen rekursiv so lange, bis kein Split mehr gefunden werden kann, der zur Entscheidung beiträgt, oder ein Stopp-Kriterium erfüllt ist (siehe Abschnitt 2.5.1)

Den besten Split finden

Der Hauptbestandteil des CART-Algorithmus besteht aus dem Finden des besten Splits. Hierbei gibt es verschiedene Verfahren, die für Regressions- bzw. Klassifikationsaufgaben unterschiedlich sind.

Klassifikation Für Klassifikationsaufgaben betrachten wir zwei Split-Kriterien. Das erste dieser Kriterien ist der *Gini-Index*. Der Gini-Index ist ein Maß für die Inhomogenität einer Population.

Definition 2.11 (Gini-Index des i-ten Knotens). *[Gé17]*
Der Gini-Index des i-ten Knotens wird definiert als

$$gini(i) = 1 - \sum_{k=1}^{n} \left(\frac{a_k}{m_i}\right)^2$$

wobei

- n*: Anzahl der verschiedenen Klassen der im i-ten Knoten vorhandenen Datensätze*
- a_k*: Anzahl der Datensätze zu Klasse k im i-ten Knoten*
- m_i*: Anzahl aller Datensätze im i-ten Knoten*

Intuitiv kann der Gini-Index mit dem Urnenmodell der Wahrscheinlichkeitsrechnung verglichen werden.

Beispiel 2.2 (Gini-Index).
Insgesamt seien $m = 10$ *Kugeln in einer Urne. Sei* $n = 2$ *die Anzahl an verschiedenen Klassen (weiße und schwarze Kugeln),* $a_1 = 4$ *die Anzahl an weißen Kugeln,* $a_2 = 6$ *die Anzahl an schwarzen Kugeln. Dann gilt bei Ziehen mit Zurücklegen, dass die Wahrscheinlichkeit eine weiße Kugel zu ziehen bei* $p_1 = \frac{4}{10}$ *und die Wahrscheinlichkeit eine schwarze Kugel zu ziehen bei* $p_2 = \frac{6}{10}$ *liegt. Somit ist die Wahrscheinlichkeit bei zweimaligem Ziehen zwei schwarze Kugeln zu ziehen* $p_1^2 = \left(\frac{4}{10}\right)^2$ *und analog für die weißen Kugeln* $p_2^2 = \left(\frac{6}{10}\right)^2$*. Die Wahrscheinlichkeit bei zweimaligem Ziehen jeweils exakt eine Kugel jeder Klasse zu ziehen ist* $p_1 \cdot p_2 + p_2 \cdot p_1 = 1 - (p_1^2 + p_2^2) = 1 - \left(\left(\frac{4}{10}\right)^2 + \left(\frac{6}{10}\right)^2\right) = 0.48$*. Dies entspricht der obigen Definition des Gini-Index. Je heterogener die vorhandenen Kugeln sind, desto größer ist die Wahrscheinlichkeit unterschiedliche Kugeln zu ziehen und damit auch der Gini-Index.*

Der beste Split ist dann derjenige, der die folgende Kostenfunktion minimiert.

Definition 2.12 (Kostenfunktion des CART Algorithmus mit Gini-Index).
[Gé17]
Die Kostenfunktion bezüglich des Gini-Index kann als ein gewichteter Gini-Index bezeichnet werden und wird definiert als

$$kosten(A, s_A) = \frac{m_{links}}{m} \cdot gini(links) + \frac{m_{rechts}}{m} \cdot gini(rechts)$$

wobei

- *A: gewähltes Attribut*
- *s_A: gewählter Schwellenwert für Attribut A*
- *m_{links} bzw. m_{rechts}: Anzahl aller Datensätze im linken bzw. rechten entstehenden Kindknoten*
- *$gini(links)$ bzw. $gini(rechts)$: Gini-Index des linken bzw. rechten Kindknoten*

Je kleiner dieser Wert ist, desto homogener sind die verbleibenden Daten in den Kindknoten und desto besser ist die gewählte Attribut-Schwellenwert-Kombination zur Unterscheidung der Klassen geeignet.
Eine andere Möglichkeit zum Finden des besten Splits in Klassifikationsaufgaben ist der sogenannte *Information Gain*, welcher auf der *Entropie* basiert.

Definition 2.13 (Entropie des i-ten Knotens). *[Gé17]*
Für die Entropie des i-ten Knotens gilt

$$entropie(i) = -\sum_{\substack{k=1 \\ \frac{a_k}{m_i} \neq 0}}^{n} \frac{a_k}{m_i} \cdot log\left(\frac{a_k}{m_i}\right)$$

Die Entropie gibt an, wie viel Information aus dem Knoten gewonnen werden kann. Sind alle Datensätze des i-ten Knotens von der selben Klasse, so gilt $entropie(i) = 0$, da durch eine weitere Aufteilung keine neuen Erkenntnisse gewonnen werden können.
Analog zur Kostenfunktion des Gini-Index kann auch hier die gewichtete Entropie der entstehenden Nachfolgerknoten berechnet werden. Die Veränderung der Entropie durch

einen Split ist dann die Differenz aus der Entropie des aufzuspaltenden Knotens und der gewichteten Entropie der Nachfolgerknoten. Ausgewählt wird dann derjenige Split, bei dem die Entropien der Nachfolgerknoten minimal sind.

Regression Ohne die Einordnung in verschiedene Klassen können weder der Informationsgewinn, noch der Gini-Index genutzt werden. Im Fall der Regression werden daher andere Kriterien benötigt um den besten Split zu finden.
Häufig verlässt man sich hierbei auf die bereits in 2.3.2 definierten mittleren quadratischen Fehler (MSE) bzw. mittleren absoluten Fehler (MAE).
Es wird derjenige Split gewählt, der den MSE bzw. MAE minimiert. Somit wird die folgende Kostenfunktion minimiert.

Definition 2.14 (Kostenfunktion des CART Algorithmus für Regression).
[Gé17]
Die Kostenfunktion für Regression wird definiert als

$$kosten(A, s_A) = \frac{m_{links}}{m} \cdot MSE(links) + \frac{m_{rechts}}{m} \cdot MSE(rechts)$$

wobei

- $MSE(links)$ *bzw.* $MSE(rechts)$*: MSE im linken bzw. rechten Kindknoten*

Analog kann diese Kostenfunktion auch mit dem MAE berechnet werden. Die auch im Verlaufe dieser Arbeit genutzte Python-Bibliothek *scikit-learn* bietet beide Optionen [dev19].

Hyperparameter

Hyperparameter sind Parameter, mit denen man den Algorithmus während der Trainingsphase einschränken kann. Sie werden bereits vor der Trainingsphase festgelegt und somit nicht aus den Daten bestimmt [CM15].
Erkennt man nach dem Training des Modells, dass die Ergebnisse auf den Testdaten nicht zufriedenstellend sind, obwohl die Trainingsdaten gut vorhergesagt werden, so

liegt Overfitting vor. Insbesondere Modelle wie Entscheidungsbäume neigen häufig zu Overfitting, da anders als bei einfacheren Modellen wie der linearen Regression im Vorhinein kaum Annahmen über die Daten getroffen werden und dementsprechend die Anzahl der Freiheitsgrade sehr groß ist [Gé17]. Eine der Möglichkeiten Overfitting zu verringern, ist der Einsatz von Hyperparametern.
Die für Entscheidungsbäume wichtigsten Hyperparameter sind:

- *Maximale Tiefe des Baums*: Je größer die Tiefe des Baums ist, desto mehr Splits werden vorgenommen. Dadurch können mehr Informationen aus den Daten erlernt werden. Liegt also Overfitting vor, so gibt man eine geringere Tiefe vor.

- *Minimale Anzahl an Datensätzen pro Split*: Ist dieser Hyperparameter gesetzt, so wird ein Split nur dann durchgeführt, wenn im betrachteten Knoten mindestens so viele Datensätze wie angegeben vorhanden sind. Ist dies nicht der Fall, so ist der Knoten ein Blatt. Über diesen Hyperparameter kann die Anzahl an Entscheidungen und damit die Gefahr des Overfittings verringert werden.

- *Minimale Anzahl an Datensätzen pro Blatt*: Dieser Hyperparameter hat eine ähnliche Funktion wie der vorherige. Führt ein Split dazu, dass ein Kindknoten entsteht, welcher weniger als die vorgegebene Anzahl an Datensätzen beinhaltet, so wird der Split verworfen.

- *Maximale Anzahl an Attributen pro Split*: Wird dieser Wert gesetzt, so wird in jedem Split statt allen Attributen nur maximal die angegebene Anzahl an Attributen berücksichtigt.

- *Maximale Anzahl an Blättern*: Mit diesem Hyperparameter wird Einfluss darauf genommen wie viele Details innerhalb der Daten abgebildet werden können. Je weniger Blätter zugelassen sind, desto weniger Details werden im Baum berücksichtigt.

Alle diese Hyperparameter beeinflussen die Struktur des Entscheidungsbaums.
Wie bereits beschrieben, ist das Standardverfahren um die Fähigkeit zur Generalisierung zu überprüfen das Aufteilen der Daten in Trainings- und Testdaten. Um den Erfolg unserer Maßnahmen gegen Overfitting zu testen, ist es jedoch nicht ratsam, die Testdaten zu nutzen [Gé17]. Wenn wir die Hyperparameter auf Grundlage der Performanz auf den Testdaten festlegen, so ist implizit die Struktur dieser Daten in das Training unseres

Modells eingeflossen und der Nutzen der Testdaten als Anhaltspunkt für die Fähigkeit zur Generalisierung ist verloren gegangen. Um dieses Problem zu umgehen, benötigt man noch eine weitere Menge an Daten, die sogenannten *Validierungsdaten*. Wie vorher auch, wird das Modell anhand der Trainingsdaten trainiert. Statt jedoch die Testdaten zum Optimieren der Hyperparameter zu nutzen, verwendet man hierzu die Validierungsdaten. Erst wenn die Ergebnisse zufriedenstellend sind, wird die Fähigkeit zur Generalisierung abschließend auf den Testdaten getestet. Wichtig ist, dass hiernach keine Anpassungen am Modell mehr vorgenommen werden dürfen, da sonst das selbe Problem besteht, wie ohne die Nutzung der Validierungsdaten.
Dieses Verfahren bringt jedoch den Nachteil mit sich, dass ein Teil der zum Training zur Verfügung stehenden Daten für die Optimierung der Hyperparameter genutzt wird und nicht mehr in die Trainingsphase des Modells einfließen kann. Damit nicht eine zu große Menge an Daten auf diese Weise „verloren“ geht, setzt man *cross-validation* (CV) ein [Gé17].

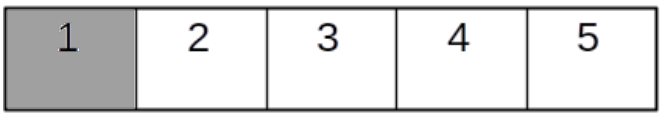


Abbildung 2.6.: 5-fache Cross-Validation

Nach dem Einteilen der Trainingsdaten in k Mengen, werden $k-1$ dieser Mengen als Trainingsdaten genutzt und die verbleibende als Validierungsdaten. Dann wird eine andere Menge als Validierungsdaten genutzt und die übrigen als Trainingsdaten. Dies wird dann so lange wiederholt, bis jede einzelne Menge einmal zum Validieren gebraucht wurde. Im Beispiel in Abbildung 2.6 wird also bspw. zuerst die Menge 1 als Validierungsdaten genutzt und Mengen 2 bis 5 als Trainingsdaten, dann ist Menge 2 der Validierungsdatensatz und 1, 3, 4, und 5 sind die Trainingsdaten usw. Nachdem mit dieser Methode das beste Modell mit seinen Hyperparametern bestimmt ist, wird dieses Modell mit allen zur Verfügung stehenden Trainingsdaten trainiert und dann final auf den Testdaten getestet.

2.5.2. Ensemble-Methoden

Um der Tendenz zum Overfitting von Entscheidungsbäumen zu begegnen gibt es sogenannte *Ensemble-Methoden.*
Bei Ensemble-Methoden werden mehrere Modelle kombiniert um die endgültige Vorhersage zu generieren. Im Falle der Klassifikation wird die Klasse vorhergesagt, welche von den einzelnen Modellen am häufigsten vorhergesagt wurde bzw. die Klasse mit der höchsten gemittelten Wahrscheinlichkeit. Im Falle der Regression wird der Durchschnitt aller Modelle als Vorhersage gewählt.
Ein sehr bekannter Ensemble Algorithmus welcher auf Entscheidungsbäumen basiert ist der Random Forest Algorithmus. Bei ihm werden mehrere kleinere Entscheidungsbäume kombiniert, die auf verschiedenen Daten trainiert wurden. Es wird für jeden einzelnen Entscheidungsbaum eine zufällige Stichprobe mit Zurücklegen der gesamten Trainingsdaten gewählt, auf dem er trainiert wird. Diese Technik nennt man *Bagging (bootstrap aggregation).*

Definition 2.15 (Bagging). *[Kub15, S. 174]*
Der Algorithmus des Bagging ist in zwei Phasen unterteilt.

Training:

1. *Bilde durch Ziehen mit Zurücklegen aus den Trainingsdaten n gleichgroße Stichproben, wobei n die Anzahl der einzelnen Modelle (bspw. Entscheidungsbäume) ist.*

2. *Trainiere auf jeder Stichprobe ein Modell.*

Vorhersage:

3. *Lasse jedes einzelne Modell die Vorhersage für den neuen Datensatz treffen und sammle die Vorhersagen.*

4. *Wähle die häufigste/wahrscheinlichste (Klassifikation) bzw. durchschnittliche (Regression) Vorhersage.*

Obwohl bei Random Forests jeder einzelne Entscheidungsbaum durch das Training auf einer kleineren Teilmenge eine höhere Verzerrung hat als beim Training auf allen Trainingsdaten haben Random Forests im Normalfall sowohl eine geringere Verzerrung als auch eine geringere Varianz als ein einzelner Entscheidungsbaum [Gé17].
Neben dem Zufall in der Auswahl der Trainingsdatensätze wird bei Random Forests auch die Auswahl der Attribute randomisiert, indem nur auf einer zufälligen Teilmenge der Attribute das jeweils beste Attribut zum Splitten bestimmt wird. Durch dieses Vorgehen werden die entstehenden Entscheidungsbäume vielfältiger und unterschiedlicher, was für die Genauigkeit der Vorhersage des Random Forest wichtig ist. Je unterschiedlicher die Fehler sind, die durch die einzelnen Entscheidungsbäume gemacht werden, desto besser funktioniert das Prinzip des Baggings [Kub15]. Die Varianz wird auf Kosten einer etwas höheren Verzerrung reduziert [Gé17], was für das Ziel der Reduzierung des Overfittings von Vorteil ist.
Ein weiterer Vorteil von Ensemble-Methoden gegenüber einzelnen Modellen wie Entscheidungsbäumen ist die Möglichkeit die Generalisierungsfähigkeit ohne zusätzliche Cross-Validation einschätzen zu können.
Durch das Bagging beim Training des Random Forest wird für jeden Entscheidungsbaum eine Menge an Datensätzen erzeugt, die nicht in das Training eingeflossen sind. Diese sogenannten *Out-of-Bag-Daten* können als Validierungsdaten für den betreffenden Entscheidungsbaum genutzt werden. Der Durchschnitt aller Entscheidungsbäume auf den Out-of-Bag-Daten bezüglich des gewählten Gütekriteriums (Genauigkeit, MAE, AUC, ...) gibt dann eine Einschätzung der Generalisierungsfähigkeit des gesamten Ensemble-Modells.

2.6. Attributauswahl

Um die Qualität des trainierten Modells zu verbessern, ist der Schritt des *Feature Engineering* von großer Bedeutung. Es ist einleuchtend, dass die Vorhersage des Modells nicht gut sein kann, wenn die Attribute die zum Training genutzt wurden nicht aussagekräftig sind.
Beim Feature Engineering kommen mehrere Techniken zum Einsatz. Zum einen kann man aus allen zur Verfügung stehenden Attributen die wichtigsten und relevantesten auswählen (*feature selection*), zum anderen können die existierenden Attribute kombiniert werden, um bessere Attribute zu erhalten (*feature extraction*) [Gé17]. Feature extraction kann mit der bereits beschriebenen Methode der Hauptkomponentenanalyse

(*PCA*) durchgeführt werden. Eine dritte Möglichkeit des Feature Engineerings ist das Sammeln neuer Daten, die auf neuen Attributen basieren.
Im Rahmen der feature selection können schrittweise die unrelevantesten Attribute entfernt werden.

Definition 2.16 (Recursive Feature Elimination (RFE)). *[GWBV02]*

1. *Trainiere ein Modell mit allen derzeit verfügbaren Attributen.*

2. *Finde das Attribut, welches am wenigsten relevant ist und lösche es aus der Menge der verfügbaren Attribute.*

3. *Wiederhole Schritte 1 und 2 bis die gewünschte Anzahl an Attributen erreicht ist.*

Ist es nicht gewünscht die Attribute auf eine feste Anzahl zu reduzieren, bietet das scikit-learn Framework die Möglichkeit, die RFE auch mit einer Cross-Validation zu kombinieren, um die optimale Anzahl an Attributen automatisiert zu finden [dev19].
Hat man sich bereits auf einen bestimmten Algorithmus für das spätere Modell festgelegt, bietet es sich an, einen verwandten Algorithmus zum Eliminieren der Attribute zu wählen, um diejenigen Attribute zu finden, die für die Performanz des späteren Modells am hilfreichsten sind. Da in dieser Arbeit Entscheidungsbäume zur Vorhersage genutzt werden, wird die RFE mit Random Forests durchgeführt.

2.7. Pruning

„Pruning“ bedeutet, dass Knoten eines Entscheidungsbaums „abgeschnitten“ werden. Die im Abschnitt 2.5.1 beschriebene Möglichkeit auf den Entscheidungsbaum mittels Hyperparametern Einfluss zu nehmen, nennt sich auch *Pre-pruning*, da bereits beim Erstellen des Baums Knoten „abgeschnitten“ und gar nicht erst weiter aufgeteilt werden, sofern bestimmte Bedingungen erfüllt sind. Im Gegensatz hierzu gibt es das Verfahren des *Post-pruning*. Hierbei wird der Baum zunächst so weit aufgespannt, dass die Trainingsdaten möglichst gut vorhergesagt werden, um im Anschluss Knoten zu entfernen [Min89]. Dabei wird bewusst Overfitting zugelassen. Das Ziel des Prunings ist es,

diejenigen Knoten zu finden, die am meisten zum Overfitting des Baums beitragen und diese dann zu entfernen.
Um dieses Ziel zu erreichen, gibt es verschiedene Methoden. Mingers [Min89] hat gezeigt, dass das *Reduced Error Pruning* (REP) und das *Critical Value Pruning* (CVP) vielversprechend sind.

2.7.1. Reduced Error Pruning

Das von Quinlan [Qui87] eingeführte REP zeichnet sich dadurch aus, dass nacheinander alle inneren Knoten des Baums in Postorder betrachtet werden und die an diesen Knoten beginnenden Teilbäume durch ein Blatt ersetzt werden. Die Performanz des so entstandenen Baums wird mit der des ursprünglichen Baums verglichen. Je nach Art des Baums wird hierfür die Genauigkeit, der F1-Wert, die AUC bzw. der RMSE oder MAE berücksichtigt. Hat sich die Performanz verbessert, so wird der Pruning-Schritt beibehalten [EMSK97]. Hierbei ist jedoch zu berücksichtigen, dass der abgeschnittene Teilbaum nur dann entfernt werden darf, wenn er keinen anderen Teilbaum enthält, der eine noch bessere Performanz hat [Qui87], weshalb sich eine Berücksichtigung der Knoten in Postorder anbietet.

2.7.2. Critical Value Pruning

CVP wird anhand des zum Erstellen des Entscheidungsbaums verwendeten Splitkriteriums durchgeführt. In unserem Fall sind dies also für Klassifikationsbäume der Gini-Index oder die Entropie, für Regressionsbäume der RMSE oder MAE. Es werden verschiedene kritische Werte als Schwellenwert festgelegt, die von den einzelnen Knoten im Baum erreicht werden müssen. Erreichen weder ein bestimmter Knoten, noch seine nachfolgenden Knoten diesen Schwellenwert, so wird der auf ihn folgende Teilbaum abgeschnitten [Min89]. Von diesen Bäumen wird im Anschluss derjenige mit der besten Performanz ausgewählt. Hierzu wird wie beim REP eine gesonderte Menge an Pruningdaten benötigt.

3. Anwendung auf Studierendendaten

Die im vorherigen Kapitel gelegten Grundlagen werden im Folgenden auf ein Projekt zur Beurteilung des Studienerfolgs übertragen. Dabei werden Daten von Studierenden der Informatikstudiengänge der Universität des Saarlandes in den Grundvorlesungen *Programmierung I* aus den Wintersemestern 2017/2018 und 2018/2019 sowie *Mathematik für Informatiker I*(MfI I) aus dem Wintersemester 2018/2019 ausgewertet. Zu Beginn der Semester wurde in der Vorlesung *Perspektiven der Informatik* ein Fragebogen an die Studierenden ausgeteilt, welcher verschiedene persönliche Merkmale erfasst. Eine genaue Beschreibung der erhobenen Daten findet sich in 3.2.1. Die Antworten der Studierenden wurden dann mit den in den Vorlesungen erreichten Noten kombiniert. Hieraus sind für die Vorlesung Programmierung I 277 und für MfI I 154 verwertbare Datensätze entstanden, bei denen sowohl die Antworten des Fragebogens als auch eine Note zur Verfügung standen.

3.1. Fragestellung und Zielsetzung

Den ersten Teil der Untersuchung der Studierendendaten stellt eine statistische Analyse der im Fragebogen erhobenen Merkmale auf Korrelationen mit den erreichten Noten dar. Ziel ist es herauszufinden, welche Merkmale am aussagekräftigsten für den Erfolg der Studierenden sind.
Im Anschluss an die statistische Analyse erfolgt die Anwendung des Maschinellen Lernens auf die Daten, um automatisiert Vorhersagen über den Erfolg der Studierenden anhand der Antworten im Fragebogen zu treffen. Es soll eine Hilfestellung gegeben werden, die bei der Entscheidung über die Zulassung zu einem Informatikstudiengang an der Universität des Saarlandes zurate gezogen werden kann. Hierfür soll ein Entscheidungsbaum eingesetzt werden, um die Interpretation der Ergebnisse auch für Laien zu ermöglichen.

3.2. Analyse der Daten

3.2.1. Struktur der Daten

In diesem Abschnitt werden die Merkmale beschrieben, welche mit Hilfe des Fragebogens, der von den Studierenden am Anfang des Semesters ausgefüllt wurde, erhoben wurden. Der Fragebogen ist an der Universität des Saarlandes im Rahmen einer Kooperation des Modeling and Simulation Lehrstuhls von Prof. Dr. Verena Wolf und des Lehrstuhls für Empirische Bildungsforschung von Prof. Dr. Roland Brünken entstanden. Die vom Lehrstuhl von Prof. Dr. Roland Brünken erstellten Erläuterungen zu den erfassten Items finden sich in Anhang B.
Der erste Abschnitt des Fragebogens erfasst allgemeine persönliche Daten der Studierenden wie das Bundesland der Abiturschule, das Geschlecht, sowie den Studiengang. Schulbezogene Daten werden ebenfalls erhoben. Dazu gehören die Durchschnittsnote der allgemeinen Hochschulreife, Angaben dazu, ob die Studierenden Leistungskurse in Mathematik, Informatik oder Physik besucht haben oder ob sie einen Grundkurs Informatik in der Schule belegt haben. Falls sie das Fach Informatik besucht haben, sollen die Studierenden angeben, wie viele Jahre dies der Fall war und wie viele Stunden pro Woche sie Informatikunterricht in der Oberstufe hatten.
Auf diesen ersten Abschnitt eher allgemeiner Fragen folgen mehrere Abschnitte mit Fragen psychologischer Natur.
Zunächst wird das Selbstkonzept der Studierenden in Mathematik und Informatik erfasst. Dabei ist das Selbstkonzept „die Gesamtheit der kognitiven Repräsentationen der eigenen Persönlichkeit bzw. des Selbst “ [Sta03, S. 347]. Um dieses Selbstkonzept zu erfassen, wird nach der eigenen Einschätzung der Leistung in Mathematik und Informatik gefragt.
Auch die Berufsmotivation im Bereich der Informatik wird untersucht. Dieser sehr umfangreiche Komplex wird in mehrere Kategorien aufgeteilt - den *beruflichen Folgenanreiz*, den *intellektuellen Folgenanreiz*, das *Image des Berufs*, das *berufliche Selbstkonzept* und den *Tätigkeitsanreiz*.
In ähnlicher Weise erfolgt auch bei den Fragen zur Messung der Persönlichkeitsmerkmale eine Aufteilung in Fragen zur *Extraversion*, *Verträglichkeit*, *Gewissenhaftigkeit* und zum *Neurotizismus*.
Bei den Selbstkonzepten, den Persönlichkeitsmerkmalen, sowie bei der hierauf folgenden Messung der Leistungsmotivation der Studierenden werden zusätzlich zu den gegebenen Antworten auch die zugehörigen Skalenmittelwerte betrachtet. Um die Leistungsmotiva-

tion abbilden zu können, nutzt der Fragebogen die Kurzform des Leistungsmotivationsinventars von Schuler und Prochaska [SP01].
Abschließend werden die psychosoziale und anforderungsbasierte Kongruenz der Studierenden gemessen. Hierzu wird das Hexagonale Modell von Holland [Hol97] als Berechnungsgrundlage genutzt, welches die sechs Interessensbereiche *praktisch-technisch*, *intelektuell-forschend*, *künstlerisch*, *sozial*, *unternehmerisch* und *konventionell* nach ihrer inhaltlichen Nähe zueinander anordnet. Durch die Selbsteinschätzung der Studierenden für welche drei Bereiche sie sich am meisten interessieren und ihre Einschätzung welche drei Bereiche für den „idealen" Informatiker am wichtigsten sind wurde der typologische C-Wert [BG94] als psychosoziale Kongruenz bestimmt. In Verbindung mit der Experteneinschätzung [BE05], dass der forschende, praktische und konventionelle Bereich am wichtigsten ist, wurde mit Hilfe der Selbsteinschätzung die anforderungsbasierte Kongruenz berechnet.
Die Daten des Fragebogens und die Noten der Studierenden sind über die verschlüsselte Matrikelnummer verbunden.

3.2.2. Korrelationsanalyse

Bevor Korrelationen berechnet werden können, muss immer eine Untersuchung der Skalenniveaus vorgenommen werden. Nur dann kann eine korrekte Korrelation bestimmt werden. So muss für den bekannten Pearson-Korrelationskoeffizienten mindestens eine Intervallskala vorliegen [Bou08]. Sofern die Variablen mindestens ordinalskaliert sind, kann die Spearman-Rangkorrelation ρ genutzt werden:

Definition 3.1 (Spearman-Rangkorrelation (Kurzversion)). *[Bou08, S. 218]*
Es gelte, dass die Rangziffern die ersten n natürlichen Zahlen sind.
Dann gilt für den Spearman-Rangkorrelationskoeffizienten:

$$\rho = 1 - \frac{6 * \sum (R_{x_i} - R_{y_i})^2}{n^3 - n}$$

wobei R_{x_i} und R_{y_i} die Rangziffern des Merkmalsträgers i bezüglich Variable x bzw. y sind.

Haben mehrere Merkmalsträger den selben Wert für Variable x oder y, so liegen *Bindungen* vor. Durch Bindungen wird die Voraussetzung, dass die Rangziffern natürliche

Zahlen sind, verletzt.

Rang bzgl. x	Rang bzgl. y	neuer Rang bzgl. x	neuer Rang bzgl. y
1	4	1	3.5
4	4	4	3.5
2	1	2	1
3	2	3	2

Nutzt man jedoch nicht die oben angeführte Kurzversion von Spearman's ρ, können auch Bindungen korrekt berücksichtigt werden.

Definition 3.2 (Spearman-Rangkorrelation). *[Cle14, S. xix]*
Spearman's ρ lässt sich als modifizierte Version des Pearson-Korrelationskoeffizienten darstellen, bei der die Rangziffern statt der Merkmalsausprägung genutzt werden:

$$\rho = \frac{\frac{1}{n}\sum_{i=1}^{n}(R(x_i) - \overline{R(x)})(R(y_i) - \overline{R(y)})}{\sqrt{(\frac{1}{n}\sum_{i=1}^{n}(R(x_i) - \overline{R(x)})^2)(\frac{1}{n}\sum_{i=1}^{n}(R(y_i) - \overline{R(y)})^2)}}$$

wobei $\overline{R(x)}$ und $\overline{R(y)}$ die Mittelwerte aller Rangziffern bzgl. x bzw. y sind.

Seit Version 0.8.0 unterstützt die im Folgenden eingesetzte Python-Bibliothek *scipy* die korrekte Berechnung für Bindungen [com10]. Wie die meisten Korrelationskoeffizienten kann ρ Werte aus dem Intervall $[-1, 1]$ annehmen. Je höher der Absolutbetrag ist, desto größer ist die Korrelation zwischen den beiden betrachteten Variablen. Ein Wert von 1 bedeutet, dass zwischen den Rängen ein perfekter gleichläufiger monotoner Zusammenhang besteht, während $\rho = -1$ einen perfekten gegenläufigen monotonen Zusammenhang bedeutet. Gilt $\rho = 0$, so liegt kein monotoner Zusammenhang zwischen den Rängen vor [Cle14].
Liegt auch keine Ordinalskala, sondern nur eine Nominalskala vor, kann auf den χ^2-Test zurückgegriffen werden [Bou08], mit dem zwei nominalskalierte Variablen auf Unabhängigkeit getestet werden können. Der Wertebereich für χ^2 ist $\mathbb{R}^+$. Auch hier gilt, dass höhere Werte einen größeren Zusammenhang zwischen den Variablen bedeuten.
Um die erhaltenen Korrelationen und χ^2-Werte korrekt zu interpretieren, benötigt man jedoch zusätzliche Untersuchungen. Eine bekannte Möglichkeit ist die Berechnung des sogenannten *p-Werts*. Der p-Wert gibt an, wie hoch die Wahrscheinlichkeit ist, dass eine Teststatistik zweier Variablen mindestens genauso groß ist wie es in den vorliegenden

Werten der Variablen zu beobachten ist [WL16]. Im Falle einer Korrelationsanalyse bedeutet dies, dass der p-Wert einen Anhaltspunkt liefert, ob die gefundenen Korrelationen statistisch signifikant sind oder nicht. Hierzu wird die Nullhypothese aufgestellt, dass die betrachteten Variablen keinen Zusammenhang haben. Ist der p-Wert nun nahe 0, so sind die vorliegenden Daten mit der Nullhypothese nicht kompatibel und es kann angezweifelt werden, dass die Nullhypothese korrekt ist [WL16]. Folglich wird davon ausgegangen, dass ein statistisch signifikanter Zusammenhang vorliegt. Durch die Wahl eines Signifikanzniveaus wird eine Grenze festgelegt, ab welchem Wert die Nullhypothese verworfen werden kann. Für die statistische Analyse der Studierendendaten legen wir das häufig genutzte [WL16] Signifikanzniveau von $\alpha = 0.05$ fest. Die reine Angabe des p-Werts liefert jedoch noch keine ausreichende Aussage über die tatsächliche Bedeutung der gefundenen Korrelation, weshalb wir auch, wie von Wasserstein und Lazar [WL16] vorgeschlagen, die zu $\alpha = 0.05$ gehörenden Konfidenzintervalle der p-Werte betrachten, die mit Hilfe der Fisher-Transformation approximiert werden.
Bei dichotomen Items kann neben dem χ^2-Test auch der Spearman-Rangkorrelationskoeffizient berechnet werden.
Die einzelnen Skalenniveaus der Items des Fragebogens werden in Tabelle 3.1 angegeben. Erläuterungen der einzelnen Items können Anhang B entnommen werden.

Item (Kategorie)	**Skalenniveau**
Vorlesungsnote	Ordinal
bestanden/nicht best.	dichotom
AbiLand	Nominal
AbiBL	Nominal
sex	dichotom
LK_Mat	dichotom
LK_Phy	dichotom
LK_Inf	dichotom
GK_Inf	Nominal
Abinote	Ordinal
SK_Mat	Ordinal
SK_Inf	Ordinal
BM_Inf	Ordinal
BFI_K	Ordinal
LMI	Ordinal
Ktyp	Ordinal
Kurs_Inf	Intervall
Std_Inf	Intervall

Tabelle 3.1.: Skalenniveaus

3.2.3. Ergebnisse

Insgesamt lässt sich sagen, dass die gefundenen Korrelationen auf einem eher niedrigen Niveau liegen. Trotz allem lassen sich eindeutige Tendenzen in den Daten erkennen, von denen angenommen werden kann, dass sie statistisch signifikant sind. Im Folgenden werden die gefundenen Ergebnisse präsentiert.
Für die Korrelationsanalyse wurden mehrere Zielszenarien betrachtet, deren Korrelation mit den im Fragebogen erhobenen Items berechnet wurde. Zum einen wurde unterschieden zwischen der Betrachtung der erreichten Noten in den Vorlesungen und einer reinen Unterteilung der Daten in die Kategorien *bestanden*$(= 0)$ und *nicht bestanden* $(= 1)$. Dabei wurde nicht unterschieden, ob die Klausur in der Haupt- oder Nachklausur bestanden wurde, da für das formale Bestehen der Vorlesung beide Klausuren gleichwertig sind. Diese beiden groben Szenarien wurden in weitere Gruppen unterteilt.
Für die Analyse der Noten wurde aufgeteilt in:

1. Programmierung I (beste Note aus Haupt- und Nachklausur) $n_{n_p} = 274$

1a. Programmierung I (beste Note aus Haupt- und Nachklausur, standardisiert und skaliert) $n_{n_ps} = 274$
Da die Noten für die Vorlesung Programmierung I aus zwei verschiedenen Semestern mit eventuell unterschiedlichem Anforderungsniveau vorliegen, werden die Noten des Wintersemesters 2017/2018 bezüglich der Verteilung der Noten aus Wintersemester 2018/2019 standardisiert und skaliert. Dies führt zu einer besseren Vergleichbarkeit zwischen den Semestern.

2. MfI I (beste Note aus Haupt- und Nachklausur) $n_{n_m} = 124$

3. beste Note beider Vorlesungen $n_{n_b} = 277$

Bei der Unterscheidung zwischen *bestanden* und *nicht bestanden* wurde aufgeteilt in:

4. Programmierung I (bestes Ergebnis aus Haupt- und Nachklausur) $n_{b_p} = 274$

5. MfI I (bestes Ergebnis aus Haupt- und Nachklausur) $n_{b_m} = 124$

6. bestes Ergebnis beider Vorlesungen $n_{b_b} = 277$

7. Anzahl bestandener Vorlesungen $n_{b_v} = 154$

Das Item mit der höchsten Korrelation in allen Szenarien war die Durchschnittsnote der allgemeinen Hochschulreife. Die Abiturnote, sowie die Noten der Vorlesungen sind dabei nach den schulischen und universitären Benotungssystemen sortiert, sodass kleinere Werte einer besseren Note entsprechen. Dies erklärt die negative Korrelation in Szenario 7. bei welchem eine höhere Anzahl an bestandenen Klausuren besser ist.

Szenario	$\rho_{Abinote}$	p-Wert	Intervall
1.	0.51	$3.1e^{-19}$	$[0.41, 0.59]$
1a.	0.50	$4.0e^{-19}$	$[0.41, 0.59]$
2.	0.46	$7.6e^{-8}$	$[0.31, 0.59]$
3.	0.52	$1.3e^{-20}$	$[0.43, 0.60]$
4.	0.31	$1.8e^{-7}$	$[0.20, 0.41]$
5.	0.35	$8.6e^{-5}$	$[0.18, 0.49]$
6.	0.32	$3.8e^{-8}$	$[0.21, 0.42]$
7.	-0.36	$5.7e^{-6}$	$[-0.49, -0.21]$

Tabelle 3.2.: Korrelation Abiturnote

Die gefundenen Korrelationen sind auf dem 5%-Signifikanzniveau statistisch signifikant. Auch die zugehörigen 95%-Konfidenzintervalle deuten darauf hin, dass zwischen der Abiturnote und den Ergebnissen in Programmierung I und MfI I ein relevanter Zusammenhang besteht. Eine mögliche Erklärung, weshalb insbesondere die Abiturnote geeignet scheint, den Erfolg in den beiden Grundvorlesungen einzuschätzen ist, dass die Abiturnote einen sehr breiten Bereich abdeckt. Dies bestätigt das Ergebnis der Metaanalyse von Trapmann, Hell, Weigand und Schuler [THWS07], dass die Abiturnote ein valides Item zur Vorhersage des Studienerfolgs ist.
Betrachtet man die weiteren Korrelationen nach Spearman, so lassen sich zwischen den Szenarien mehrere Parallelen finden.
In Tabelle 3.3 findet sich eine Übersicht über alle Korrelationen für die $|\rho| \geq 0.2$ gilt. Alle aufgeführten Korrelationen sind statistisch signifikant mit $p < 0.05$. Keines der 95%-Konfidenzintervalle der angegebenen Korrelationen enthält 0, weshalb die Korrelationen als verlässlich angesehen werden können. Durch die natürliche Ordnung der Noten, bei denen kleine Werte besser sind, ergeben sich bis auf Szenario 7 hauptsächlich negative Korrelationen zu den einzelnen Attributen, bei denen hohe Werte besser sind.

	1.	1a.	2.	3.	4.	5.	6.	7.
SKMat_1	-0.34	-0.36	-0.37	-0.36	-0.26	-0.34	-0.27	0.29
SKMat_2	-0.29	-0.31	-0.31	-0.31	-	-0.22	-	-
SKMat_3	-0.28	-0.29	-0.29	-0.29	-0.20	-0.27	-0.21	0.24
SKMat_4	-0.25	-0.27	-0.35	-0.27	-	-0.28	-	0.22
mean_SKMat	-0.33	-0.35	-0.37	-0.35	-0.22	-0.32	-0.23	0.27
SKInf_1	-0.28	-0.25	-0.27	-0.27	-0.21	-0.29	-0.22	0.32
SKInf_4	-	-	-0.21	-	-	-	-	0.22
mean_SKInf	-0.20	-	-0.23	-0.21	-	-0.24	-	0.26
Kurs_Inf	-	-	-0.24	-	-	-	-	0.25
Std_Inf	-0.24*	-0.23*	-	-0.24*	-0.27*	-	-0.26*	0.23
Ktyp_exp	-0.24	-0.25	-0.27	-0.25	-	-	-	-
BFI_K_3	-0.23	-0.24	-	-0.23	-	-	-	-
BFI_K_7	-	-	0.21	-	-	-	-	-0.21
BFI_K_11	-	-	-	-	-	-0.21	-	-
mean_BFI_K_G	-0.22	-0.22	-	-0.23	-	-	-	-
BM_Inf_14	-	-	-0.20	-	-	-	-	-
BM_Inf_17	-0.22	-0.21	-	-0.22	-	-	-	0.22
BM_Inf_18	-	-	-	-	-	-0.23	-	-
LMI_3	-	-	-0.20	-0.21	-	-	-	0.22
LMI_6	-	-	-0.21	-	-	-	-	-
LMI_28	-	-	-	-	-	-	-	0.22

Tabelle 3.3.: Korrelationen, *: nur auf den Daten von WS18/19 berechnet

Betrachtet man die Korrelationen, fallen beim Vergleich zwischen den einzelnen Szenarien mehrere Gemeinsamkeiten auf. In allen Szenarien ist das Selbstkonzept in Mathematik ein wichtiger Teilbereich. Die erhaltenen Korrelationen lassen den Schluss zu, dass für die Vorlesungen Programmierung I und MfI I das Selbstkonzept in Mathematik eine höhere Korrelation mit dem Erfolg aufweist als das Selbstkonzept in Informatik. Während dieses Ergebnis für die Mathematikvorlesung zu erwarten war, ist dies für Programmierung I auf den ersten Blick erstaunlich. Hierfür sind zwei Erklärungen denkbar. Zum einen hat Programmierung I einen hohen Anteil an Themen mit mathematischem Charakter, beispielsweise in Form von Korrektheitsbeweisen [Smo12], zum anderen haben alle befragten Studierenden maximal 8 Jahre und im Mittel 2.4 Jahre Informatik in der Schule belegt, während Mathematik innerhalb Deutschlands zu den Pflichtfächern zählt. Somit ist für die Studierenden die Möglichkeit größer, im Laufe

der Schulzeit durch mehr Erfahrung ein realistischeres Selbstkonzept in Mathematik als in Informatik zu entwickeln. Jedoch zeigt sich auch, dass mehr Jahre Unterricht in Informatik eine signifikant messbare Korrelation zur Note in MfI I und damit auch zur Anzahl der bestandenen Klausuren aufweisen. Für Programmierung I kann diese Aussage auf Grundlage der vorhandenen Daten mit Korrelationen von $\rho = -0.075$ und einem p-Wert von $p = 0.22$ für Szenario 1, $\rho = -0.041$ und $p = 0.5$ für Szenario 1a, beziehungsweise $\rho = -0.096$ und $p = 0.11$ für Szenario 4, nicht getroffen werden. Jedoch findet sich eine signifikante Korrelation der Leistung in Programmierung I zu der Anzahl an Stunden pro Woche die die Studierenden Informatikunterricht in der Schule hatten. Dieses Item liegt jedoch nur für die 154 Studierenden der Befragung im Wintersemester 2018/2019 vor. Für den späteren Einsatz mit Maschinellem Lernen ist es also nur für die Szenarien 2., 5. und 7. geeignet.
Die anforderungsbasierte Kongruenz kann für die Vorhersage der Noten sowohl in MfI I als auch Programmierung I relevant sein und ist dabei ein Prädiktor mit einer höheren Korrelation als die psychosoziale Kongruenz.
Die erfassten Persönlichkeitsmerkmale sind als Kriterien für die Vorhersage des Erfolgs in Programmierung I und MfI I zum Großteil nicht geeignet. Lediglich die Gewissenhaftigkeit weist eine nennenswerte Korrelation zu den Ergebnissen in Programmierung I auf. Dies deckt sich auch in der Höhe des beobachteten Effekts mit den Ergebnissen von O'Connor und Paunonen [OP07] bezüglich der Korrelation zwischen Gewissenhaftigkeit und Studienerfolg. Für MfI I ist eine Korrelation zu finden mit der Einschätzung der Studierenden, ob sie sich selbst als wortkarg ansehen und weniger leicht anderen Menschen Vertrauen schenken bzw. nicht an das Gute im Menschen glauben.
Auch die Berufsmotivation für Informatik bietet kaum Kriterien, die sich für die Vorhersage der Noten in Programmierung I und MfI I sinnvoll nutzen lassen. Der Tätigkeitsanreiz in andere Computersysteme einzudringen, sowie eine höhere Selbsteinschätzung der informatischen Kompetenzen zeigen eine Korrelation mit der Note in MfI I, während für Programmierung I und in der Folge für die Anzahl der bestandenen Klausuren eine hohe Beurteilung der eigenen Programmierkenntnisse eine signifikante Korrelation aufweisen.
Das Leistungsmotivationsinventar scheint allgemein kein geeignetes Werkzeug für die Vorhersage des Erfolgs in Programmierung I und MfI I zu sein. Lediglich die Bevorzugung von eher schwierigen Aufgaben korreliert mit der Note in MfI I.
Betrachtet man die Fragebogenitems auf Nominalskalenniveau so fällt auf, dass auf Grundlage des χ^2-Tests einzig für Szenario 5 ein statistisch signifikanter Zusammenhang mit dem Besuch des Leistungskurs Mathematik anzunehmen ist ($p = 0.049$). Für alle anderen Szenarien finden sich keine statistisch signifikanten Zusammenhänge mit einem nominalskalierten Item. Untersucht man den gefundenen Zusammenhang zwischen dem

Bestehen der MfI I Klausur und dem Besuch des Leistungskurs Mathematik genauer, so kann eine Korrelation von $\rho = 0.28$ mit einem 95%-Konfidenzintervall von $[0.11, 0.43]$ beobachtet werden.

3.3. Einsatz Maschinellen Lernens

Nachdem die statistische Analyse abgeschlossen ist, werden im Folgenden Vorhersagen mittels Entscheidungsbäumen auf den Daten durchgeführt. Um die Entscheidungsbäume zu erstellen, wurde die bereits erwähnte Python-Bibliothek *scikit-learn*[dev19] genutzt, welche Entscheidungsbäume mit Hilfe des CART-Algorithmus erstellt.
Die oben angeführten Szenarien 1 bis 3 sind natürlicherweise Regressionsprobleme, da Noten vorhergesagt werden, die auf einer kontinuierlichen Punkteverteilung basieren. Jedoch können die elf Einzelnoten der Szenarien 1, 2 und 3 auch als elf Klassen betrachtet werden, weshalb ebenfalls eine Vorhersage mit Hilfe eines Klassifikationsbaums denkbar ist. Szenario 1a wird aufgrund der Vielzahl an entstehenden Klassen im Verhältnis zur Anzahl an Datensätzen nur als Regressionsproblem betrachtet. Szenarien 4 bis 7 sind eindeutig Klassifikationsprobleme.
Für eine korrekte Interpretation der erhaltenen Genauigkeiten bei Klassifikationsproblemen ist es nötig, den Anteil der häufigsten Klasse an allen Datensätzen zu kennen. Hierüber gibt Tabelle 3.4 eine Übersicht. Die erreichten Noten werden zwischen 100 für die Note 1.0 und 500 für die Note 5.0 angegeben.

Szenario	häufigste Klasse	Anteil
1.	500	28.47%
2.	500	39.52%
3.	500	28.16%
4.	bestanden	71.53%
5.	bestanden	60.48%
6.	bestanden	71.84%
7.	2 Klausuren	46.75%

Tabelle 3.4.: Verhältnisse der Klassen

3.3.1. Basisbäume

Um einen ersten Überblick über die Performanz zu erhalten, sind in den nachfolgenden Tabellen für alle Szenarien die Genauigkeiten auf den Trainings- und Testdaten für klassifizierende Entscheidungsbäume zu finden, bei denen die Hyperparameter nicht angepasst wurden. Es wurden hierfür alle zur Verfügung stehenden Attribute genutzt und 30% der Daten als Testdatensatz zurückgehalten. Für die Szenarien 1 bis 3 sind zusätzlich der mittlere absolute Fehler und die Wurzel des mittleren quadratischen Fehlers im Falle eines Regressionsbaums angegeben. Für die Szenarien mit binären Labels finden sich die AUC in Bezug auf die ROC, sowie der F1-Score. Zu allen angegebenen Performanzkriterien wird auch der Mittelwert der erhaltenen Werte bei 5-facher Cross-Validation auf den Trainingsdaten angegeben.

Szenario	Genauigkeit Training	Genauigkeit Test	Genauigkeit CV
1.	95.81%	20.48%	24.21%
2.	100%	23.68%	14.70%
3.	97.41%	19.05%	24.15%
4.	96.86%	69.88%	63.86%
5.	100%	44.74%	67.32%
6.	98.45%	64.29%	68.88%
7.	100%	51.06%	40.48%

Tabelle 3.5.: Genauigkeiten Basisbaum

Szenario	RMSE Training	RMSE Test	RMSE CV	MAE Training	MAE Test	MAE CV
1.	35.32	164.55	187.02	9.58	120.12	147.63
1a.	35.32	145.56	173.94	9.58	114.65	137.21
2.	0.0	146.99	158.40	0.0	112.11	121.15
3.	17.91	177.71	170.19	3.83	134.40	128.96

Tabelle 3.6.: Fehler Basisbaum

Szenario	AUC Training	AUC Test	AUC CV	F1 Training	F1 Test	F1 CV
4.	0.9976	0.6356	0.5609	94.12%	48.98%	37.88%
5.	1.0	0.4137	0.6765	100%	27.59%	62.23%
6.	0.9994	0.5729	0.6210	97.09%	40.00%	43.30%

Tabelle 3.7.: AUC und F1-Score Basisbaum

Erwartungsgemäß liegt bei allen Bäumen ein deutliches Overfitting vor, sodass die Vorhersagen auf den Trainingsdaten annähernd perfekt sind, während die Vorhersagen auf den Testdaten sehr schlecht sind. Insbesondere unter Berücksichtigung der Klassenverteilungen sind die erzielten Genauigkeiten unzureichend.
Um die Ergebnisse zu verbessern, gibt es die in 2.4.2 erwähnten Methoden, wobei in diesem Fall lediglich das Anpassen der Hyperparameter, das Anpassen der Attribute und die Rauschreduktion in Frage kommen, da das Modell vorgegeben ist und es zum jetzigen Zeitpunkt nicht möglich ist, neue Daten zu sammeln.
Die Rauschreduktion durchzuführen ist jedoch nicht sinnvoll. Einzelne Datensätze auszuschließen bedeutet, die tatsächlich aufgetretenen Ergebnisse eines Studierenden nicht zu berücksichtigen. Dies kann nicht genauso behandelt werden wie beispielsweise das Rauschen in den Signalen eines Sensors, da hier keine Messfehler im eigentlichen Sinne vorliegen.

3.3.2. Feature Engineering

Zunächst wird der Einfluss des Feature Engineering auf die Ergebnisse betrachtet. Dabei werden im Einzelnen die folgenden Methoden betrachtet:

- Recursive Feature Elimination inklusive Cross-Validation zur Bestimmung der optimalen Attributanzahl
- Recursive Feature Elimination mit festgelegter Anzahl an Attributen
- Betrachtung der Attribute mit den höchsten Korrelationen

RFE mit CV

Um bei einer Recursive Feature Elimination Ergebnisse zu erzielen die für das geplante Modell verwertbar sind, ist es wichtig für die Attributauswahl ein dem geplanten Modell ähnliches aber trotz allem stabiles Modell zu wählen. Sollen Entscheidungsbäume zur Vorhersage eingesetzt werden, bieten sich also Ensemble-Methoden, wie Random Forest, an, die auf Entscheidungsbäumen basieren. Mit Hilfe von Random Forests, bestehend aus 100 Entscheidungsbäumen, und 5-facher Cross-Validation wurden automatisiert die folgenden optimalen Anzahlen an Attributen ermittelt.

Szenario	Genauigkeit	F1	AUC	RMSE	MAE
1.	7	-	-	62	7
1a.	-	-	-	53	8
2.	70	-	-	24	4
3.	84	-	-	35	38
4.	47	6	5	-	-
5.	66	66	81	-	-
6.	79	3	12	-	-
7.	85	-	-	-	-

Tabelle 3.8.: optimale Anzahl an Attributen

Die Ergebnisse der Vorhersagen bei Nutzung der Recursive Feature Elimination mit Cross-Validation stellen sich wie in den folgenden Tabellen dar. Die Angaben der Veränderungen mit Hilfe der Pfeile beziehen sich immer auf die Ergebnisse der Basisbäume.

Szenario	Genauigkeit Training	Genauigkeit Test	Genauigkeit CV
1.	95.81%	21.69% ↑	25.84% ↑
2.	100%	26.32% ↑	21.36% ↑
3.	97.41%	19.05%	24.15%
4.	96.86%	72.29% ↑	65.44% ↑
5.	100%	50.00% ↑	62.88% ↓
6.	98.45%	63.10% ↓	72.00% ↑
7.	100%	51.06%	40.48%

Tabelle 3.9.: Genauigkeiten RFE mit Cross-Validation

Szenario	RMSE Training	RMSE Test	RMSE CV	MAE Training	MAE Test	MAE CV
1.	35.32	171.69 ↑	184.64 ↓	9.58	139.64 ↑	128.67 ↓
1a.	35.32	139.86 ↓	161.85 ↓	9.58	106.54 ↓	115.45 ↓
2.	0.0	129.03 ↓	123.93 ↓	0.0	123.95 ↑	111.80 ↓
3.	17.91	184.16 ↑	158.74 ↓	3.83	145.60 ↑	120.83 ↓

Tabelle 3.10.: Fehler RFE mit Cross-Validation

Szenario	AUC Training	AUC Test	AUC CV	F1 Training	F1 Test	F1 CV
4.	0.9976	0.5431 ↓	0.6034 ↑	94.12%	30.77% ↓	41.21% ↑
5.	1.0	0.4494 ↑	0.6375 ↓	100%	38.71% ↑	57.46% ↓
6.	0.9994	0.6454 ↑	0.5814 ↓	97.09%	46.43% ↑	50.35% ↑

Tabelle 3.11.: AUC und F1-Score RFE mit Cross-Validation

Die Ergebnisse sind im Vergleich zu den Basisergebnissen im Allgemeinen durch die Anwendung von RFE mit Cross-Validation nicht verbessert worden. Für die Genauigkeit ergeben sich kaum Veränderungen. Lediglich in Szenario 2 sind die Ergebnisse bei der Cross-Validation etwas besser. Im Falle der Regressionsbäume sind die Fehler auf den Testdaten außer für Szenarien 1a und 2 größer geworden, wohingegen die Cross-Validation-Werte durchweg besser sind. Für die AUC scheint die RFE weder eindeutig positiv noch eindeutig negativ zu sein, wohingegen der F1-Score von der Attributauswahl per RFE und Cross-Validation profitiert. Zu beachten ist jedoch, dass in einigen Szenarien die Anzahl der ausgewählten Attribute weiterhin sehr groß ist.

RFE ohne CV

Prüft man die Graphen der CV-Scores in Anhang C, kann man erkennen, dass für alle Szenarien - 1 bei RMSE, 1a bei RMSE, 2 bei Genauigkeit, 3 bei Genauigkeit, MAE und RMSE, 4 bei Genauigkeit, 5 bei Genauigkeit, AUC und F1, 6 bei Genauigkeit und AUC, sowie 7 bei Genauigkeit - bereits für deutlich kleinere Anzahlen an Attributen ein ähnlich guter CV-Score erreicht werden kann. Daher wird in diesen Fällen die RFE mit einer manuell festgelegten Anzahl an Attributen durchgeführt. Die Anzahl der festgelegten Attribute ist dabei:

Szenario	Genauigkeit	F1	AUC	RMSE	MAE
1.	7	-	-	8	7
1a.	-	-	-	8	7
2.	48	-	-	24	4
3.	6	-	-	27	13
4.	6	6	5	-	-
5.	35	35	66	-	-
6.	28	3	4	-	-
7.	59	-	-	-	-

Tabelle 3.12.: manuell festgelegte Anzahl an Attributen

Führt man die Performanzanalyse mit diesen Attributen durch, erhält man folgende Ergebnisse, die wiederum in Zusammenhang mit den Basisbäumen gesetzt werden:

Szenario	Genauigkeit Training	Genauigkeit Test	Genauigkeit CV
1.	95.81%	21.69% ↑	25.84% ↑
2.	100%	23.68%	25.92% ↑
3.	97.41%	21.43% ↑	26.29% ↑
4.	96.86%	67.47% ↓	61.33% ↓
5.	100%	52.63% ↑	69.67% ↑
6.	98.45%	61.90% ↓	69.92% ↑
7.	100%	55.32% ↑	42.89% ↑

Tabelle 3.13.: Genauigkeiten RFE feste Anzahl

Szenario	RMSE Training	RMSE Test	RMSE CV	MAE Training	MAE Test	MAE CV
1.	35.32	163.22 ↓	165.04 ↓	9.58	139.64 ↑	128.67 ↓
1a.	35.32	140.97 ↓	149.83 ↓	9.58	106.54 ↓	115.45 ↓
2.	0.0	129.03 ↓	123.93 ↓	0.0	123.95 ↑	111.80 ↓
3.	17.91	175.85 ↓	147.58 ↓	3.83	122.26 ↓	111.18 ↓

Tabelle 3.14.: Fehler RFE feste Anzahl

Szenario	AUC Training	AUC Test	AUC CV	F1 Training	F1 Test	F1 CV
4.	0.9976	0.5431 ↓	0.6034 ↑	94.12%	30.77% ↓	41.21% ↑
5.	1.0	0.4851 ↑	0.6245 ↓	100%	40.00% ↑	63.43% ↑
6.	0.9994	0.5129 ↓	0.6595 ↑	97.09%	46.43% ↑	50.35% ↑

Tabelle 3.15.: AUC und F1-Score RFE feste Anzahl

Es lässt sich feststellen, dass auch durch eine niedrigere Anzahl an Attributen die Performanz der Entscheidungsbäume insgesamt nicht deutlich verbessert werden konnte. Die erzielten Genauigkeiten sind sehr ähnlich zur RFE mit CV. Bei Szenario 3 ist jedoch eine deutliche und bei Szenario 7 eine leichte Verbesserung ersichtlich. Insbesondere bei der Betrachtung des RMSE und MAE ist Szenario 3 deutlich verbessert worden. Teilweise sind die erzielten Ergebnisse jedoch schlechter geworden, wie beispielsweise bei der Genauigkeit in Szenario 4. Für den F1-Score ist die Verringerung der Attribute in Szenario 5 vorteilhaft, die AUC profitiert nur auf den Testdaten, nicht jedoch bei der Cross-Validation.
Die ausgewählten Attribute zeigen große Überschneidungen mit den am höchsten

korrelierten Attributen. Die RFE auf Basis von Random Forests favorisiert jedoch deutlich die jeweiligen Skalenmittelwerte und legt bei den Szenarien 2, 5, 6 und 7 - also den Szenarien mit MfI I - mehr Wert auf das Leistungsmotivationsinventar und die Persönlichkeitsmerkmale, hier insbesondere das Item BFI_K_11 „bin eher der ‚stille Typ', wortkarg".

höchste Korrelationen

Trotz der Überschneidungen existieren auch einige Unterschiede, weshalb eine gesonderte Betrachtung der Entscheidungsbäume auf den in Abschnitt 3.2.2 gefundenen Attributen sinnvoll erscheint.

Szenario	Genauigkeit Training	Genauigkeit Test	Genauigkeit CV
1.	95.81%	25.30% ↑	27.99% ↑
2.	100%	31.58% ↑	27.69% ↑
3.	97.41%	22.62% ↑	24.64% ↑
4.	96.34% ↓	69.88%	65.49% ↑
5.	100%	71.05% ↑	62.81% ↓
6.	97.93% ↓	63.10% ↓	70.45% ↑
7.	100%	53.19% ↑	40.30% ↓

Tabelle 3.16.: Genauigkeiten höchste Korrelationen

Szenario	RMSE Training	RMSE Test	RMSE CV	MAE Training	MAE Test	MAE CV
1.	35.32	174.25 ↑	165.21 ↓	9.58	131.99 ↑	124.82 ↓
1a.	35.32	136.18 ↓	150.88 ↓	9.58	100.86 ↓	114.38 ↓
2.	0.0	142.63 ↓	138.77 ↓	0.0	105.79 ↓	96.50 ↓
3.	17.91	175.13 ↓	159.36 ↓	3.83	130.24 ↓	122.42 ↓

Tabelle 3.17.: Fehler höchste Korrelationen

Szenario	AUC Training	AUC Test	AUC CV	F1 Training	F1 Test	F1 CV
4.	0.9967 ↓	0.5763 ↓	0.5977 ↑	93.07% ↓	41.86% ↓	44.07% ↑
5.	1.0	0.6964 ↑	0.6236 ↓	100%	62.09% ↑	56.54% ↓
6.	0.9989 ↓	0.5393 ↓	0.6043 ↓	96.08% ↓	31.11% ↓	40.99% ↓

Tabelle 3.18.: AUC und F1-Score höchste Korrelationen

Die Verwendung der Attribute mit den höchsten Korrelationen trägt ebenfalls nicht in jedem Szenario zur deutlichen Verbesserung der Vorhersagen bei. Lediglich in den

Szenarien 1a, 2 und 5 kann eine eindeutige Verbesserung der Ergebnisse festgestellt werden. Im Gegenzug verschlechtern sich die Ergebnisse in Szenario 4 und 6.

Zusammenfassung

Zusammenfassend gilt, dass die Vorhersagequalität allein durch das Feature Engineering nicht wesentlich verbessert werden konnte. Für die einzelnen Szenarien haben sich jeweils verschiedene Methoden zur Attributauswahl als am besten geeignet herausgestellt.
Unter Berücksichtigung aller Qualitätskriterien sind tendenziell die besten Methoden:

- **Szenario 1** Attribute mit den höchsten Korrelationen
- **Szenario 1a** Attribute mit den höchsten Korrelationen
- **Szenario 2** Attribute mit den höchsten Korrelationen
- **Szenario 3** RFE ohne CV
- **Szenario 4** Basisbaum
- **Szenario 5** Attribute mit den höchsten Korrelationen
- **Szenario 6** RFE mit CV
- **Szenario 7** RFE ohne CV

Insgesamt muss jedoch beachtet werden, dass die Ergebnisse nicht bedeutend besser oder sogar schlechter sind, als für einen naiven Prädiktor der jeweils die häufigste Klasse vorhersagt.

3.3.3. Anpassen der Hyperparameter

Neben dem Feature Engineering bietet das Anpassen der in 2.5.1 erläuterten Hyperparameter eine weitere wichtige Möglichkeit, auf die Ergebnisse der Entscheidungsbäume Einfluss zu nehmen. Um die beste Kombination an Hyperparametern zu finden, wird die Funktion *GridSearchCV* [dev19, S. 2094] aus der *scikit-learn*-Bibliothek genutzt. Hierdurch können unterschiedliche vorgegebene Kombinationen an Hyperparametern im Rahmen einer Cross-Validation überprüft werden.
Für die einzelnen Szenarien werden als Grundlage jeweils sowohl alle verfügbaren Attribute, als auch die in 3.3.2 gefundenen besten Attributkombinationen genutzt. Zusätzlich zu den bereits beschriebenen Hyperparametern wird für jedes Szenario abhängig vom verwendeten Performanzkriterium auch das beste Splitkriterium mit Hilfe von *GridSearchCV* bestimmt.

Verwendung aller Attribute

Unter Verwendung aller Attribute werden die folgenden Ergebnisse erzielt:

Szenario	Genauigkeit Training	Genauigkeit Test	Genauigkeit CV
1.	33.51% ↓	36.14% ↑	33.04% ↑
2.	40.70% ↓	36.84% ↑	42.00% ↑
3.	38.86% ↓	29.76% ↑	35.30% ↑
4.	76.96% ↓	71.08% ↑	70.24% ↑
5.	94.19% ↓	52.63% ↑	85.10% ↑
6.	93.78% ↓	65.48% ↑	75.09% ↑
7.	58.88% ↓	53.19% ↑	57.97% ↑

Tabelle 3.19.: Genauigkeiten mit Anpassen der Hyperparameter und allen Attributen

Szenario	RMSE Training	RMSE Test	RMSE CV	MAE Training	MAE Test	MAE CV
1.	129.26 ↑	144.84 ↓	138.92 ↓	97.49 ↑	102.29 ↓	112.60 ↓
1a.	110.60 ↑	139.86 ↓	125.03 ↓	72.62 ↑	119.25 ↑	104.82 ↓
2.	97.03 ↑	109.90 ↓	108.21 ↓	44.19 ↑	93.95 ↓	81.43 ↓
3.	114.21 ↑	153.02 ↓	130.41 ↓	86.48 ↑	125.12 ↓	98.52 ↓

Tabelle 3.20.: Fehler mit Anpassen der Hyperparameter und allen Attributen

Szenario	AUC Training	AUC Test	AUC CV	F1 Training	F1 Test	F1 CV
4.	0.8491 ↓	0.6540 ↑	0.6707 ↑	40.54% ↓	40.00% ↓	42.85% ↑
5.	0.9457 ↓	0.5015 ↑	0.8316 ↑	92.31% ↓	40.00% ↑	80.62% ↑
6.	0.8869 ↓	0.5295 ↓	0.7019 ↑	87.50% ↓	29.27% ↓	53.61% ↑

Tabelle 3.21.: AUC und F1-Score mit Anpassen der Hyperparameter und allen Attributen

Es ist deutlich zu erkennen, dass in allen Szenarien das Overfitting reduziert wurde. Die Fähigkeit der Generalisierung der Entscheidungsbäume ist verbessert worden. Interessant ist jedoch Szenario 2. Hier ist die Performanz sowohl der Regressionsbäume als auch der Klassifikationsbäume wesentlich besser als die der Basisbäume. Auch in Szenario 5 haben sich insbesondere die Cross-Validation-Werte deutlich verbessert. Dieses Verhalten kann unter anderem an der gewählten Aufteilung in Trainings- und Testdaten liegen. Die naheliegende Idee, durch wiederholte Cross-Validation auf verschiedenen initialen Aufteilungen einen Mittelwert zu bilden, ist nicht zu empfehlen [VB12]. Es ist nicht anzunehmen, dass die erhaltenen Mittelwerte die wahre Genauigkeit

besser abbilden.
Stattdessen bietet es sich an, analog zu den Korrelationen, Konfidenzintervalle zu betrachten. Um die Konfidenzintervalle zu berechnen, wird Bootstrapping [Efr79] genutzt. Beim Bootstrapping wird durch mehrfaches Ziehen von Datensätzen mit Zurücklegen aus den vorhandenen Daten die betrachtete Statistik, wie beispielsweise die Genauigkeit oder der RMSE, berechnet. Dieses Verfahren wird mehrfach wiederholt um - bei einem 95%-Konfidenzniveau - das 97.5- und 2.5-Perzentil zu bestimmen und daraus das Konfidenzintervall für die jeweilige Statistik zu bilden.
Die mit 1000-fachem Ziehen berechneten 95%-Konfidenzintervalle für die oben angegebenen Statistiken sind:

Szenario	Genauigkeit	RMSE	MAE	AUC	F1
1.	$[18.04\%, 35.08\%]$	$[137.53, 168.54]$	$[107.45, 147.66]$	-	-
1a.	-	$[122.14, 150.72]$	$[100.20, 132.21]$	-	-
2.	$[14.75\%, 43.55\%]$	$[104.00, 155.30]$	$[84.52, 138.01]$	-	-
3.	$[17.78\%, 35.56\%]$	$[133.41, 166.41]$	$[106.99, 147.97]$	-	-
4.	$[54.47\%, 75.18\%]$	-	-	$[0.4473, 0.6351]$	$[8.00\%, 46.58\%]$
5.	$[45.90\%, 70.31\%]$	-	-	$[0.4424, 0.7043]$	$[25.62\%, 65.31\%]$
6.	$[54.42\%, 71.33\%]$	-	-	$[0.4464, 0.6269]$	$[14.70\%, 45.34\%]$
7.	$[29.33\%, 56.76\%]$	-	-	-	-

Tabelle 3.22.: 95%-Konfidenzintervalle mit Anpassen der Hyperparameter und allen Attributen

Das Ergebnis aus [VB12], dass die Cross-Validation die Statistiken eher optimistisch einschätzt, scheint sich zu bestätigen, da der CV-Wert meistens an der (positiven) Grenze des Intervalls oder sogar außerhalb des Intervalls liegt.
Die jeweils mit Cross-Validation gefundenen Werte der Hyperparameter können Anhang D entnommen werden.

Verwendung ausgewählter Attribute

Um die Wirksamkeit des Feature Engineering beurteilen zu können, werden auch für die in Abschnitt 3.3.2 gefundenen besten Methoden der Attributauswahl für jedes Szenario die Hyperparameter optimiert und die Konfidenzintervalle berechnet. Auch hier sind die einzelnen Hyperparameter im Anhang D zu finden.

Szenario	Genauigkeit Training	Genauigkeit Test	Genauigkeit CV
1.	38.74% ↓	37.35% ↑	36.16% ↑
2.	47.67% ↓	36.84% ↑	44.29% ↑
3.	40.93% ↓	29.76% ↑	38.84% ↑
4.	76.96% ↓	71.08% ↑	70.24% ↑
5.	76.74% ↓	63.16% ↑	73.40% ↑
6.	94.82% ↓	57.14% ↓	75.68% ↑
7.	75.70% ↓	46.81% ↓	52.57% ↑

Tabelle 3.23.: Genauigkeiten mit Anpassen der Hyperparameter und ausgewählten Attributen

Szenario	RMSE Training	RMSE Test	RMSE CV	MAE Training	MAE Test	MAE CV
1.	110.90 ↑	149.83 ↓	136.22 ↓	93.61 ↑	105.90 ↓	105.93 ↓
1a.	111.57 ↑	120.82 ↓	121.10 ↓	75.07 ↑	96.92 ↓	102.33 ↓
2.	84.22 ↑	111.90 ↓	99.00 ↓	45.93 ↑	82.76 ↓	67.10 ↓
3.	108.02 ↑	139.01 ↓	132.08 ↓	67.36 ↑	132.26 ↓	97.60 ↓

Tabelle 3.24.: Fehler mit Anpassen der Hyperparameter und ausgewählten Attributen

Szenario	AUC Training	AUC Test	AUC CV	F1 Training	F1 Test	F1 CV
4.	0.8491 ↓	0.6540 ↑	0.6707 ↑	40.54% ↓	40.00% ↓	42.85% ↑
5.	0.8849 ↓	0.6280 ↑	0.7649 ↑	72.97% ↓	56.25% ↑	68.62% ↑
6.	0.8595 ↓	0.5420 ↓	0.7671 ↑	78.26% ↓	34.04% ↓	64.18% ↑

Tabelle 3.25.: AUC und F1-Score mit Anpassen der Hyperparameter und ausgewählten Attributen

Szenario	Genauigkeit	RMSE	MAE	AUC	F1
1.	[19.72%, 37.50%]	[133.17, 166.83]	[102.65, 139.56]	-	-
1a.	-	[111.34, 136.48]	[95.48, 123.90]	-	-
2.	[19.04%, 46.77%]	[110.08, 166.79]	[76.94, 130.47]	-	-
3.	[19.12%, 36.00%]	[129.87, 161.62]	[108.17, 146.09]	-	-
4.	[55.12%, 75.59%]	-	-	[0.4455, 0.6380]	[7.84%, 45.95%]
5.	[50.85%, 75.41%]	-	-	[0.4930, 0.7343]	[28.55%, 69.57%]
6.	[53.85%, 70.68%]	-	-	[0.4679, 0.6465]	[14.55%, 47.31%]
7.	[32.87%, 56.41%]	-	-	-	-

Tabelle 3.26.: 95%-Konfidenzintervalle mit Anpassen der Hyperparameter und ausgewählten Attributen

Die 95%-Konfidenzintervalle sind im Vergleich zu der Berücksichtigung aller Attribute sehr ähnlich. Es kann für kein Szenario eine echte Verbesserung festgestellt werden. Ein Teil der Unterschiede ist lediglich auf unterschiedliche Stichproben beim Bootstrapping zurückzuführen, was anhand von Szenario 4 deutlich wird, da hier die selben Attribute genutzt wurden.
Insgesamt ergeben sich also nur minimale Unterschiede bei Verwendung von Feature Engineering. Da die Ergebnisse weiterhin nicht gut sind, wird im nächsten Abschnitt auf eine andere Art der Optimierung eingegangen.

3.3.4. Pruning

(Post-)Pruning stellt eine weitere Möglichkeit dar, die Performanz von Entscheidungsbäumen, bei denen Overfitting vorliegt, zu verbessern. Im Folgenden werden die Ergebnisse dargestellt, welche durch die in 2.7 vorgestellten Methoden auf den Studierendendaten erreicht werden können.

Basisbäume

Um die Ergebnisse des REP und CVP zu erhalten, wird das Pruning jeweils mit dem betrachteten Performanzkriterium durchgeführt. Für die angegebenen AUC-Werte werden also die Entscheidungsbäume anhand der jeweiligen AUC-Werte gepruned, usw. Wie in 2.7 beschrieben, werden zusätzlich zu den Trainings- und Testdaten auch Pruningdaten benötigt. Für die Testdaten werden 30% der Daten verwendet. Die restlichen 70% der Daten werden erneut im Verhältnis 70/30 in Trainings- und Pruningdaten aufgespalten.
Die Vergleichbarkeit mit den Basisbäumen aus 3.3.1 ist aufgrund der unterschiedlichen Aufteilung der Daten nicht gewährleistet, weshalb zunächst ein Überblick über die vollständig aufgebauten Entscheidungsbäume gegeben wird.

Szenario	Genauigkeit Training	Genauigkeit Pruning	Genauigkeit Test
1.	100%	21.05%	19.85%
2.	100%	36.00%	24.19%
3.	97.64%	28.07%	11.59%
4.	100%	56.14%	59.85%
5.	100%	64.00%	59.68%
6.	99.21%	43.86%	59.70%
7.	100%	46.88%	41.77%

Tabelle 3.27.: Genauigkeiten ohne Pruning auf allen Attributen

Szenario	RMSE Training	RMSE Pruning	RMSE Test	MAE Training	MAE Pruning	MAE Test
1.	0.0	170.76	186.48	0.0	129.12	141.25
1a.	0.0	153.49	168.21	0.0	130.63	130.53
2.	0.0	114.16	154.85	0.0	73.6	116.61
3.	16.34	172.44	166.92	3.15	139.12	117.61

Tabelle 3.28.: Fehler ohne Pruning auf allen Attributen

Szenario	AUC Training	AUC Pruning	AUC Test	F1 Training	F1 Pruning	F1 Test
4.	1.0	0.5372	0.5764	100%	28.57%	41.76%
5.	1.0	0.6494	0.5973	100%	64.00%	54.55%
6.	0.9998	0.3953	0.4710	98.04%	23.81%	18.18%

Tabelle 3.29.: AUC und F1-Score ohne Pruning auf allen Attributen

Auch für die Einschätzung der Effektivität des Prunings in Kombination mit Feature Engineering ist zunächst die Angabe der Resultate der vollständigen Entscheidungsbäume nötig.

Szenario	Genauigkeit Training	Genauigkeit Pruning	Genauigkeit Test
1.	100%	15.79%	19.85%
2.	100%	32.00%	25.81%
3.	97.64%	19.30%	18.84%
4.	100%	56.14%	59.85%
5.	100%	56.00%	58.06%
6.	99.21%	61.40%	67.91%
7.	100%	59.38%	41.77%

Tabelle 3.30.: Genauigkeiten ohne Pruning auf ausgewählten Attributen

Szenario	RMSE Training	RMSE Pruning	RMSE Test	MAE Training	MAE Pruning	MAE Test
1.	0.0	158.44	174.59	0.0	120.70	134.26
1a.	0.0	146.63	161.92	0.0	121.90	118.60
2.	0.0	133.09	157.17	0.0	96.80	118.06
3.	16.34	172.44	166.92	3.15	139.12	117.61

Tabelle 3.31.: Fehler ohne Pruning auf ausgewählten Attributen

Szenario	AUC Training	AUC Pruning	AUC Test	F1 Training	F1 Pruning	F1 Test
4.	1.0	0.5372	0.5764	100%	28.57%	41.76%
5.	1.0	0.5682	0.5838	100%	56.00%	53.57%
6.	0.9998	0.5189	0.6029	98.04%	26.67%	45.57%

Tabelle 3.32.: AUC und F1-Score ohne Pruning auf ausgewählten Attributen

REP

REP ohne Feature Engineering Mit Reduced Error Pruning ergeben sich die folgenden Ergebnisse.

Szenario	Genauigkeit Training	Genauigkeit Pruning	Genauigkeit Test
1.	62.30% ↓	21.05%	20.59% ↑
2.	65.45% ↓	36.00%	32.26% ↑
3.	37.80% ↓	38.60% ↑	32.61% ↑
4.	84.17% ↓	71.93% ↑	68.94% ↑
5.	90.91% ↓	68.00% ↑	62.90% ↑
6.	79.53% ↓	64.91% ↑	69.40% ↑
7.	81.16% ↓	50.00% ↑	45.57% ↑

Tabelle 3.33.: Genauigkeiten mit REP auf allen Attributen

Szenario	RMSE Training	RMSE Pruning	RMSE Test	MAE Training	MAE Pruning	MAE Test
1.	127.20 ↑	142.99 ↓	147.48 ↓	33.06 ↑	117.76 ↓	137.94 ↓
1a.	114.37 ↑	107.57 ↓	126.99 ↓	86.12 ↑	105.39 ↓	105.37 ↓
2.	88.83 ↑	102.89 ↓	125.30 ↓	19.92 ↑	69.40 ↑	105.13 ↓
3.	71.50 ↑	146.15 ↓	159.32 ↓	49.75 ↑	111.40 ↓	122.96 ↑

Tabelle 3.34.: Fehler mit REP auf allen Attributen

Szenario	AUC Training	AUC Pruning	AUC Test	F1 Training	F1 Pruning	F1 Test
4.	0.9884 ↓	0.5904 ↑	0.5584 ↓	84.37% ↓	31.25% ↑	30.77% ↓
5.	0.9006 ↓	0.7532 ↑	0.6330 ↑	86.49% ↓	71.43% ↑	58.18% ↑
6.	0.9067 ↓	0.5541 ↑	0.5538 ↑	60.32% ↓	31.58% ↑	25.35% ↑

Tabelle 3.35.: AUC und F1-Score mit REP auf allen Attributen

Es fällt auf, dass zwar in allen Szenarien das Overfitting, teilweise deutlich, reduziert werden konnte, jedoch sind die Ergebnisse nicht durchgehend besser als ohne Pruning. Profitieren können die Regressionsbäume, sowie die Klassifikationsbäume der Szenarien 2, 3, 4, 6 und 7
Bei den AUC- und F1-Werten in Szenario 4 wird ein Nachteil des REP, das Overpruning, deutlich. Zwar wird die Performanz auf den Pruningdaten verbessert, die Generalisierungsfähigkeit, gemessen auf den unbekannten Testdaten, verschlechtert sich jedoch. Dies liegt daran, dass die aus den Trainingsdaten gelernten Zusammenhänge während des Prunings nicht beachtet werden und somit für die Generalisierungsfähigkeit potenziell wichtige Informationen aus den Trainingsdaten verloren gehen [Qui87, EMSK97].

REP mit Feature Engineering Analog zum Pre-Pruning mittels Hyperparametern, wird auch die Kombination des Prunings und des Feature Engineerings aus 3.3.2 betrachtet.
Mit Pruning und Feature Engineering in Kombination erhält man die folgenden Ergebnisse.

Szenario	Genauigkeit Training	Genauigkeit Pruning	Genauigkeit Test
1.	68.03% ↓	29.82% ↑	19.85%
2.	65.45% ↓	40.00% ↑	35.48% ↑
3.	44.88% ↓	42.11% ↑	36.96% ↑
4.	84.17% ↓	71.93% ↑	68.94% ↑
5.	81.82% ↓	68.00% ↑	56.45% ↓
6.	81.89% ↓	64.91% ↑	71.64% ↑
7.	88.41% ↓	62.50% ↑	40.51% ↓

Tabelle 3.36.: Genauigkeiten mit REP auf ausgewählten Attributen

Szenario	RMSE Training	RMSE Pruning	RMSE Test	MAE Training	MAE Pruning	MAE Test
1.	48.63 ↑	139.73 ↓	173.23 ↓	20.61 ↑	103.92 ↓	137.81 ↑
1a.	57.53 ↑	120.13 ↓	138.35 ↓	37.15 ↑	94.27 ↓	101.63 ↓
2.	88.83 ↑	102.89 ↓	125.30 ↓	29.71 ↑	79.32 ↓	106.43 ↓
3.	68.43 ↑	137.30 ↓	150.31 ↓	60.71 ↑	117.23 ↓	118.02 ↑

Tabelle 3.37.: Fehler mit REP auf ausgewählten Attributen

Szenario	AUC Training	AUC Pruning	AUC Test	F1 Training	F1 Pruning	F1 Test
4.	0.9884 ↓	0.5904 ↑	0.5584 ↓	84.37% ↓	31.25% ↑	30.77% ↓
5.	0.8801 ↓	0.7143 ↑	0.6016 ↑	84.21% ↓	66.67% ↑	56.60% ↑
6.	0.9259 ↓	0.5831 ↑	0.6473 ↑	54.17% ↓	27.59% ↑	53.85% ↑

Tabelle 3.38.: AUC und F1-Score mit REP auf ausgewählten Attributen

Für Szenario 4 ergeben sich die selben Ergebnisse wie im vorherigen Abschnitt, da die selben Attribute genutzt werden.
Im Vergleich zum REP auf allen Attributen ergeben sich insgesamt nur geringe Unterschiede. Es fällt jedoch auf, dass die Klassifikationsbäume insgesamt mehr vom Feature Engineering profitieren als die Regressionsbäume. Unter Berücksichtigung der Generalisierungsfähigkeit, die mit den Testdaten gemessen wird, lassen sich für die einzelnen Szenarien die nachfolgenden Tendenzen festhalten, ob Feature Engineering zielführend ist oder nicht:

- **Szenario 1** ohne Feature Engineering besser
- **Szenario 1a** ohne Feature Engineering besser
- **Szenario 2** kaum Unterschied
- **Szenario 3** mit Feature Engineering besser
- **Szenario 4** kein Unterschied
- **Szenario 5** ohne Feature Engineering besser
- **Szenario 6** mit Feature Engineering besser
- **Szenario 7** ohne Feature Engineering besser

CVP

Beim CVP besteht die Gefahr des Overprunings nicht, sondern die erhaltenen Entscheidungsbäume neigen im Gegenteil zum Underpruning [EMSK97]. Für die einzelnen Schwellenwerte werden bei Klassifikationsbäumen (Splitkriterium Gini) Werte im Intervall $[0.05, 1.0]$ mit Schrittweite 0.05 und bei Regressionsbäumen (Splitkriterium MSE) Werte im Intervall $[100, 250000]$ mit Schrittweite 100 eingesetzt.

CVP ohne Feature Engineering Critical Value Pruning ohne Feature Engineering führt zu den folgenden Ergebnissen.

Szenario	Genauigkeit Training	Genauigkeit Pruning	Genauigkeit Test
1.	100%	21.05%	19.85%
2.	74.55% ↓	36.00%	24.19%
3.	33.07% ↓	36.84% ↑	38.41% ↑
4.	91.67% ↓	59.65% ↑	59.09% ↓
5.	90.91% ↓	68.00% ↑	62.90% ↑
6.	95.28% ↓	45.61% ↑	65.67% ↑
7.	94.20% ↓	46.88%	37.97% ↓

Tabelle 3.39.: Genauigkeiten mit CVP auf allen Attributen

Szenario	RMSE Training	RMSE Pruning	RMSE Test	MAE Training	MAE Pruning	MAE Test
1.	52.49 ↑	161.47 ↓	180.89 ↓	10.65 ↑	125.44 ↓	141.35 ↑
1a.	10.26 ↑	153.49	162.43 ↓	1.31 ↑	130.64	126.20 ↑
2.	83.47 ↑	104.03 ↓	134.89 ↓	0.0	73.6	116.61
3.	107.10 ↑	156.08 ↓	140.16 ↓	18.06 ↑	122.42 ↓	122.82 ↑

Tabelle 3.40.: Fehler mit CVP auf allen Attributen

Szenario	AUC Training	AUC Pruning	AUC Test	F1 Training	F1 Pruning	F1 Test
4.	1.0	0.5372	0.5764	94.29% ↓	28.57%	32.94% ↓
5.	0.9686 ↓	0.7532 ↑	0.6557 ↑	86.49% ↓	71.43% ↑	58.18% ↑
6.	0.9891 ↓	0.4128 ↑	0.5214 ↑	72.22% ↓	26.09% ↑	36.59% ↑

Tabelle 3.41.: AUC und F1-Score mit CVP auf allen Attributen

In den Szenarien 4 zum AUC-Wert sowie dem Klassifikationsbaum in Szenario 1 bleibt der Entscheidungsbaum durch den Einsatz von CVP unverändert. Auch im Vergleich zu REP ohne Feature Engineering fällt auf, dass bis auf eine Ausnahme (Szenario 3 Genauigkeit) die entstandenen Bäume teilweise deutlich größer sind. Dies steht im Einklang mit der Annahme, dass REP zum Overpruning und CVP zum Underpruning tendiert [EMSK97].
Gemeinsamkeiten zwischen den einzelnen durch CVP entstandenen Entscheidungsbäumen lassen sich ebenfalls feststellen. Bei allen Bäumen von Szenario 5 wird der gleiche Schwellenwert für das Pruning gewählt, sodass die selben Bäume entstehen. Auch für Szenario 6 stimmen die Bäume zur Genauigkeit und AUC überein.
Beim Klassifikationsbaum zu Szenario 3 entsteht lediglich ein Stumpf mit einer Entscheidung und der Einteilung in die Noten 1.0 und 5.0.

Ein weiteres interessantes Ergebnis zeigt sich in Szenario 5 bei den F1-Werten. Hier wird von REP und CVP ohne Feature Engineering die gleiche Performanz erreicht. Jedoch unterscheiden sich die entstandenen Bäume. Der Entscheidungsbaum nach dem REP ist kleiner als der Baum nach CVP. Dies verdeutlicht anschaulich die Eigenschaft des REP, unter allen Bäumen mit der selben Performanz auf den Pruningdaten den kleinsten zu finden [Qui87].

CVP mit Feature Engineering Die Kombination aus Feature Engineering und CVP ergibt leicht andere Ergebnisse.

Szenario	Genauigkeit Training	Genauigkeit Pruning	Genauigkeit Test
1.	84.42% ↓	17.54% ↑	18.38% ↓
2.	100%	32.00%	25.81%
3.	33.07% ↓	36.84% ↑	38.41% ↑
4.	91.67% ↓	59.65% ↑	59.09% ↑
5.	89.09% ↓	64.00% ↑	62.90% ↑
6.	85.04% ↓	63.16% ↑	73.13% ↑
7.	100%	59.38%	41.77%

Tabelle 3.42.: Genauigkeiten mit CVP auf ausgewählten Attributen

Szenario	RMSE Training	RMSE Pruning	RMSE Test	MAE Training	MAE Pruning	MAE Test
1.	58.19 ↑	149.65 ↓	172.38 ↓	16.75 ↑	116.46 ↓	135.78 ↑
1a.	0.0	146.63	161.92	0.0	121.90	118.60
2.	83.47 ↑	109.28 ↓	131.22 ↓	24.01 ↑	94.89 ↓	109.71 ↓
3.	97.25 ↑	149.41 ↓	141.01 ↓	50.70 ↑	125.84 ↓	122.75 ↑

Tabelle 3.43.: Fehler mit CVP auf ausgewählten Attributen

Szenario	AUC Training	AUC Pruning	AUC Test	F1 Training	F1 Pruning	F1 Test
4.	1.0	0.5372	0.5764	94.29% ↓	28.57%	32.94% ↓
5.	0.9444 ↓	0.7143 ↑	0.6238 ↑	84.21% ↓	66.67% ↑	56.60% ↑
6.	0.9998	0.5189	0.6029	62.75% ↓	27.59% ↑	55.00% ↑

Tabelle 3.44.: AUC und F1-Score mit CVP auf ausgewählten Attributen

Der Entscheidungsbaum von Szenario 7 wird nicht verändert. Zusätzlich werden auch die Bäume des Szenarios 1a, sowie die Bäume zu den AUC-Werten der Szenarien 4 und 6 nicht gekürzt durch CVP in Verbindung mit Feature Engineering.

Neben den offensichtlich gleichen Entscheidungsbäumen in Szenario 4 hat sich auch der Klassifikationsbaum des Szenarios 3 im Vergleich zu ohne Feature Engineering nicht verändert.
Wie auch beim REP wird betrachtet, ob das Feature Engineering Vorteile bringt.

- **Szenario 1** mit Feature Engineering minimal besser
- **Szenario 1a** mit Feature Engineering besser
- **Szenario 2** mit Feature Engineering minimal besser
- **Szenario 3** kaum Unterschied
- **Szenario 4** kein Unterschied
- **Szenario 5** ohne Feature Engineering besser
- **Szenario 6** mit Feature Engineering besser
- **Szenario 7** mit Feature Engineering besser

Zusammenfassung Pruning

Weder durch REP noch CVP konnten wesentliche Verbesserungen im Vergleich zum Pre-Pruning erzielt werden. Jedoch sind kleinere Entscheidungsbäume nach *Occam's Razor* zu präferieren. Occam's Razor besagt, dass unter allen gleichperformanten Möglichkeiten Daten zu beschreiben, diejenige gewählt werden sollte, welche die kürzeste beziehungsweise die am wenigsten komplexeste ist [Mit97]. Auch wenn es einige Gegenargumente gegen die Allgemeingültigkeit dieses Prinzips gibt [Mit97], so ist im Rahmen dieser Arbeit die Anwendung von Occam's Razor trotz allem sinnvoll. Es sollen Anhaltspunkte für die Entscheidung über eine Zulassung zum Hochschulstudium gegeben werden, weshalb die Interpretierbarkeit und damit eine geringe Größe der Entscheidungsbäume ein wichtiges Kriterium ist. Betrachtet man die Größen der Entscheidungsbäume nach Post- und Pre-Pruning, so lässt sich feststellen, dass für die einzelnen Szenarien im Schnitt die folgenden Verfahren kleinere Bäume erzeugen:

- **Szenario 1** Pre-Pruning
- **Szenario 1a** Pre-Pruning
- **Szenario 2** Pre-Pruning

- **Szenario 3** Pre-Pruning
- **Szenario 4** Pre-Pruning
- **Szenario 5** REP
- **Szenario 6** CVP
- **Szenario 7** Pre-Pruning

3.3.5. Confidence

Um einschätzen zu können, wie verlässlich die Vorhersage eines Entscheidungsbaums ist, reicht die Angabe der allgemeinen Performanz nicht aus. Es muss für jede einzelne Vorhersage entschieden werden können, wie wahrscheinlich es ist, dass die Vorhersage korrekt ist. Zunächst betrachten wir den Fall von (binären) Klassifikationsbäumen.
Eine intuitive Methode wäre es, die Verteilung der Klassen im Blattknoten zu betrachten, der vom eingegebenen Datensatz erreicht wird. Hier kann dann der Anteil der Trainingsdatensätze die die vorhergesagte Klasse haben an allen im Blatt vertretenen Trainingsdatensätzen betrachtet werden und dies als Wahrscheinlichkeit angesehen werden, dass die Vorhersage korrekt ist. Da die Blätter eines Entscheidungsbaums häufig nicht mehr von vielen Trainingsdatensätzen erreicht werden, muss jedoch auch die absolute Anzahl in die Berechnung der Wahrscheinlichkeit einfließen.

Beispiel 3.1. *Blatt A wird von vier Datensätzen der Klasse 1 und null Datensätzen der Klasse 2 erreicht. Blatt B wird von 70 Datensätzen der Klasse 1 und zwei Datensatz der Klasse 2 erreicht. Eine reine Bestimmung der Verhältnisse würde einer Vorhersage der Klasse 1 in Blatt A eine höhere Korrektheitswahrscheinlichkeit (100%) zuordnen als bei Erreichen von Blatt B (98.59%), obwohl wir uns bei Blatt B über die Korrektheit der Vorhersage sicherer sein können aufgrund der deutlich größeren Anzahl an Datensätzen.*

Um dies zu berücksichtigen kann der sogenannte *Wilson-Score* genutzt werden, da es sich bei der Unterscheidung zwischen *bestanden* und *nicht bestanden* um eine Binomialverteilung handelt.

Definition 3.3 (Wilson-Score). *[Wil27, S. 209]*
Der tatsächliche Anteil p der vorhergesagten Klasse liegt auf dem α-Signifikanzniveau nach dem Wilson-Score im Intervall

$$p = \frac{p_0 + \frac{\lambda^2}{2*n} \pm \lambda\sqrt{\frac{p_0(1-p_0)+\frac{\lambda^2}{4n}}{n}}}{1+\frac{\lambda^2}{n}}$$

wobei

- p_0*: Anteil der „positiven“ vorhergesagten Klasse im Blatt*
- λ*:* $1 - \frac{\alpha}{2}$*-Quantil der Normalverteilung*
- n*: Anzahl der Trainingsdaten im Blatt*

Existieren wie in Szenarien 1 bis 3 und 7 mehr als zwei Klassen, so können die Anzahlen der Klassen, die nicht der Vorhersage entsprechen, als negatives Ergebnis der Binomialverteilung zusammengefasst werden und es liegt wiederum eine Binomialverteilung vor. Im Rahmen der Einschätzung der Korrektheit der Vorhersage ist insbesondere die untere Grenze des Wilson-Score Intervalls von Interesse.
In der Implementierung ist es daher mit Hilfe der Funktion *calculate_confidence_for_sample()* möglich, für eine einzelne Vorhersage die untere Grenze des Wilson-Score Intervalls zu berechnen. Damit diese Berechnungen sinnvolle Ergebnisse liefern können, sollte die Anzahl der Trainingsdaten in den Blättern nicht zu gering sein. Der Entscheidungsbaum sollte also entweder durch Pre- oder Post-Pruning modifiziert worden sein. Zusätzlich wird der Pfad im Entscheidungsbaum visualisert, der zur gegebenen Vorhersage führt.
Für Regressionsbäume ist eine andere Methode nötig, um ein Maß für die (Un-) Sicherheit der Vorhersage zu bestimmen. Betrachtet man die Verteilungen der Noten in Anhang E, ergibt sich weder eine Normalverteilung, noch eine Studentsche t-Verteilung [Stu08] oder vergleichbare Verteilung. Da wir angeben wollen, wie weit entfernt unsere Vorhersage von der tatsächlichen Note ist, kann der MSE bzw. RMSE als Anhaltspunkt für die Güte der Vorhersage genutzt werden. Dieser ist bereits beim Training des Entscheidungsbaums berechnet worden. Um eine genauere Einschätzung der Korrektheit der Vorhersage zu ermöglichen, wird neben dem RMSE durch die Funktion *show_uncertainty_for_sample()* auch die Anzahl der Trainingsdaten im betreffenden Blatt, sowie eine Visualisierung der Verteilung dieser Daten gegeben.
Eine Alternative bietet die Funktion *calculate_bootstrap_confidence_interval()*. Hiermit kann ein Konfidenzintervall um den Wert der Vorhersage berechnet werden. Da wie erwähnt die zugrundeliegende Verteilung keine Normalverteilung ist, bietet Bootstrapping eine Möglichkeit das Konfidenzintervall auf Grundlage der empirischen Verteilung zu berechnen. Dieses Verfahren ist jedoch mit einem hohen Rechenaufwand verbunden, weshalb die Anzahl an Bootsptrapping-Durchläufen manuell festgelegt werden kann.

3.3.6. Künstliche Daten

Insgesamt lässt sich festhalten, dass die Vorhersagen der Entscheidungsbäume nicht gut sind. Ein möglicher Grund ist die geringe Anzahl an Datensätzen, die zur Verfügung stehen. Eine weitere Möglichkeit ist, dass die Korrelationen zwischen den Attributen und Labels zu schwach sind, um mit Hilfe eines Entscheidungsbaum zufriedenstellende Ergebnisse zu erhalten. Um zu überprüfen, ob mit mehr Daten oder höheren Korrelationen die Performanz verbessert wird, werden künstliche Daten erzeugt.
Es werden verschiedene Korrelationsmatrizen benutzt, um mit Hilfe der Cholesky-Zerlegung der Korrelationsmatrix zufällige Datensätze zu erzeugen, deren Korrelationen mit den Korrelationsmatrizen übereinstimmen [Mad15]. Um zu vermeiden, dass lediglich die Korrelationen durch den Entscheidungsbaum gelernt werden, wird für die Trainingsdaten zusätzlich ein Fehler auf die Werte addiert bzw. subtrahiert. Dabei hat sich beim Basisfall mit $n = 277$ Datensätzen gezeigt, dass für die Szenarien 1 bis 3 ein größerer Fehler erzeugt werden muss, um mit den originalen Studierendendaten vergleichbare Ergebnisse zu erzielen.
Für die Erstellung des Entscheidungsbaums wird analog zu 3.3.3 *GridSearchCV* genutzt, um Pre-Pruning durchzuführen.
Es zeigt sich, dass für ähnlich hohe Korrelationen wie die in 3.2.2 gefundenen Korrelationen durch eine reine Anhebung der Anzahl an Datensätzen nur begrenzte Verbesserungen erreicht werden können.

Szenario	Genauigkeit Training	Genauigkeit Test	Genauigkeit CV
1.	46.63%	28.57%	43.89%
2.	55.44%	41.67%	50.03%
3.	47.67%	28.57%	43.99%
4.	78.76%	47.62%	72.56%
5.	72.02%	61.90%	66.34%
6.	78.76%	47.62%	72.56%
7.	70.98%	50.00%	60.18%

Tabelle 3.45.: Genauigkeiten künstliche Daten $n = 277$, niedrige Korrelationen

Szenario	RMSE Training	RMSE Test	RMSE CV	MAE Training	MAE Test	MAE CV
1.	34.15	107.12	103.01	26.71	74.01	74.92
1a.	34.42	163.42	96.08	26.94	91.86	70.75
2.	48.82	64.05	77.51	33.37	49.30	54.64
3.	46.64	95.48	101.99	10.05	146.19	75.48

Tabelle 3.46.: Fehler künstliche Daten $n = 277$, niedrige Korrelationen

Szenario	AUC Training	AUC Test	AUC CV	F1 Training	F1 Test	F1 CV
4.	0.8478	0.5524	0.6507	64.96%	43.59%	51.60%
5.	0.8065	0.5435	0.6585	70.06%	41.94%	58.09%
6.	0.8478	0.5524	0.6507	64.96%	43.59%	51.60%

Tabelle 3.47.: AUC und F1-Score künstliche Daten $n = 277$, niedrige Korrelationen

Szenario	Genauigkeit Training	Genauigkeit Test	Genauigkeit CV
1.	44.00% ↓	42.33% ↑	40.03% ↓
2.	44.71% ↓	44.00% ↑	42.99% ↓
3.	50.00% ↑	14.00% ↓	40.03% ↓
4.	71.71% ↓	71.33% ↑	71.72% ↓
5.	67.14% ↓	59.00% ↓	63.58% ↓
6.	76.86% ↓	73.67% ↑	72.57% ↑
7.	53.29% ↓	55.33% ↑	49.42% ↓

Tabelle 3.48.: Genauigkeiten künstliche Daten $n = 1000$, niedrige Korrelationen

Szenario	RMSE Training	RMSE Test	RMSE CV	MAE Training	MAE Test	MAE CV
1.	31.35 ↓	142.17 ↑	94.23 ↓	19.69 ↓	93.23 ↑	65.36 ↓
1a.	30.09 ↓	87.47 ↓	88.57 ↓	21.59 ↓	64.18 ↓	65.74 ↓
2.	43.13 ↓	117.15 ↑	89.94 ↑	12.09 ↓	47.83 ↓	62.94 ↑
3.	38.12 ↓	71.06 ↓	93.99 ↓	13.22 ↑	143.80 ↓	65.96 ↓

Tabelle 3.49.: Fehler künstliche Daten $n = 1000$, niedrige Korrelationen

Szenario	AUC Training	AUC Test	AUC CV	F1 Training	F1 Test	F1 CV
4.	0.8865 ↑	0.6056 ↑	0.6107 ↓	98.22% ↑	37.65% ↓	40.64% ↓
5.	0.8709 ↑	0.5863 ↑	0.6095 ↓	94.08% ↑	44.53% ↑	50.19% ↓
6.	0.8072 ↓	0.5468 ↓	0.6264 ↓	64.62% ↓	26.98% ↓	37.26% ↓

Tabelle 3.50.: AUC und F1-Score künstliche Daten $n = 1000$, niedrige Korrelationen

Werden statt der Anzahl an Datensätzen die Korrelationen erhöht, ergibt sich ein ähnliches Bild. Mit den selben Methoden beim Erzeugen des Baums werden die Entscheidungsbäume nicht wesentlich performanter, obwohl höhere Korrelationen im Bereich von durchschnittlich $\bar{\rho} \approx 0.5$ verwendet werden.

Szenario	Genauigkeit Training	Genauigkeit Test	Genauigkeit CV
1.	46.11%	14.29%	46.37%
2.	53.89%	4.76%	48.82%
3.	47.15%	14.29%	47.44%
4.	78.24%	63.10%	70.53%
5.	80.83%	54.76%	72.03%
6.	78.24%	63.10%	70.53%
7.	87.56%	75.00%	71.03%

Tabelle 3.51.: Genauigkeiten künstliche Daten $n = 277$, höhere Korrelationen, $\bar{\rho} \approx 0.5$

Szenario	RMSE Training	RMSE Test	RMSE CV	MAE Training	MAE Test	MAE CV
1.	39.55	122.04	89.96	25.78	91.67	64.50
1a.	28.56	124.09	87.81	3.32	64.61	64.72
2.	17.25	159.07	80.61	12.12	136.43	51.13
3.	46.10	111.33	87.75	16.81	77.60	62.18

Tabelle 3.52.: Fehler künstliche Daten $n = 277$, höhere Korrelationen, $\bar{\rho} \approx 0.5$

Szenario	AUC Training	AUC Test	AUC CV	F1 Training	F1 Test	F1 CV
4.	0.8744	0.6399	0.6834	89.47%	32.00%	47.21%
5.	0.9520	0.6071	0.7559	78.67%	32.65%	65.51%
6.	0.8744	0.6399	0.6834	89.47%	32.00%	47.21%

Tabelle 3.53.: AUC und F1-Score künstliche Daten $n = 277$, höhere Korrelationen, $\bar{\rho} \approx 0.5$

Auch die Erhöhung der Anzahl an Datensätzen auf $n = 1000$ bringt keine merkliche Verbesserung.

Szenario	Genauigkeit Training	Genauigkeit Test	Genauigkeit CV
1.	51.71% ↑	41.33% ↑	39.47% ↓
2.	46.43% ↓	44.00% ↑	44.88% ↓
3.	43.86% ↓	41.67% ↑	38.29% ↓
4.	71.71% ↓	71.33% ↑	71.72% ↑
5.	81.14% ↑	73.67% ↑	70.57% ↓
6.	72.29% ↓	71.33% ↑	72.29% ↑
7.	65.57% ↓	60.67% ↓	64.00% ↓

Tabelle 3.54.: Genauigkeiten künstliche Daten $n = 1000$, höhere Korrelationen

Szenario	RMSE Training	RMSE Test	RMSE CV	MAE Training	MAE Test	MAE CV
1.	28.06 ↓	76.27 ↓	86.79 ↓	9.41 ↓	78.87 ↓	60.07 ↓
1a.	24.94 ↓	81.96 ↓	84.29 ↓	15.27 ↑	93.55 ↑	63.47 ↓
2.	21.88 ↑	58.45 ↓	83.70 ↑	10.20 ↓	54.67 ↓	56.87 ↑
3.	36.54 ↓	118.78 ↑	86.75 ↓	17.11 ↑	104.10 ↑	61.10 ↓

Tabelle 3.55.: Fehler künstliche Daten $n = 1000$, höhere Korrelationen

Szenario	AUC Training	AUC Test	AUC CV	F1 Training	F1 Test	F1 CV
4.	0.7114 ↓	0.4841 ↓	0.5940 ↓	93.73% ↑	23.29% ↓	37.28% ↓
5.	0.8900 ↓	0.6833 ↑	0.7514 ↓	75.65% ↓	59.07% ↑	62.17% ↓
6.	0.8170 ↓	0.5705 ↓	0.6088 ↓	81.36% ↓	24.56% ↓	37.15% ↓

Tabelle 3.56.: AUC und F1-Score künstliche Daten $n = 1000$, höhere Korrelationen

Erst ab Korrelationen von durchschnittlich $\bar{\rho} \approx 0.7$ ergeben sich für die Szenarien 4 bis 7 gute Ergebnisse durch die Entscheidungsbäume für $n = 277$. Für die Szenarien 1 bis 3 sind selbst Korrelationen von durchschnittlich $\bar{\rho} \approx 0.8$ mit $n = 277$ Datensätzen nicht ausreichend um deutliche Verbesserungen zu erreichen. Die entsprechenden Tabellen hierzu finden sich in Anhang F.
Es kann also die Schlussfolgerung gezogen werden, dass die Vorhersage des Erfolgs in Programmierung I und Mathematik für Informatiker I nur leicht von einer höheren Anzahl an Datensätzen profitieren wird. Hauptsächlich müssen andere Attribute erfasst werden, deren Korrelationen mit den Ergebnissen in diesen beiden Vorlesungen deutlich größer sind. Allein durch eine Auswahl der bereits verfügbaren Attribute kann dieses Ziel nicht erreicht werden, da auch die größten Korrelationen immer noch recht schwach sind. Dementsprechend muss der Fragebogen für die Studierenden überarbeitet werden, sofern Entscheidungsbäume genutzt werden sollen. Eine Alternative wäre es, andere

Verfahren des Maschinellen Lernens zur Vorhersage zu nutzen. Vielversprechend könnten neuronale Netze sein, welche in der Lage sind auch komplexere Zusammenhänge als Entscheidungsbäume zu erfassen. Allerdings ist hierfür voraussichtlich eine größere Datenmenge nötig. Ein entscheidender Nachteil der Verwendung neuronaler Netze ist die dann fehlende Einsicht in den Entstehungsprozess der Vorhersagen. Beim Einsatz zur Studienzulassung sollte ersichtlich sein, wie eine bestimmte Vorhersage automatisiert zustande kommt.
Im Fragebogen wurden bereits die in mehreren psychologischen Studien zur Vorhersage des Studienerfolgs empfohlenen Kriterien erfasst [OP07, KF12]. Die für die Studierendendaten erhaltenen Korrelationen und Ergebnisse sind, wie in 3.2.3 erwähnt, vergleichbar mit denjenigen der Studien. Die Metastudie von O'Connor und Paunonen [OP07] zeigt, dass die Betrachtung der fünf großen Persönlichkeitsmerkmale Gewissenhaftigkeit, Extraversion, Neurotizismus, Verträglichkeit und die Offenheit für Erfahrungen nur für eine geringe Varianzaufklärung im Bereich von ungefähr 6% verantwortlich sind. Auch die feineren Unterscheidungen dieser Kriterien die in Ansätzen ebenfalls im Fragebogen der Studierenden zu finden sind, erhöhen die Varianzaufklärung nur um weitere $5 - 7\%$. Somit ist keine wesentliche Verbesserung der Vorhersagen mit weiteren Daten zu erwarten, sofern der gleiche Fragebogen genutzt wird. Hier ist weitere Forschung im Bereich der Psychologie notwendig, um eventuell für Entscheidungsbäume bessere Prädiktoren des Studienerfolgs zu finden.

3.4. Zusammenfassung

Unter Berücksichtigung der Klassenverhältnisse aus Tabelle 3.4 sind die erhaltenen Entscheidungsbäume nicht wesentlich besser geeignet für die Vorhersage als der naive Prädiktor. Eine Abweichung von ungefähr einer bis anderthalb Notenstufen im Absolutbetrag ist weiterhin beachtlich. Auffallend ist jedoch, dass die besten Ergebnisse in den Szenarien 2 und 5 erreicht werden, also in denjenigen die sich auf die Vorlesung MfI I beziehen. Es scheint also, als sei der Fragebogen zur Vorhersage der Leistungen in MfI I besser geeignet, als für die Vorhersage der Leistungen in Programmierung I. Auch bei der Betrachtung der Korrelationen aus Tabelle 3.3 scheint sich dieses Ergebnis zu bestätigen, da für diese Szenarien mehr Items eine Korrelation $\rho \geq 0.2$ erreichen und die erhaltenen Korrelationen im Schnitt größer sind als in den Szenarien zu Programmierung I.
Ein mögliches Problem bei der Erfassung der Daten sind fehlende Informationen bezüglich Studierender, die die Zulassung zur Klausur nicht erreichen. Zwar liegen anhand der Fragebögen Daten zu Studierenden vor, die die Klausuren nicht mitgeschrieben haben,

diese Informationen können jedoch nicht genutzt werden um als Studierende gezählt zu werden, die die Klausur nicht bestanden haben. Hierfür fehlen weitere Angaben zu diesen Studierenden. Es müsste zusätzlich der Grund erfasst werden, weshalb es keine Teilnahme an den Klausuren gab. Neben der nicht erreichten Klausurzulassung können auch Gründe wie ein Abbruch der Vorlesung zu Gunsten einer anderen Veranstaltung verantwortlich sein für die Nicht-Teilnahme an der Klausur. Ein solcher Abbruch kann nicht zwangsläufig auf mangelnde Leistungen zurückgeführt werden, da es keine Begrenzung gibt, wie viele Vorlesungen zu Beginn des betreffenden Semesters belegt werden können. Somit muss bei einem erhöhten Arbeitsaufwand im Laufe des Semesters eine Auswahl getroffen werden. Insbesondere bei Vorlesungen wie Programmierung I und MfI I, welche zu Beginn des Studiums stattfinden, gibt es darüber hinaus viele Studierende die den Studiengang wechseln. Es ist also offensichtlich, dass es viele verschiedene Gründe für die Nicht-Teilnahme an den Abschlussklausuren einer Vorlesung gibt. Somit ist die Gruppe an Studierenden für die lediglich die Antworten des Fragebogens vorliegen kaum repräsentativ für die Gruppe an Studierenden die aufgrund unzureichender Leistungen nicht an der Klausur teilgenommen haben. Daher können die entsprechenden Antworten der Studierenden nicht als Attribute für Datensätze verwendet werden, die mit dem Label *nicht bestanden* versehen werden.

4. Bestehende Angebote zu Maschinellem Lernen für Schülerinnen und Schüler

In diesem Kapitel sollen die derzeit bereits bestehenden Angebote für Schülerinnen und Schüler zum Thema Künstliche Intelligenz und Maschinelles Lernen aufgezeigt werden.

4.1. Schulische Entwicklung

In den schulischen Lehrplänen findet sich Maschinelles Lernen nicht wieder. Einige Bundesländer führen Künstliche Intelligenz im Allgemeinen als einen möglichen Kontext im Inhaltsbereich Informatik, Mensch und Gesellschaft an [LPB06], [Min18]. Auch als eines von vielen möglichen Vertiefungsthemen wird Künstliche Intelligenz in den Lehrplänen von Berlin [LPB06], Brandenburg [Min18], Hessen [Hes16] und Schleswig-Holstein [Min02] genannt. Selbst im neuen Informatiklehrplan des Saarlands von 2019 findet sich das Thema Künstliche Intelligenz nicht wieder [LPN19]. Dieser Lehrplan befindet sich momentan jedoch noch in der Erprobungsphase, weshalb vor der Ausgabe der finalen Version noch Anpassungen vorgenommen werden können. Im Hinblick auf die in Kapitel 1.1 aufgezeigte Relevanz des Maschinellen Lernens als allgemeinbildender Inhalt des Faches Informatik ist das Fehlen dieses Themenfelds in den Lehrplänen sehr bedenklich.

4.2. Schülerlabore

Neben der schulischen Bildung gibt es für viele verschiedene Fachrichtungen Schülerlabore, die zumeist an Hochschulen angesiedelt sind. In diesen Schülerlaboren können Schülerinnen und Schüler Erfahrungen sammeln, die über die Bildungsstandards und Schulcurricula hinausgehen. Auch im Fach Informatik gibt es mittlerweile einige Schülerlabore. Das im Rahmen dieser Arbeit entstandene Unterrichtsmodul wird im Schülerlabor für Informatik an der Universität des Saarlandes eingesetzt werden, welches sich momentan noch im Aufbau befindet.
Von allen im Schülerlabor-Atlas 2019 [Ler19] aufgeführten Schülerlaboren bieten derzeit mit *phaeno*, der *Fachhochschule Südwestfalen* und *InfoSphere* nur drei Labore Inhalte zu Künstlicher Intelligenz an. Beim Angebot von *phaeno* handelt es sich um die

Sonderausstellung „Smarte neue Welt“ zu den Auswirkungen der Digitalisierung und der Künstlichen Intelligenz auf den Alltag. Die *Fachhochschule Südwestfalen* plant für die KinderUni 2020 eine Veranstaltung zu Künstlicher Intelligenz.
Das einzige fortlaufende Angebot zu Künstlicher Intelligenz findet sich im Schülerlabor *InfoSphere* der RWTH Aachen. Hier werden zwei verschiedene Module zum Themenbereich Künstliche Intelligenz angeboten. Ein Modul für die Unter- und Mittelstufe, bei dem ein Chatbot programmiert wird, sowie ein Modul für die Oberstufe, bei dem ebenfalls anhand der Programmierung eines Chatbots die Grundprinzipien intelligenter Systeme vermittelt werden. Chatbots sind Programme die in der Lage sind automatisiert Antworten im Rahmen eines Chats zu geben. Aufgrund des Programmieransatzes sind grundlegende Vorkenntnisse im Bereich der Programmierung für dieses Modul nötig.
Im Gegensatz zu diesen beiden Modulen soll das für diese Arbeit neu entwickelte Modul den Schülerinnen und Schülern einen Einblick in Maschinelles Lernen ohne Programmierung ermöglichen. Es sind keinerlei informatikspezifische Vorkenntnisse nötig.
Eine etwas andere Art des Angebots ist die Webseite `https://machinelearningforkids.co.uk`. Hierbei handelt es sich um eine Plattform, mit der auf einfache Art und Weise ein Modell des überwachten Lernens erstellt werden kann. Neben dieser Möglichkeit finden sich mehrere Projekte, die die trainierten Modelle mit Hilfe von Scratch oder Python in ein Programm einbetten. Die Webseite bietet ein gutes Angebot um kleinere praktische Projekte zu Maschinellem Lernen im schulischen Kontext durchzuführen, die sich auf das überwachte Lernen beschränken. Allerdings bietet sie lediglich die Möglichkeit einen Einblick in das Benutzen des überwachten Lernens zu erhalten, ohne dabei auf die zugrundeliegenden Algorithmen einzugehen.
Das Projekt `https://ki-macht-schule.de/` eines Teams der Arbeitsgruppe *Künstliche Intelligenz: Fakten, Chancen, Risiken* verfolgt einen ähnlichen Ansatz wie das im Anschluss vorgestellte Unterrichtsmodul. Neben der Vorstellung von Anwendungsbeispielen des Maschinellen Lernens im Alltag und der Erklärung eines konkreten Algorithmus steht die selbstständige Programmierung eines Modells im Rahmen eines Wettbewerbs im Mittelpunkt. Zur Programmierung werden Python oder Scratch genutzt. Insgesamt ist das Projekt auf acht Schulstunden ausgelegt und soll in der Oberstufe durchgeführt werden.

5. Beschreibung des Unterrichtsmoduls

Im folgenden Kapitel wird das Unterrichtsmodul zum Maschinellen Lernen vorgestellt. Das Modul ist als zusammenhängender Workshop über eine Dauer von drei Zeitstunden entworfen worden. Es kann jedoch auch im schulischen Kontext im Rahmen einer kurzen Unterrichtsreihe durch geeignete Einteilungen in kürzere Einheiten eingesetzt werden. Hierzu bieten sich die bereits eingeplanten Pausen im Ablauf des Moduls an.

5.1. Zielgruppe

Das Modul richtet sich in erster Linie an Oberstufenschülerinnen und -schüler. Dabei sind keinerlei Vorkenntnisse im Fach Informatik vorausgesetzt. Als fachliches Vorwissen benötigen die Schülerinnen und Schülern Bruchrechnung, sowie ein grundlegendes Verständnis für Vierfeldertafeln und das Modellieren von Tests im Rahmen der bedingten Wahrscheinlichkeitsrechnung aus Klassenstufe 9 [Min16, S.122]. Dabei reichen die in dieser Klassenstufe behandelten Begriffe der richtig und falsch positiven beziehungsweise negativen Testergebnisse aus. Die erst im E-Kurs der Oberstufe vermittelten Inhalte zu Hypothesentests und Fehler erster und zweiter Art können das Verständnis beschleunigen, sind aber nicht explizit erforderlich. Hilfreich, aber nicht zwingend notwendig, sind ebenfalls ein Grundverständnis für den Algorithmusbegriff sowie Bäume in der Informatik. Neben diesem fachlichen Wissen sollten die Teilnehmer in der Lage sein, an einer Diskussion über ethische und moralische Fragen des Einsatzes von Künstlicher Intelligenz teilzunehmen.

5.2. Lernziele

Das übergeordnete Lernziel ist, dass die Teilnehmer einen ersten Einblick in das weite Themenfeld des Maschinellen Lernens erhalten und dabei ohne jegliche Programmierkenntnisse erfolgreich sind.
Die Schülerinnen und Schüler ...

- klassifizieren Daten anhand eines vorgegebenen Entscheidungsbaums. (1)
- konstruieren Entscheidungsbäume zu vorgegebenen Daten. (2)
- verstehen das Prinzip der Homogenität als Splitkriterium. (3)

- vergleichen Entscheidungsbäume im Hinblick auf Genauigkeit und Größe. (4)
- erklären das Phänomen des Overfittings in eigenen Worten. (5)
- begründen die Notwendigkeit von Trainings- und Testdaten. (6)
- erklären den Einfluss der Hyperparameter eines Entscheidungsbaums. (7)
- diskutieren Chancen und Risiken des Einsatzes Maschinellen Lernens, insbesondere im medizinischen Bereich. (8)
- begründen die nicht immer ausreichende Aussagekraft der Genauigkeit und übertragen die Begriffe Sensitivität und Spezifität auf ein medizinisches Szenario. (9)

5.3. Ablauf

Im Anhang G findet sich eine tabellarische Übersicht über das Unterrichtsmodul. Die zum Modul gehörende Präsentation kann über das öffentliche Repository `https://github.com/FranzWalgenbach/ModulML` heruntergeladen werden. Diese Präsentation ist in Zusammenarbeit mit Lukas Wachter entstanden.
Wird das Modul im Rahmen eines Workshops, beispielsweise im Schülerlabor an der Universität des Saarlandes, durchgeführt, so stellen sich zu Beginn die Tutoren kurz vor. Die Schülerinnen und Schüler sollen im Anschluss kurz berichten, was sie bereits über Maschinelles Lernen und Künstliche Intelligenz wissen. Zusätzlich hierzu wird ein kurzer Fragebogen (siehe Anhang H) ausgeteilt, bei dem das Vorwissen sowie die Vorstellung von Künstlicher Intelligenz erfragt wird. Am Ende des Workshops soll dieser Fragebogen erneut ausgefüllt werden, womit ein zeitlicher Vergleich ermöglicht wird und Veränderungen der Vorstellungen der Schülerinnen und Schüler abgebildet werden können.
An diesen lockeren Einstieg schließt sich der Einführungsvortrag an, welcher eine kurze Übersicht über verschiedene Arten des Maschinellen Lernens gibt. Die Nennung von Beispielen für Systeme, die Maschinelles Lernen einsetzen, bietet hierbei eine nahtlose Überleitungsmöglichkeit. Es wird anhand eines Beispiels erklärt, was lernende Algorithmen von klassischen Algorithmen abgrenzt. Die Schülerinnen und Schüler sollen nach der Erklärung des überwachten Lernens das erste Arbeitsblatt (siehe Anhang I.1) bearbeiten bevor Trainings- und Testdaten erläutert werden. Durch gezielte Fragen des Tutors sollen die Schülerinnen und Schüler selbstständig die Idee des Aufteilens der Daten in Trainings- und Testdaten entwickeln.

Den Abschluss des Überblicks zu Maschinellem Lernen bildet ein interaktives Quiz zu den bisherigen Inhalten, welches über `https://tinyurl.com/quizML` abrufbar ist. Nach einer kurzen Pause führt ein Vortrag des Tutors in die Struktur und Fachbegriffe von (Binär-)Bäumen anhand Abbildung 5.1 ein.

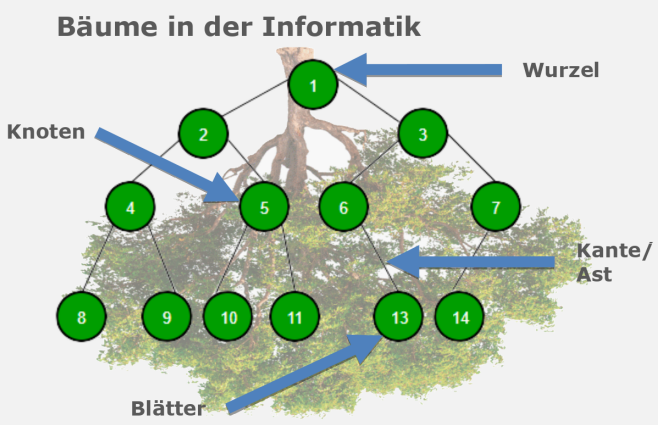


Abbildung 5.1.: Bäume in der Informatik

Nach dieser eher abstrakten Erklärung zu Bäumen in der Informatik bildet ein einfacher Beispielbaum zur Klassifikation von Katzen und Hunden (siehe Anhang K.1), der an der Tafel oder am Whiteboard angeschrieben wird, den ersten Entscheidungsbaum den die Schülerinnen und Schüler kennenlernen. Gemeinsam mit dem Tutor sollen im Plenum mehrere Beispieldatensätze auf Karten an der Tafel eingeordnet werden, um das Prinzip der Entscheidungsfindung mit Hilfe eines Entscheidungsbaums zu entdecken. Durch verschiedenfarbige Rückseiten der Karten kann im Anschluss die Korrektheit überprüft werden.

Beispiel 5.1. *Es soll die Frage „Ist heute geeignetes Wetter, um ins Freibad zu gehen?“ durch einen Entscheidungsbaum beantwortet werden. Für das Label gilt 1=ja, 0=nein.*

Label	*Sonnenschein*	*Regen*	*Temperatur in °C*
0	*1*	*0*	*12*
0	*0*	*0*	*-3*
0	*0*	*1*	*24*
1	*0*	*0*	*23*
1	*1*	*1*	*33*
1	*1*	*0*	*28*

Tabelle 5.1.: Beispiel Trainingsdaten für einen Entscheidungsbaum

Durch das gemeinsame Erstellen eines Entscheidungsbaums (ein möglicher Baum ist in Anhang J.1 zu finden) zu den vorgegebenen Daten aus dem obigen Beispiel 5.1 wird ein Grundverständnis für die Entstehung und Struktur eines Entscheidungsbaums vermittelt und im Anschluss durch die Bearbeitung des folgenden Arbeitsblatts in Einzelarbeit vertieft.

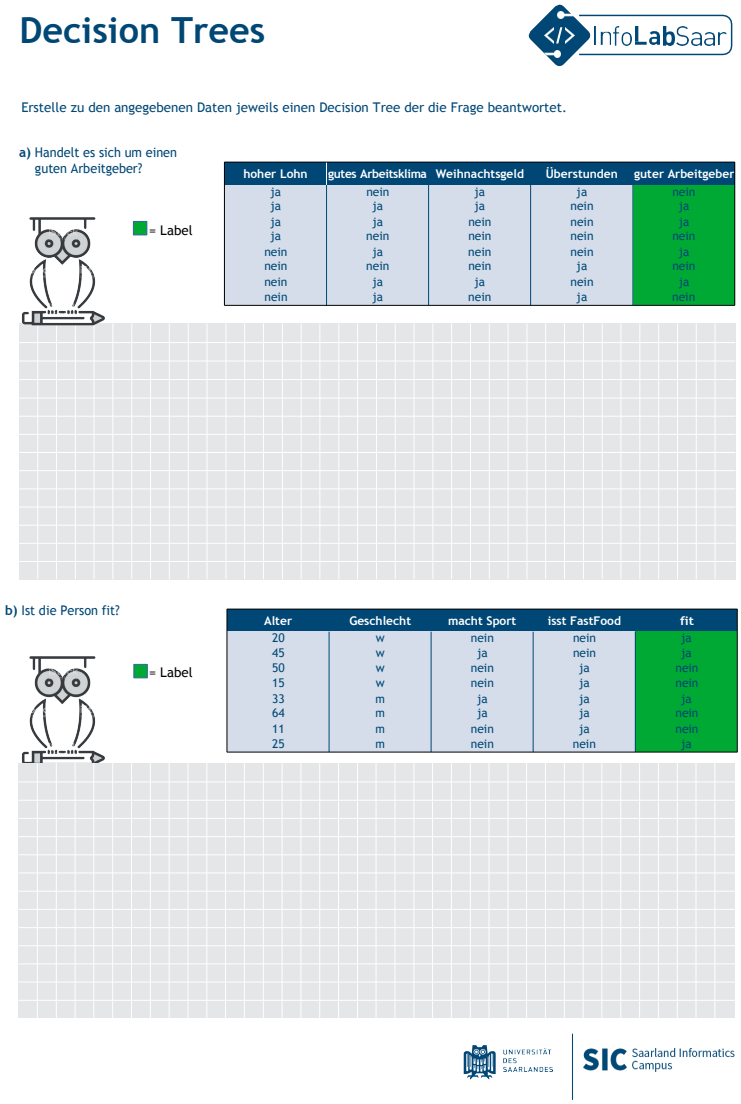


Abbildung 5.2.: AB Entscheidungsbäume Version A

Das Arbeitsblatt existiert in zwei Versionen, bei denen die Reihenfolge der Spalten in Aufgabenteil a) vertauscht ist. Die zweite Version ist in Anhang I.2 zu finden. Hierdurch entstehen durch die erwartungsgemäße Bearbeitung der Spalten von links nach rechts

unterschiedliche Entscheidungsbäume zu gleichen Daten. Diese Uneindeutigkeit wird bei der anschließenden Besprechung der Lösungen der Schülerinnen und Schüler angesprochen. Es soll mit den Teilnehmern diskutiert werden, welcher Entscheidungsbaum besser geeignet ist. Hierbei werden die Größe, Effizienz und Genauigkeit thematisiert. Der Tutor führt hiernach in das im weiteren Vortrag führende Beispiel der Regenvorhersage ein und stellt sowohl die Struktur der Daten als auch die mit den Daten verbundene Fragestellung „Wird es regnen?“ vor. In Verbindung hiermit sollen die Schülerinnen und Schüler erkennen, dass neben der Quantität der Daten auch die Qualität eine Rolle spielt. Beispielsweise sollten für die Regenvorhersage in Saarbrücken nach Möglichkeit nur die regionalen Wetterdaten für das Training des Modells genutzt werden. Die Daten für dieses Beispiel stammen vom Deutschen Wetterdienst und sind frei über das „Climate Data Center OpenData“ abrufbar. Als kurze Wiederholung wird die Entscheidungsfindung an einem Entscheidungsbaum für die Regenvorhersage nachvollzogen. Im Anschluss wird der Entstehungsprozess eines Entscheidungsbaums konkreter thematisiert. Anhand der Darstellung der Daten in einem Streudiagramm zu zwei Attributen in Abbildung 5.3 sollen die Schülerinnen und Schüler Vorschläge machen, wie zwischen den einzelnen Kategorien unterschieden werden kann, um einen möglichst performanten Baum zu erhalten. Somit wird eine Verbindung zum vorher bearbeiteten Arbeitsblatt hergestellt. Es wird erwartet, dass die Idee der Schwellenwerte für die einzelnen Attribute von den Schülerinnen und Schülern erkannt wird.

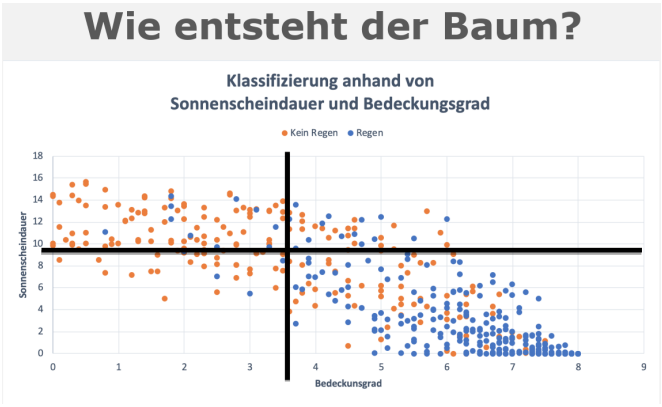


Abbildung 5.3.: Unterteilung der Daten durch Schwellenwerte

Das Prinzip der Homogenität als Splitgrundlage soll selbstständig durch den Vergleich mehrerer Beispielbäume zu verschiedenen Attribut-Schwellenwert-Kombinationen entdeckt werden. Hierzu wird vom Tutor begleitend auf den Entscheidungsbaum zu Hunden und Katzen zurückgegriffen und die größer werdende Homogenität innerhalb der Knoten visualisiert.

Im Anschluss an diesen Block folgt eine Pause.
Nach der Pause wird ein weiteres Quiz `https://tinyurl.com/quizBaum` eingesetzt, um die gelernten Inhalte zu wiederholen und um die Aufmerksamkeit der Schülerinnen und Schüler wieder auf den Workshop zu richten. Die Teilnehmer lesen nach der Besprechung des Quiz auf der Webseite des Moduls `modulml.cs.uni-saarland.de/results` eigenständig die Erklärung zur Genauigkeit und dem Problem des Overfittings. Mit der Vorstellung der Webseite erhalten sie den Auftrag im Anschluss die Notwendigkeit des Aufteilens in Trainings- und Testdaten und Overfitting in eigenen Worten zu erklären. Da neben der Gesamtgenauigkeit auch Sensitivität und Spezifität je nach Kontext von großer Bedeutung sein können, folgt eine Erklärung zu Sensitivität und Spezifität anhand der gerade auf der Webseite gesehenen Vierfeldertafel und des schon vorher genutzten Streudiagramms.

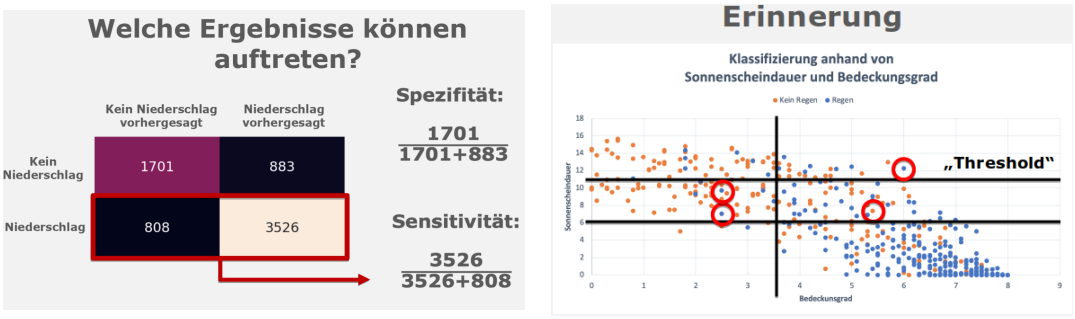


Abbildung 5.4.: Sensitivität und Spezifität am Beispiel

Hierbei wird an die aus der Wahrscheinlichkeitsrechnung aus Klassenstufe 9 bekannten Vierfeldertafeln angeknüpft. Mit Hilfe des Streudiagramms wird der Einfluss des Schwellenwerts auf die Sensitivität und Spezifität verdeutlicht. Anhand des Schwellenwerts wird der Fachbegriff des Parameters eingeführt und es erfolgt die Überleitung zu den weiteren Hyperparametern eines Entscheidungsbaums. Dabei wird die Auswahl der Hyperparameter auf die maximale Tiefe, die maximale Anzahl an berücksichtigten Attributen, die minimale Anzahl an Datensätzen pro Split und die minimale Anzahl an Datensätzen pro Blatt reduziert. Für die Hyperparameter finden sich in der Präsentation Visualisierungen mit deren Hilfe ihr Einfluss auf die Struktur eines Entscheidungsbaums veranschaulicht wird. Um diesen Einfluss weiter zu erkunden, bearbeiten die Schülerinnen und Schüler im Anschluss das Arbeitsblatt zum Tuning der Hyperparameter (Anhang I.3). Dazu nutzen sie erneut die Webseite auf der sie die Werte der Hyperparameter anpassen und systematisch deren Einfluss auf die Gestalt des Baums, die Genauigkeit und die Sensitivität beobachten können. Die Ergebnisse und Beobachtungen werden an der Tafel gesammelt und besprochen. Dabei sollen durch die Schülerinnen und Schüler auch

Erklärungen für die beobachteten Phänomene gegeben werden.
Hieran schließt sich die letzte Phase des Workshops an, in der die ethische Diskussion zum Einsatz des Maschinellen Lernens stattfindet. Die Diskussion wird angeleitet durch die Übertragung der Sensitivität und Spezifität auf Daten von Brustkrebspatienten und die Frage, ob bei medizinischen Diagnosen eine hohe Sensitivität oder eine hohe Spezifität favorisiert werden soll. Nach dieser ersten Diskussionsphase entdecken die Schülerinnen und Schüler ein weiteres moralisches Dilemma, indem sie auf der Webseite hypothetische Patientendatensätze eingeben (siehe Anhang I.4) und verschiedene, bereits vorher trainierte, Modelle für eine Diagnose nutzen. Hierbei werden sie bemerken, dass bei manchen „Patienten“ je nach verwendetem Algorithmus unterschiedliche Diagnosen gestellt werden. Die Diskussion wird anhand dieser Problematik weitergeführt und durch den Tutor je nach noch zur Verfügung stehenden Zeit auch auf andere kritische Einsatzzwecke des Maschinellen Lernens gelenkt. Beispiele hierfür sind der Twitter-Chatbot Tay [Lee16] und das Bewerbungssystem von Amazon welches mit Hilfe Künstlicher Intelligenz Bewerbungen in ein (diskriminierendes) Ranking einordnete [Das18].
Nach der Diskussion wird der Workshop mit einer kurzen Feedbackrunde durch die Teilnehmer und dem erneuten Ausfüllen des Fragebogens zu den Vorstellungen von Künstlicher Intelligenz abgeschlossen.

5.4. Webseite

Die während des Moduls eingesetzte Webseite steht den Schülerinnen und Schülern unter `modulml.cs.uni-saarland.de` zur Verfügung. Die Webseite bietet wie bereits erwähnt einen Erklärungstext zu Overfitting an.
In erster Linie handelt es sich jedoch um ein Tool, mit dem die Teilnehmer selbst Entscheidungsbäume zur Regenvorhersage trainieren können und dabei die im Modul behandelten Hyperparameter sowie die Auswahl der Attribute anpassen können, ohne jedoch die Programmierung selbst vornehmen zu müssen. Um den Einfluss der Hyperparameter beobachten zu können, werden die entstandenen Entscheidungsbäume wie in Abbildung 5.5 angezeigt, sofern ihre Tiefe ≤ 6 ist.

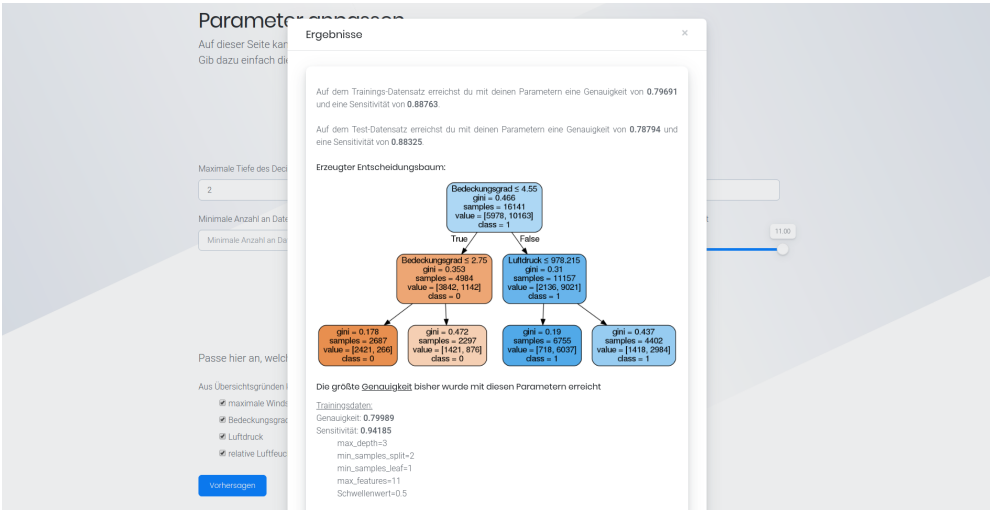


Abbildung 5.5.: Darstellung auf der Webseite

Ab dieser Größe dauert das Rendern des Graphen unverhältnismäßig lange und die Darstellung wird so klein, dass die strukturellen Veränderungen des Baums nicht mehr gut zu erkennen sind. Zusätzlich zur Visualisierung des Entscheidungsbaums werden den Teilnehmern die Genauigkeiten und Sensitivitäten auf den Trainings- und Testdaten mitgeteilt. Bei der Bearbeitung des zum Hyperparameter-Tuning gehörenden Arbeitsblatts wird nach der für die Genauigkeit auf den Testdaten besten Parameterkombination gefragt. Um die Beantwortung dieser Frage zu erleichtern und um den Schülerinnen und Schülern eine Referenz zu ermöglichen, wird ihnen die bisher beste Parameterkombination angezeigt.
Auch die ethische Diskussion zum medizinischen Einsatz des Maschinellen Lernens wird auf der Webseite durch die Bereitstellung mehrerer trainierter Modelle unterstützt, mit deren Hilfe Diagnosen für selbst eingegebene Patientendatensätze gestellt werden können. Dabei stehen folgende Modelle zur Verfügung:

- ein Entscheidungsbaum ohne Optimierungen
- ein auf die Gesamtgenauigkeit optimierter Entscheidungsbaum
- ein Random Forest ohne Optimierungen
- ein auf Sensitivität optimierter Random Forest
- ein K-Nächste-Nachbarn-Modell

Neben dieser Version der Webseite existiert unter `saml.cs.uni-saarland.de` eine weitere Webseite, die die Inhalte des Moduls in gekürzter Form vermittelt. Zum einen

kann diese Variante der Webseite zum Selbststudium eingesetzt werden, wenn kein Tutor zur Verfügung steht. Zum anderen dient sie den Teilnehmern des Moduls im Anschluss als eine Zusammenfassung mit der sie die Grundlagen des Moduls wiederholen können. Des Weiteren kann diese Webseite im Vorfeld des Moduls an Lehrpersonen weitergegeben werden, die sich für eine Teilnahme im Rahmen des Schülerlabors interessieren, um ihnen einen Überblick über die vermittelten Inhalte zu bieten.

6. Didaktische Begründung

Die für das im vorherigen Kapitel beschriebene Unterrichtsmodul relevanten didaktischen Überlegungen sollen im Folgenden beschrieben werden.

6.1. Aufbau

Die in Abschnitt 5.2 formulierten Lernziele lassen sich in der von Krathwohl überarbeiteten Lernzieltaxonomie nach Bloom [Kra02] größtenteils den ersten drei Stufen zuordnen. Implizit ist in allen Lernzielen die Stufe des *Wissens* als Voraussetzung eingeschlossen. Die Lernziele (3) bis (7) entsprechen der zweiten Stufe *Verstehen*. Lernziele (1), (2), (8) und (9) gehören zur Stufe des *Anwendens*. In Lernziel (4) spielt neben dem *Verstehen* auch das *Analysieren* eine Nebenrolle, da die Struktur der Entscheidungsbäume analysiert werden muss, um die unterschiedlichen Bäume zu vergleichen.
Die Auswahl der Inhalte des Moduls zu Maschinellem Lernen folgt dem Grundgedanken des Dagstuhl-Dreiecks zur Bildung in der digitalen vernetzten Welt [Ges16b]. Dieses Dreieck setzt die drei Perspektiven der digitalen vernetzten Welt - die technologische, gesellschaftlich-kulturelle und anwendungsbezogene Perspektive - in Zusammenhang. Die Gesellschaft für Informatik e.V. und die Unterzeichner der Dagstuhl-Erklärung fordern, dass alle drei Perspektiven eines Themas im Unterricht behandelt werden sollen, um eine „fundierte[...] und nachhaltige[...] Bildung in der digitalen vernetzten Welt “ [Ges16b, S.2] zu ermöglichen. Im vorgestellten Modul finden sich alle drei Perspektiven wieder. Die technologische Perspektive wird allgemein durch die Erklärung der generellen Funktionsweise eines Systems welches Maschinelles Lernen nutzt und im Speziellen durch die Erklärungen zu Entscheidungsbäumen vermittelt. Anhand der Diskussion über ethische und moralische Fragestellungen im Zusammenhang mit Maschinellem Lernen wird die gesellschaftlich-kulturelle Perspektive eingenommen und der Einfluss des Maschinellen Lernens auf die Gesellschaft und Individuen betrachtet. Auch die anwendungsbezogene Perspektive spielt im Modul eine Rolle. So nutzen die Teilnehmer an verschiedenen Stellen Modelle des Maschinellen Lernens in unterschiedlichen Anwendungsgebieten. Das Anpassen der Hyperparameter verbindet die technologische mit der anwendungsbezogenen Perspektive.
Um den Schülerinnen und Schülern das weite Themenfeld des Maschinellen Lernens näher zu bringen, ist eine didaktische Reduktion unerlässlich. Die Festlegung auf Ent-

scheidungsbäume als exemplarische Vertreter des Maschinellen Lernens ist dabei sowohl eine qualitative als auch quantitative Reduktion [Gru67], da hierdurch sowohl die Schwierigkeit als auch der Umfang des Stoffs reduziert wird. Die quantitative Reduktion in Form des elementarischen Lernens findet sich bei der Eingabe der Patientendatensätze zu Brustkrebs und der Verwendung verschiedener Algorithmen. Das Ziel hiervon ist nicht das Verständnis für die Funktionsweise der einzelnen Algorithmen, sondern die Bewertung voneinander abweichender maschineller Diagnosen unter ethischen Gesichtspunkten. Innerhalb der Präsentation stellt die Visualisierung der Entscheidungsbäume bei der Erklärung der Hyperparameter mit Hilfe von verallgemeinerten Graphen eine Generalisierung und qualitative Reduktion des Inhalts dar. Die Auswahl der betrachteten Hyperparameter ist wiederum eine exemplarische quantitative Reduktion. Insbesondere die Festlegung auf einen einzelnen Algorithmus als Vertreter des Maschinellen Lernens ist ebenfalls eine quantitative Reduktion des Lernstoffes. Diese Beschränkung auf einen bestimmten Algorithmus wird den Teilnehmern zu Beginn des Moduls offen kommuniziert durch den Überblick über verschiedene Arten des Maschinellen Lernens. Die Wahl von Entscheidungsbäumen ist darin begründet, dass diese leicht nachzuvollziehen sind - sofern sie nicht zu groß werden - und sie gut visualisiert werden können. Hierdurch kann der Prozess der Vorhersage, im Gegensatz zu beispielsweise neuronalen Netzen, von den Schülerinnen und Schülern auf einfache Art und Weise nachvollzogen werden. Des Weiteren kann durch die aus dem Mathematikunterricht bekannten Bäume in der bedingten Wahrscheinlichkeitsrechnung eine Verknüpfung zu bereits bekannten Inhalten hergestellt werden.

Auch wenn die häufig in der Didaktikliteratur zu findende Angabe von maximal 25 Minuten für die Aufmerksamkeitsspanne eines Lernenden (beispielsweise [KSHG+08]) nicht auf empirischen Daten beruht [WK07], so ist es trotz allem sinnvoll, das Vermitteln des Lernstoffs in Phasen zu unterteilen, die zwischen Vortrag des Lehrenden und Aktivität der Lernenden abwechseln. Daher finden sich im Verlauf des Moduls sowohl Phasen, in denen der Tutor neue Inhalte frontal präsentiert, als auch Phasen, in denen die Lernenden selbstständig Aufgaben bearbeiten. Um die Zeiten der frontalen Wissensdarstellung interaktiver zu gestalten, werden den Teilnehmern Zwischenfragen gestellt und diese somit in den Prozess der Wissensvermittlung eingebunden. So wird die Klassifizierung von Daten mit Hilfe eines vorgegebenen Baums, der Entstehungsprozess eines Entscheidungsbaums und die damit zusammenhängende Erklärung wie konkret zu gegebenen Daten ein Entscheidungsbaum von Hand erstellt werden kann gemeinsam mit den Schülerinnen und Schülern erarbeitet. Auch die Erklärung der Sensitivität und Spezifität wird nicht durch den Tutor frontal vorgestellt sondern mit einem Beispiel im Gespräch mit den Teilnehmern entwickelt.

Im Sinne des Genetischen Prinzips [Wag74] wird der Entstehungsprozess des Entscheidungsbaums durch die Lernenden im gemeinsamen Gespräch mit dem Tutor selbst nachvollzogen. Die zentralen Aspekte des Aufbaus und der Wahl des Splits werden von den Schülerinnen und Schülern selbstständig durch die Bearbeitung des Arbeitsblatts in Anhang I.2 und die Beschäftigung mit dem Hunde und Katzen Beispiel entdeckt. Um vor dem komplizierteren Schritt der Entstehung des Baums das Verständnis der Schülerinnen und Schüler zum Nutzen und Sinn von Entscheidungsbäumen zu festigen, sollen sie zunächst in einem Beispiel die Klassifizierung von Daten üben und nachvollziehen. Ebenso wie die Invertierung der Reihenfolge von Entstehung und Nutzung eines Entscheidungsbaums, zielt die Wahl des einfachen Hunde und Katzen Beispiels zum Klassifizieren auf das didaktische Prinzip „vom Einfachen zum Schweren“ ab.

6.2. Arbeitsformen

Das Ziel des Unterrichtsmoduls ist in erster Linie die Vermittlung von fachlichen Kenntnissen zu Maschinellem Lernen. Der generelle Ablauf und damit auch Wahl der Arbeitsformen ist daher auf eine stetige Moderation durch den Lehrenden ausgelegt. Dabei bedeutet diese Moderation zwar, dass die Vermittlung des Wissens häufig frontal stattfindet, jedoch keinesfalls ausschließlich auf diese Weise erfolgt. So findet sich im Unterrichtsmodul eine Mischung aus Einzelarbeit, Partnerarbeit, Tafelarbeit, Diskussionen und Vorträgen des Tutors.
Wird das Modul außerhalb des schulischen Unterrichts im Rahmen des Schülerlabors durchgeführt, soll die Wissensvermittlung in einer lockeren Atmosphäre stattfinden, welche zu einem lernförderlichen Klima und zur Lernfreude beiträgt [Hel07]. Es soll bewusst eine Gelassenheit bei den Schülerinnen und Schülern erzeugt werden, die nicht an eine schulische Unterrichtssituation erinnert. Neben dem Auftreten des Tutors, spielen hierfür auch die im Verlauf des Moduls immer wieder vorkommenden kurzen Diskussionen und Gespräche mit den Teilnehmern eine Rolle. Das Ziel ist es, eventuell vorhandene Hemmungen bei den Schülerinnen und Schülern abzubauen, sodass sie sich an den Diskussionen in der letzten Phase des Moduls beteiligen. Diese leben von einer Beteiligung möglichst aller Teilnehmer. Ein weiterer wesentlicher Bestandteil der zu einer gelockerten Arbeitsatmosphäre beitragen soll, sind die beiden Quiz. Diese sind durch ein Ranking, basierend auf der Schnelligkeit der (korrekten) Beantwortung der Fragen, charakterisiert. Somit kann ein kurzer Wettbewerb entstehen, welcher aufgrund ihrer häufigen Leistungszielorientierung [Kö98] für viele Schülerinnen und Schüler motivierend wirkt und ein „belebendes Element“ [Hey96, S. 112] bedeuten kann. Da jedoch für die Lernmotivation insbesondere die Kooperation wichtig ist [Hel07] und

das Konkurrenzprinzip nicht überbetont werden darf [Hey96], sollen die Quiz von den Schülerinnen und Schülern mehr als Auflockerung und weniger als Leistungsüberprüfung verstanden werden. Dies muss durch den angemessenen Umgang und das Aufgreifen der beim Quiz gemachten Fehler als Lerngelegenheit für alle Teilnehmer durch den Tutor sichergestellt werden. Auch die sonstigen Bestandteile des Moduls verzichten auf einen Wettbewerb. Je nach Verfügbarkeit sollten die Quiz von den Schülerinnen und Schülern an Computern oder Laptops, statt an ihren persönlichen Smartphones, beantwortet werden, um eventuelle Quellen der Ablenkung durch beispielsweise erhaltene Nachrichten zu vermeiden.

6.3. Aufgaben

Das Arbeitsblatt „Decision Trees“ existiert in zwei Versionen, die sich lediglich in der Reihenfolge der Spalten in Aufgabe a) unterscheiden. Je eine Hälfte der Schülerinnen und Schüler bearbeitet eine Version des Arbeitsblatts. Dabei sollen sie nach Möglichkeit während der Bearbeitung noch nicht erkennen, dass die Reihenfolge der Attributspalten in zwei Varianten existiert. Ziel hiervon ist es, dass die beiden Schülergruppen durch die erwartungsgemäße Betrachtung der Attribute von links nach rechts zwei deutlich unterschiedliche Entscheidungsbäume erzeugen. Somit können sie selbst die Mehrdeutigkeit der Entscheidungsbäume nachvollziehen. Wenn strikt von links nach rechts vorgegangen wird, ist der entstehende Baum der einen Schülergruppe wesentlich kleiner als derjenige der anderen. Die Schülerinnen und Schüler können selbst entdecken, dass die Größe eines Entscheidungsbaums eine wichtige Rolle bei der Beantwortung der Frage „Welcher Entscheidungsbaum ist besser?“ spielt. Hierbei sollen sie die beiden von Kubat [Kub15] erwähnten Vorteile der besseren Interpretierbarkeit und der Vermeidung nicht benötigter Splits kleiner Entscheidungsbäume erkennen.
Aufgabe b) des Arbeitsblatts „Decision Trees“ dient der Binnendifferenzierung der Schülergruppe in Form der Aufgabendifferenzierung [Rot09]. Sie ist als Zusatzaufgabe für leistungsfähige Teilnehmer konzipiert. Die Verwendung des nicht binären Attributs „Alter“ erhöht die Anforderungen im Gegensatz zu Aufgabe a) leicht.

7. Auswertung des Moduls

Neben einer eigenen Reflektion der Durchführungen des Unterrichtsmoduls, sollen im Folgenden auch die Rückmeldungen der Teilnehmer dargestellt werden.

7.1. Feedback der Schülerinnen und Schüler

Wie in 5.3 erläutert, wurden zu Beginn und am Ende des Workshops die Fragebögen aus Anhang H von den Schülerinnen und Schülern ausgefüllt. Diese Fragebögen lassen einige interessante Einsichten zu. Der Fragebogen entstand jedoch erst vor der letzten Durchführung, weshalb lediglich 14 Personen über diesen Weg Feedback geben konnten. Vergleicht man die Vorstellungen der Teilnehmer zu Künstlicher Intelligenz, sowie die von ihnen genannten Beispiele vor und nach dem Unterrichtsmodul, so lässt sich feststellen, dass die nach der Durchführung genannten Beispiele realistischer und konkreter sind. So werden beispielsweise aus „Roboter wie in iRobot“, „Die Roboter von Google die KI benutzen“ und „Roboter, SciFi [...]“ nach dem Modul „Machine Learning, Einteilung von Datensätzen, Immer mehr Einsatz im Alltag“, „Medizin, Botanik, Alexa, Siri, Google, Brettspiele mit AI verbessern“, und „Eigenständige Problemlösung von Maschinen“.
Positiv ist auch, dass niemand nach dem Modul die Präsenz des Maschinellen Lernens im Alltag und die Wichtigkeit hierüber ein Grundverständnis zu haben weniger wichtig einschätzt als vorher. Die Zustimmung zur Aussage „In meinem Alltag spielt Maschinelles Lernen eine große Rolle“ stieg bei sechs Personen um eine Stufe und bei jeweils einer Person um zwei oder sogar drei Stufen an. Bei sechs Personen änderte sich die Zustimmung nicht. Bei der Aussage „Es ist wichtig ein Grundverständnis für Maschinelles Lernen zu haben“ stieg die Zustimmung um eine Stufe bei drei Personen während sie bei den restlichen elf Personen unverändert blieb. Von diesen stimmten der Aussage jedoch bereits vorher sechs Personen vollkommen zu.
Das mündliche Feedback der Schülerinnen und Schüler fiel überwiegend positiv aus. Alle Teilnehmer hatten ein grundsätzliches Interesse an Künstlicher Intelligenz und fanden es gut, einen Einblick in einen Teil der zugrundeliegenden Mechanismen zu erhalten. Als besonders gelungen empfanden sie die ihrer Meinung nach anschaulichen Beispiele, welche ihr Verständnis gefördert haben. Insbesondere der Einstieg der mit Hilfe von Googles *Autodraw*, sowie einem Tic-Tac-Toe Beispiel für bestärkendes Lernen erfolgt, gefiel den Schülerinnen und Schülern. Weniger gut gefiel manchen Teilnehmern der eher theoretische Abschnitt zu Sensitivität und Spezifität, auch wenn die meisten der Meinung

waren, dass es nicht zu kompliziert war. Alle stimmten jedoch darin überein, dass das generelle Wissen hierzu bei der ethischen Bewertung des Einsatzes von Maschinellem Lernen förderlich ist. Die Diskussionen am Ende des Moduls empfanden die meisten Schülerinnen und Schüler als einen gelungenen Abschluss. Eine genannte Begründung zu dieser Einschätzung war, dass viele bereits aufgrund der Länge des Moduls ermüdet waren und nicht mehr genügend Konzentration für weitere fachliche Inhalte aufbringen konnten, weshalb die Diskussionen als eine Auflockerung angesehen wurden.

7.2. Reflektion

Aus Perspektive des Tutors wirkten die meisten teilnehmenden Schülerinnen und Schüler von den Inhalten nicht überfordert. Dies zeigte sich auch bei der Bearbeitung der Quiz, da die gegebenen Antworten zum Großteil korrekt waren. Diese Quiz bereiteten den Schülerinnen und Schülern offensichtlich Spaß. Die Mitarbeit in der ersten Phase des Moduls war nur mäßig, da die meisten Teilnehmer wirkten, als hätten sie Hemmungen zu sprechen und dabei falsche Antworten zu geben. Mit Fortschritt des Moduls legten sich diese Hemmungen jedoch. Insbesondere das erste Quiz trug hierzu bei. Im Vergleich zu der ersten Durchführung, bei der die Notwendigkeit von Trainings- und Testdaten nur theoretisch in einem Vortrag erklärt wurde, half auch das gemeinsame Erarbeiten der Aufteilung in Trainings- und Testdaten mit Hilfe des Arbeitsblatts in Anhang I.1, die Interaktivität in der ersten Phase zu erhöhen und somit Hemmungen abzubauen.
Das Arbeitsblatt zum Anpassen der Hyperparameter war sehr hilfreich. Hierdurch war es möglich, den Schülerinnen und Schülern Hilfestellungen zu geben, bei der Suche nach interessanten Beispielwerten. Es zeigte sich, dass ohne diese Hinweise nur wenige Teilnehmer systematisch und zielführend Werte ausprobierten. Eine mögliche Erklärung ist, dass die Schülerinnen und Schüler erst durch die Arbeit mit der Webseite ein erstes Verständnis für die Hyperparameter entwickelten. Mit den gegebenen Hilfen schien dies jedoch bei den meisten Teilnehmern erfolgreich zu sein. Die gefundenen Zusammenhänge zwischen den einzelnen Hyperparametern und den Genauigkeiten und Sensitivitäten konnten von fast allen Teilnehmern korrekt beschrieben und von vielen auch erläutert werden.

8. Zusammenfassung

In dieser Arbeit wurden zunächst Entscheidungsbäume auf Daten von Studierenden der Informatikstudiengänge der Universität des Saarlandes angewandt. Es sollte untersucht werden, ob mit Hilfe von Entscheidungsbäumen die Entscheidung über die Zulassung zum Studiengang unterstützt werden kann.
Zu diesem Zweck wurden Fragebögen zu Persönlichkeitsmerkmalen und schulischen Fächern genutzt, die zu Beginn des Semesters durch Studierende ausgefüllt wurden. Die Korrelationen der einzelnen Items dieser Fragebögen mit den erreichten Noten in den Vorlesungen Programmierung I und Mathematik für Informatiker I lagen jedoch auf einem niedrigen Niveau. Die in der Folge angewandten Entscheidungsbäume konnten keine Ergebnisse erzielen, die für eine Zulassungsentscheidung zum Studium ausreichend sind. Weder durch Anpassen der Hyperparameter, Auswahl bestimmter Items des Fragebogens, noch durch Post-Pruning in Form von Reduced Error Pruning und Critical Value Pruning konnte die Performanz der Bäume wesentlich verbessert werden. Es konnte anhand künstlich erzeugter Daten gezeigt werden, dass wesentlich höhere Korrelationen für den erfolgreichen Einsatz eines Entscheidungsbaums in diesem Kontext notwendig sind. Nach heutigem Stand der Persönlichkeitsforschung in Bezug auf den Erfolg im Studium sind die erreichten Korrelationen jedoch zu erwarten. Dementsprechend scheint eine Vorhersage des Studienerfolgs mit Hilfe eines vergleichbaren Fragebogens nicht zielführend zu sein, sofern Entscheidungsbäume genutzt werden.

Im zweiten Teil der Arbeit wurde ein dreistündiges Unterrichtsmodul zu Maschinellem Lernen für Schülerinnen und Schüler der Oberstufe entworfen und didaktisch begründet. Dieses Modul soll einen Einblick in die Funktionsweise des Maschinellen Lernens geben. Dabei wird bewusst auf die konkrete Programmierung verzichtet, da sich das Unterrichtsmodul an Schülerinnen und Schüler ohne informatische Vorkenntnisse richtet. Dies dient, wie in der Motivation zu Beginn der Arbeit beschrieben, dem Zweck, möglichst vielen Schülerinnen und Schülern die Möglichkeit zu geben, das für die Allgemeinbildung wichtige Themenfeld der Künstlichen Intelligenz kennenzulernen. Um ein Verständnis für die Grundzüge des Maschinellen Lernens zu ermöglichen, wurden Entscheidungsbäume gewählt. Diese sind auch für Lernende ohne Vorkenntnisse verständlich und müssen nicht als ‚Black-Box' behandelt werden. Neben den fachlichen Inhalten wird auch ein verantwortungsvoller Umgang mit Maschinellem Lernen thematisiert und diskutiert.

Die mehrfache Durchführung des Moduls hat ergeben, dass die meisten Teilnehmer Interesse an Maschinellem Lernen haben und positive Rückmeldungen zum Unterrichtsmodul gegeben haben. Das Modul war in der Lage, die Vorstellungen der Teilnehmer zu Künstlicher Intelligenz realistischer zu gestalten. Auch die Wichtigkeit des Maschinellen Lernens im Alltag konnte verdeutlicht werden.

8.1. Ausblick

Derzeit findet eine erste Erprobung eines neuen saarländischen Informatiklehrplans statt, welcher jedoch nicht auf eine größere Anzahl an Unterrichtsstunden ausgelegt ist. Perspektivisch wird im Saarland das Schulfach Informatik sicherlich eine größere Bedeutung erhalten. Damit wird mehr Zeit für den Informatikunterricht zur Verfügung stehen, weshalb eine Überarbeitung des Lehrplans mit neuen Themenfeldern nötig wird. Hier sollte die Künstliche Intelligenz als wichtiges Thema der modernen Informatik aufgenommen werden. Das entworfene Unterrichtsmodul kann hierfür Anregungen liefern.
Bis zur Einführung der Künstlichen Intelligenz im Lehrplan kann durch außerschulisches Lernen ein Eindruck in das Thema vermittelt und das Interesse der Schülerinnen und Schüler geweckt werden. Das in dieser Arbeit entworfene Unterrichtsmodul zu Maschinellem Lernen soll daher im Rahmen des neu eingerichteten Schülerlabors *InfoLabSaar* der Universität des Saarlandes als festes Modul für Oberstufenschülerinnen und -schüler angeboten werden.

Literaturverzeichnis

[BE05] Bergmann, C. ; Eder, Ferdinand: *Allgemeiner Interessen-Struktur-Test (AIST). Testmanual.* Beltz, 2005

[BFSO84] Breiman, L. ; Friedman, J. ; Stone, C.J. ; Olshen, R.A.: *Classification and Regression Trees.* Taylor & Francis, 1984

[BG94] Brown, S. D. ; Gore, P. A.: An Evaluation of Interest Congruence Indices: Distribution Characteristics and Measurement Properties. In: *Vocational Behavior* 45 (1994), Nr. 3, S. 310–327

[Bou08] Bourier, Günther: *Beschreibende Statistik.* Gabler, 2008

[CD14] Chai, Tianfeng ; Draxler, R.R.: Root mean square error (RMSE) or mean absolute error (MAE)?– Arguments against avoiding RMSE in the literature. In: *Geoscientific Model Development* 7 (2014), 06, S. 1247–1250. `http://dx.doi.org/10.5194/gmd-7-1247-2014`. – DOI 10.5194/gmd–7–1247–2014

[Cle14] Cleff, Thomas: *Exploratory Data Analysis in Business and Economics.* Springer, 2014

[CM15] Claesen, Marc ; Moor, Bart D.: Hyperparameter Search in Machine Learning. In: *CoRR* abs/1502.02127 (2015)

[com10] community, SciPy: *SciPy Reference Guide Release 0.8.0.* `https://docs.scipy.org/doc/scipy-0.8.x/scipy-ref.pdf`. Version: 2010. – [05.11.2019]

[Das18] Dastin, Jeffrey: *Amazon scraps secret AI recruiting tool that showed bias against women.* `https://www.reuters.com/article/us-amazon-com-jobs-automation-insight/amazon-scraps-secret-ai-recruiting-tool-that-showed-bias-against-women-idUSKCN1MK08G`. Version: 2018. – [29.10.2019]

[dev19] developers scikit-learn: *scikit-learn user guide Release 0.21.3.* `https://scikit-learn.org/stable/_downloads/scikit-learn-docs.pdf`. Version: 2019. – [05.11.2019]

[Efr79] Efron, B.: Bootstrap Methods: Another Look at the Jackknife. In: *Ann. Statist.* 7 (1979), 01, Nr. 1, 1–26. http://dx.doi.org/10.1214/aos/1176344552. – DOI 10.1214/aos/1176344552

[EMSK97] Esposito, Floriana ; Malerba, Donato ; Semeraro, Giovanni ; Kay, John: A Comparative Analysis of Methods for Pruning Decision Trees. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 19 (1997), 06, S. 476 – 491. http://dx.doi.org/10.1109/34.589207. – DOI 10.1109/34.589207

[Fis88] Fisher, Ronald A.: *UCI Machine Learning Repository*. http://archive.ics.uci.edu/ml. Version: 1988. – [30.10.2019]

[Ges16a] Gesellschaft für Informatik e.V. : Bildungsstandards Informatik für die Sekundarstufe II (Januar 2016). In: *LOG IN* 36 (2016), Nr. 183/184

[Ges16b] Gesellschaft für Informatik e.V.: *Dagstuhl-Erklärung Bildung in der digitalen vernetzten Welt*. https://gi.de/fileadmin/GI/Hauptseite/Themen/Dagstuhl-Erkla__rung_2016-03-23.pdf, 2016. – [07.11.2019]

[Goo90] Goorhuis, Henk: Warum gehört das Thema Künstliche Intelligenz in die Allgemeinbildung? In: Cyranek, Günther (Hrsg.) ; Forneck, Hermann (Hrsg.) ; Goorhuis, Henk (Hrsg.): *Beiträge zur Didaktik der Informatik*. Diesterweg, 1990, Kapitel 6, S. 103–125

[Gru67] Gruner, Gustav: Die didaktische Reduktion als Kernstück der Didaktik. In: *Die deutsche Schule* 59 (1967), S. 414–430

[GWBV02] Guyon, Isabelle ; Weston, Jason ; Barnhill, Stephen ; Vapnik, Vladimir: Gene Selection for Cancer Classification using Support Vector Machines. In: *Machine Learning* 46 (2002), Jan, Nr. 1, S. 389–422

[Gé17] Géron, Aurélien: *Hands-On Machine Learning with Scikit-Learn and TensorFlow*. O'Reilly Media, 2017

[Hel07] Helmke, Andreas: *Was wissen wir über guten Unterricht? Wissenschaftliche Erkenntnisse zur Unterrichtsforschung und Konsequenzen für die Unterrichtsentwicklung*. https://www.bildung.koeln.de/imperia/md/content/selbst_schule/downloads/andreas_helmke_.pdf, 2007. – [07.11.2019]

[Hes16] Hessisches Kultusministerium: *Kerncurriculum gymnasiale Oberstufe Informatik*. https://kultusministerium.hessen.de/sites/default/files/media/kcgo-in.pdf, 2016. – [09.11.2019]

[Hey96] Heymann, Hans W.: *Allgemeinbildung und Mathematik*. Beltz Weinheim, 1996

[Hol97] Holland, John L.: *Making vocational choices: A theory of vocational personalities and work environments*. Psychological Assessment Resources, 1997

[Jol02] Jolliffe, Ian T.: *Principal Component Analysis*. Springer, 2002

[KF12] Kappe, Rutger ; Flier, Henk van d.: Predicting academic success in higher education: what's more important than being smart? In: *European Journal of Psychology of Education* 27 (2012), Nr. 4, S. 605 – 619. http://dx.doi.org/10.1007/s10212-011-0099-9. – DOI 10.1007/s10212–011–0099–9

[Kla93] Klafki, Wolfgang: Allgemeinbildung heute–Grundzüge internationaler Erziehung. In: *Pädagogisches Forum* 1 (1993), S. 21 – 28

[Kra02] Krathwohl, David R.: A Revision of Bloom's Taxonomy: An Overview. In: *Theory into Practice* 41 (2002), Nr. 4

[KSHG+08] Kadmon, Martina ; Strittmatter-Haubold, Veronika ; Greifeneder, Rainer ; Ehlail, Fadja ; Lammerding-Köppel, Maria: Das Sandwich-Prinzip – Einführung in Lerner zentrierte Lehr-Lernmethoden in der Medizin. In: *Zeitschrift für Evidenz, Fortbildung und Qualität im Gesundheitswesen* 102 (2008), Nr. 10, S. 628 – 633

[Kub15] Kubat, Miroslav: *An Introduction to Machine Learning*. Springer, 2015

[Kul04] Kultusministerkonferenz: *Einheitliche Prüfungsanforderungen Informatik, Beschluss der Kultusministerkonferenz vom 01.12.1989 i.d.F. vom 05.02.2004*. https://www.kmk.org/fileadmin/veroeffentlichungen_beschluesse/1989/1989_12_01-EPA-Informatik.pdf, 2004. – [09.11.2019]

[Kul07] Kultusministerkonferenz, Sekretariat der: *Handreichung für die Erarbeitung von Rahmenlehrplänen der Kultusministerkonferenz für den berufsbezogenen Unterricht in der Berufsschule und ihre Abstimmung*

mit Ausbildungsordnungen des Bundes für anerkannte Ausbildungsberufe. https://www.kmk.org/fileadmin/veroeffentlichungen_beschluesse/2007/2007_09_01-Handreich-Rlpl-Berufsschule.pdf, 2007. – [10.12.2019]

[Kö98] Köller, Olaf: *Zielorientierungen und schulisches Lernen*. Waxmann, 1998

[Lan11] Lange, Katharina: *Nichtparametrische Analyse diagnostischer Gütemaße bei Clusterdaten*, Georg-August-Universität Göttingen, Diss., 2011

[LCB] LeCun, Yann ; Cortes, Corinna ; Burges, Christopher J. C.: *The MNIST database of handwritten digits*. http://yann.lecun.com/exdb/mnist/, . – [30.10.2019]

[Lee16] Lee, Peter: *Learning from Tay's introduction*. https://blogs.microsoft.com/blog/2016/03/25/learning-tays-introduction/. Version: 2016. – [29.10.2019]

[Ler19] LernortLabor - Bundesverband der Schülerlabore e.V.: *Schülerlabor-Atlas 2019*. 2019

[LPB06] *Rahmenlehrplan für die gymnasiale Oberstufe Informatik*. https://www.berlin.de/sen/bildung/unterricht/faecher-rahmenlehrplaene/rahmenlehrplaene/mdb-sen-bildung-unterricht-lehrplaene-sek2_informatik.pdf, 2006. – [09.11.2019]

[LPG10] Ministerium für Bildung und Kultur im Saarland (Hrsg.): *Lehrplan Informatik G-Kurs (vierstündig)*. https://www.saarland.de/dokumente/thema_bildung/IN-GOS-4.pdf, 2010. – [09.11.2019]

[LPN19] Ministerium für Bildung und Kultur im Saarland (Hrsg.): *Lehrplan Informatik Gymnasiale Oberstufe Leistungskurs Hauptphase*. https://www.saarland.de/dokumente/thema_bildung/LP_In_HP_LK_2019.pdf, 2019. – [09.11.2019]

[Mad15] Madar, Vered: Direct formulation to Cholesky decomposition of a general nonsingular correlation matrix. In: *Statistics & Probability Letters* 103 (2015), 142 - 147. http://dx.doi.org/https://doi.org/10.1016/j.spl.2015.03.014. – DOI https://doi.org/10.1016/j.spl.2015.03.014

[Min89] Mingers, John: An Empirical Comparison of Pruning Methods for Decision Tree Induction. In: *Machine Learning* 4 (1989), 01, S. 227–243. http://dx.doi.org/10.1023/A:1022604100933. – DOI 10.1023/A:1022604100933

[Min02] Ministerium für Bildung, Wissenschaft, Forschung und Kultur des Landes Schleswig-Holstein: *Lehrplan für die Sekundarstufe II Gymnasium, Gesamtschule, Fachgymnasium Informatik*. https://lehrplan.lernnetz.de/index.php?DownloadID=73, 2002. – [09.11.2019]

[Min16] Ministerium für Bildung und Kultur im Saarland: *Lehrplan Mathematik Gymnasium Klassenstufe 9*. https://www.saarland.de/dokumente/thema_bildung/LP_Ma_Gym_9_2016.pdf, 2016. – [09.11.2019]

[Min18] Ministerium für Bildung, Jugend und Sport Land Brandenburg: *Rahmenlehrplan für den Unterricht in der gymnasialen Oberstufe im Land Brandenburg Informatik*. https://bildungsserver.berlin-brandenburg.de/fileadmin/bbb/unterricht/rahmenlehrplaene/gymnasiale_oberstufe/curricula/2018/RLP_GOST_Informatik_BB_2018.pdf, 2018. – [09.11.2019]

[Mit97] Mitchell, Tom M.: *Machine Learning*. McGraw-Hill, 1997

[MR05] Maimon, Oded ; Rokach, Lior: *Data Mining and Knowledge Discovery Handbook*. Springer, 2005

[OP07] O'Connor, Melissa C. ; Paunonen, Sampo V.: Big Five personality predictors of post-secondary academic performance. In: *Personality and Individual Differences* 43 (2007), Nr. 5, 971 - 990. http://dx.doi.org/https://doi.org/10.1016/j.paid.2007.03.017. – DOI https://doi.org/10.1016/j.paid.2007.03.017. – ISSN 0191–8869

[Qui86] Quinlan, J. R.: Induction of decision trees. In: *Machine Learning* 1 (1986), Mar, Nr. 1, 81–106. http://dx.doi.org/10.1007/BF00116251. – DOI 10.1007/BF00116251

[Qui87] Quinlan, J.R.: Simplifying decision trees. In: *International Journal of Man-Machine Studies* 27 (1987), Nr. 3, 221 - 234. http://dx.doi.org/https://doi.org/10.1016/S0020-7373(87)80053-6. – DOI https://doi.org/10.1016/S0020–7373(87)80053–6. – ISSN 0020–7373

[Qui93] Quinlan, J. R.: *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, 1993

[Rot09] Roth, Ralf: *Hinweise und Anregungen zur Differenzierung*. http://www.gu-thue.de/material/RROTH_Differenzierung.pdf, 2009. – [24.10.2019]

[SHM+16] Silver, David ; Huang, Aja ; Maddison, Chris J. ; Guez, Arthur ; Sifre, Laurent ; Driessche, George van d. ; Schrittwieser, Julian ; Antonoglou, Ioannis ; Panneershelvam, Veda ; Lanctot, Marc ; Dieleman, Sander ; Grewe, Dominik ; Nham, John ; Kalchbrenner, Nal ; Sutskever, Ilya ; Lillicrap, Timothy ; Leach, Madeleine ; Kavukcuoglu, Koray ; Graepel, Thore ; Hassabis, Demis: Mastering the Game of Go with Deep Neural Networks and Tree Search. In: *Nature* 529 (2016), S. 484 – 489

[Smo12] Smolka, Gert: *Programmierung - Eine Einführung in die Informatik mit Standard ML*. De Gruyter, 2012

[SP01] Schuler, Heinz ; Prochaska, Michael: *Leistungsmotivationsinventar*. Hogrefe, 2001

[Sta03] Stadtler, Thomas: *Lexikon der Psychologie*. Kröner, 2003

[Stu08] Student: The Probable Error of a Mean. In: *Biometrika* 6 (1908), Nr. 1, 1–25. http://www.jstor.org/stable/2331554

[THWS07] Trapmann, Sabrina ; Hell, Benedikt ; Weigand, Sonja ; Schuler, Heinz: Die Validität von Schulnoten zur Vorhersage des Studienerfolgs – eine Metaanalyse. In: *Zeitschrift für Pädagogische Psychologie* 21 (2007), Nr. 1, S. 11–27. http://dx.doi.org/10.1024/1010-0652.21.1.11. – DOI 10.1024/1010–0652.21.1.11

[VB12] Vanwinckelen, Gitte ; Blockeel, Hendrik: On estimating model accuracy with repeated cross-validation. In: De Baets, Bernard (Hrsg.) ; Manderick, Bernard (Hrsg.) ; Rademaker, Michaël (Hrsg.) ; Waegeman, Willem (Hrsg.): *BeneLearn 2012: Proceedings of the 21st Belgian-Dutch Conference on Machine Learning, Belgian-Dutch Conference on Machine Learning (BeneLearn), Ghent, 24-25 May 2012*, 2012, 39–44

[Wag74] Wagenschein, Martin: Entdeckung der Axiomatik. In: *Der Mathematikunterricht* 20 (1974), Nr. 1, S. 52 – 70

[Wil27] Wilson, Edwin B.: Probable Inference, the Law of Succession, and Statistical Inference. In: *Journal of the American Statistical Association* 22 (1927), Nr. 158, 209–212. http://www.jstor.org/stable/2276774

[Wit03] Witten, Helmut: Allgemeinbildender Informatikunterricht? Ein neuer Blick auf H. W. Heymanns Aufgaben allgemeinbildender Schulen. In: *INFOS* (2003), S. 59 – 75

[WK07] Wilson, Karen A. ; Korn, James H.: Attention during Lectures: Beyond Ten Minutes, 2007

[WL16] Wasserstein, Ronald L. ; Lazar, Nicole A.: The ASA Statement on p-Values: Context, Process, and Purpose. In: *The American Statistician* 70 (2016), Nr. 2, 129-133. https://doi.org/10.1080/00031305.2016.1154108

Anhang

B. Codebook zur Befragung

Quelle: Lehrstuhl Prof. Dr. Roland Brünken, Universität des Saarlandes

Codebook Befragung Informatik WS 17/18

allgemein:

- Fehlende Werte = -99
-

Variablenname	Bedeutung/Itemwortlaut	Codierung/ Skala
EinVerst	Einverständnis	1 ja 2 nein
AbiLand	Land, in dem man Abi gemacht hat	1 D 2 anderes
AbiLand_s	Anderes Land	Freitext
AbiBL	Bundesland, in dem man Abi gemacht hat	1 Bayern 2 Berlin 3 Brandenburg 4 Bremen 5 Hamburg 6 Hessen 7 Meck-Pomm 8 Niedersachsen 9 NRW 10 Rheinland-Pfalz 11 Saarland 12 Sachsen 13 Sachsen-Anhalt 14 Schleswig-H 15 Thüringen 16 BaWü
AbiSchule	Schule in SL, in dem man Abi gemacht hat	
AbiSchule_s	andere Schule	Freitext
Sex	Geschlecht	1 weiblich 2 männlich
Abinote	(Durchschnittsnote allg. Hochschulreife)	
matrikel	Matr.-Nr.	
Studiengang	Welchen Studiengang studieren Sie?	1 Informatik 2 Medieninformatik 3 Bioinformatik 4 Cybersicherheit 5 Eingebettete Systeme 6 Computer- und Kommunikationstechnik 7 Mathematik und Informatik 8 Nebenfach Informatik

Ausbildung	***Haben Sie in Ihrer Schulzeit folgende Kurse besucht?***	***1 ja 2 nein***
LK_Mat	Leistungskurs Mathematik	
LK_Phy	Leistungskurs Physik	
LK_Inf	Leistungs-/Erweiterungskurs Informatik	
GK_Inf	Grundkurs Informatik	
Kurs_Inf	Wie viele Jahre hatten Sie Informatik in der Schule?	Kommazahl
Stund_Inf	Wie viele Stunden/Woche hatten Sie Informatik in der Oberstufe?	

Selbstkonzept Mathe		
SKMat_1	Ich bin für Mathe	1 nicht begabt 7 sehr begabt
SKMat_2	Neues in Mathe zu lernen fällt mir...	1 schwer 7 leicht
SKMat_3	Ich kann in Mathe...	1 wenig 7 viel
SKMat_4	In Mathe fallen mir viele Aufgaben...	1 schwer 7 leicht
mean_SKMat	***Skalenmittelwert Selbstkonzept Informatik***	

Selbstkonzept Informatik		
SKInf_1	Ich bin für Informatik...	1 nicht begabt 7 sehr begabt
SKInf_2	Neues in Informatik zu lernen fällt mir...	1 schwer 7 leicht
SKInf_3	Ich kann in Informatik...	1 wenig 7 viel
SKInf_4	In Informatik fallen mir viele Aufgaben...	1 schwer 7 leicht
mean_SKInf	***Skalenmittelwert Selbstkonzept Informatik***	

Berufsmotivation Informatik		**In Unipark erfasst:** 1 stimme vollständig zu 4 stimme gar nicht zu **umkodiert in** 1 stimme gar nicht zu 4 stimme vollständig zu (i) invertiert
BM_Inf_1	Informatik bietet mir die Karrieremöglichkeiten, die ich will.	Folgenanreiz Beruf
BM_Inf_2	Informatik bietet mir die Berufsmöglichkeiten, die ich mir wünsche.	Folgenanreiz Beruf
BM_Inf_3	Informatik ist für mich eine intellektuell wissenschaftliche Herausforderung.	Folgenanreiz intellektuell
BM_Inf_4	Sich schnell veränderndes Wissen und neuartige Anwendungen in Informatik fordern mich heraus.	Folgenanreiz intellektuell
~~BM_Inf_5~~	~~Ich freue mich darauf, im Informatikstudium mit anderen zusammenzuarbeiten.~~	~~Tätigkeitsanreiz~~
BM_Inf_6	Informationstechnische Systeme funktionsfähig zu machen, ist eine	Folgenanreiz intellektuell

	reizvolle Herausforderung für mich.	
~~BM_Inf_7~~	~~Das Image des vereinsamten Programmierers gilt nach wie vor.~~	~~Image (i)~~
BM_Inf_8	Informatiker sind Nerds.	Image (i)
BM_Inf_9	Ich bin sicher, dass ich den Anforderungen eines Informatikstudiums gewachsen bin.	Selbskonzept
BM_Inf_10	Programmieren bzw. die Vorstellung davon ist irgendwie Horror für mich.	Selbstkonzept (i)
~~BM_Inf_11~~	~~Ich weiß eigentlich gar nicht genau, was das Berufsfeld des Informatikers ist.~~	~~Folgenanreiz Beruf (i)~~
~~BM_Inf_12~~	~~Informatik ist nicht viel mehr als Programmieren.~~	~~Folgenanreiz Beruf (i)~~
~~BM_Inf_13~~	~~Informatik - das können nur Männer.~~	~~Image (i)~~
BM_Inf_14	In andere Computersysteme einzudringen, ist für mich eine faszinierende Tätigkeit.	Tätigkeitsanreiz
BM_Inf_15	Mich gegen informationstechnische Angriffe zu wehren, will ich unbedingt lernen.	Tätigkeitsanreiz
~~BM_Inf_16~~	~~Das Informatikstudium bildet eigentlich nicht für den späteren Beruf aus.~~	~~Folgenanreiz Beruf (i)~~
BM_Inf_17	Ich beurteile meine Programmierkenntnisse, die ich vor dem Studium hatte, als sehr gut.	Selbstkonzept
BM_Inf_18	Ich schätze meine Kompetenzen auf dem Gebiet der Informatik als sehr hoch ein.	Selbstkonzept
Persönlichkeit		1 sehr unzutreffend 2 eher unzutreffend 3 weder noch 4 eher zutreffend 5 sehr zutreffend **(i) invertiert**
BFI_K_1	... bin eher zurückhaltend, reserviert.	Extraversion (i)
BFI_K_2	... neige dazu, andere zu kritisieren.	Verträglichkeit (i)
BFI_K_3	... erledige Aufgaben gründlich.	Gewissenhaftigkeit
BFI_K_4	... werde leicht deprimiert, niedergeschlagen.	Neurotizismus
BFI_K_6	... bin begeisterungsfähig und kann andere leicht mitreißen.	Extraversion
BFI_K_7	... schenke anderen leicht Vertrauen, glaube an das Gute im Menschen.	Verträglichkeit
BFI_K_8	... bin bequem, neige zur Faulheit.	Gewissenhaftigkeit (i)
BFI_K_9	... bin entspannt, lasse mich durch Stress nicht aus der Ruhe bringen.	Neurotizismus (i)
BFI_K_11	... bin eher der „stille Typ“, wortkarg.	Extraversion
BFI_K_12	... kann mich kalt und distanziert verhalten.	Verträglichkeit (i)
BFI_K_13	... bin tüchtig und arbeite flott.	Gewissenhaftigkeit
BFI_K_14	... mache mir viele Sorgen.	Neurotizismus

BFI_K_16	... gehe aus mir heraus, bin gesellig.	Extraversion
BFI_K_17	... kann mich schroff und abweisend anderen gegenüber verhalten.	Verträglichkeit (i)
BFI_K_18	... mache Pläne und führe sie auch durch.	Gewissenhaftigkeit
BFI_K_19	... werde leicht nervös und unsicher.	Neurotizismus
mean_BFI_K_N	***Skalenmittelwert Neurotizismus***	
mean_BFI_K_G	***Skalenmittelwert Gewissenhaftigkeit***	
mean_BFI_K_V	***Skalenmittelwert Verträglichkeit***	
mean_BFI_K_E	***Skalenmittelwert Extraversion***	
Leistungsmotivationsinventar		In Unpark erfasst: 1 trifft vollständig zu 7 trifft gar nicht zu **umkodiert in** 7 trifft vollständig zu 1 trifft gar nicht zu **(i) invertiert**
LMI_1	Ich habe mir vorgenommen, es beruflich weit zu bringen.	
LMI_2	Ich bin überzeugt davon, dass ich es beruflich zu etwas bringen werde.	
LMI_3	Mir sind Aufgaben lieber, die mir leicht von der Hand gehen, als solche, bei denen ich mich sehr einsetzen muss.	(i)
LMI_4	Ich bin überzeugt, mich bisher in Ausbildung und Beruf mehr engagiert zu haben als meine Kollegen.	
LMI_5	Ich empfinde Befriedigung darüber, meine eigene Leistung zu steigern.	
LMI_6	Aufgaben, bei denen ich nicht ganz sicher bin, ob ich sie lösen kann, reizen mich ganz besonders.	
LMI_7	Es ist mir sehr wichtig, eine verantwortungsvolle Position zu erreichen.	
LMI_8	Wenn ich eine Prüfung ablege, bin ich auch davon überzeugt, dass ich sie bestehe.	
LMI_9	Mein Ehrgeiz ist leicht herauszufordern.	
LMI_10	Ich beschäftige mich besonders gern mit Problemen, bei denen es eine harte Nuss zu knacken gibt.	
LMI_11	Ich bin zuversichtlich, dass meine Leistung die Anerkennung anderer finden wird.	
LMI_12	Ich suche mir gern Aufgaben, an denen ich meine Fähigkeiten prüfen kann.	
LMI_13	Am glücklichsten bin ich mit einer Aufgabe, bei der ich alle meine Kräfte	

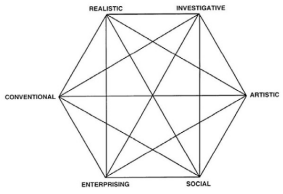
	einsetzen kann.
LMI_14	Wenn mir etwas nicht so gut gelungen ist, wie ich es mir vorgenommen hatte, strenge ich mich anschließend noch mehr an.
LMI_15	Der Wunsch, besser zu sein als andere, ist ein großer Ansporn für mich.
LMI_16	Schwierige Probleme reizen mich mehr als einfache.
LMI_17	Auch wenn ich vor schwierigen Aufgaben stehe, bin ich immer guten Mutes.
LMI_18	Wenn ich mit anderen zusammenarbeite, übernehme ich gewöhnlich die Initiative.
LMI_19	Meine Bekannten würden es als typisch für mich ansehen, dass ich mich durch alle Schwierigkeiten durchbeiße.
LMI_20	Ich empfinde Befriedigung über intensive, konzentrierte Arbeit.
LMI_21	Ich arbeite gern an Aufgaben, die ein hohes Maß an Geschick erfordern.
LMI_22	Es bereitet mir Freude, mich ganz in eine Aufgabe zu vertiefen.
LMI_23	Es ist für mich ein beruflicher Ansporn, einmal eine wichtige Stellung zu erreichen.
LMI_24	Ich glaube, dass ich mich beruflich mehr anstrenge als die meisten meiner Kollegen.
LMI_25	Wenn ich mit anderen zusammenarbeite, nehme ich die Sachen gern selbst in die Hand.
LMI_26	Ich eigne mir lieber neue Kenntnisse an, als mich mit Dingen zu beschäftigen, die ich schon beherrsche.
LMI_27	Wenn ich etwas erreicht habe, lag das vor allem an meinem Geschick und meinen Fähigkeiten.
LMI_28	Durch eine schwierige Aufgabe fühle ich mich besonders herausgefordert.
LMI_29	Wenn ein Risiko besteht, eine Aufgabe nicht zu schaffen, gebe ich mir ganz besondere Mühe.
LMI_30	Es ist mir wichtig, meine Tüchtigkeit zu steigern.
mean_LMI	***Skalenmittelwert Leistungsmotivationsinventar***
Interessen	Die folgenden sechs Persönlichkeitstypen stellen ein Modell für die berufliche Orientierung dar.

	M - Machen: technisch und/oder handwerklich arbeiten **D - Denken:** lernen, forschen, Probleme lösen **K - Kreativ sein:** künstlerisch, innovativ arbeiten **H - Helfen:** Menschen unterstützen, sich sozial engagieren **U - Unternehmen:** Menschen führen, sich durchsetzen **V - Vorstrukturiert arbeiten:** nach vorgegebenen Regeln, feste Arbeitsabläufe	
Der ideale Informatiker (Fremdeinschätzung)	Bitte tragen Sie in die folgenden drei Kästchen die Kennbuchstaben (M, D, K, H, U, V) der drei Typen ein, die für den idealen Informatiker/die ideale Informatikerin am wichtigsten sind Diese Werte sind für eine Vorhersage nicht sinnvoll, sondern die Kongruenzwerte Ktyp_psych und Ktyp_exp (Erläuterung S. 7ff.)	
TypFremd_1	Kennbuchstaben (M, D, K, H, U, V) der drei Typen	An erster Stelle (am wichtigsten)
TypFremd_2	Kennbuchstaben (M, D, K, H, U, V) der drei Typen	An zweiter Stelle
TypFremd_3	Kennbuchstaben (M, D, K, H, U, V) der drei Typen	An dritter Stelle
Eigener Persönlichkeitstyp	Diese Werte sind für eine Vorhersage nicht sinnvoll, sondern die Kongruenzwerte Ktyp_psych und Ktyp_exp (Erläuterung S. 7ff.)	
TypSelbst_1	Kennbuchstaben (M, D, K, H, U, V) der drei Typen	An erster Stelle (am wichtigsten)
TypSelbst_2	Kennbuchstaben (M, D, K, H, U, V) der drei Typen	An zweiter Stelle
TypSelbst_3	Kennbuchstaben (M, D, K, H, U, V) der drei Typen	An dritter Stelle
Kongruenzberechnung (Erläuterung Seite 7ff.)		
Ktyp_psych	Psychosoziale Kongruenz (Passung zur psychosozialen Gruppe der Kommilitonen)	0=Inkongruenz 18=hohe Kongruenz
Ktyp_exp	Anforderungsbasierte Kongruenz (Passung zu Anforderungen des Studiums – Expertenbasiert)	0=Inkongruenz 18=hohe Kongruenz
Feedback		
Feedback	An dieser Stelle haben Sie die Möglichkeit, uns ein Feedback bezüglich der Online-Befragung zu geben (z.B. technische Probleme/ Akzeptanz etc.) sowie Ihre persönliche Einschätzung Ihres Studiums an der Universität des Saarlandes (z.B. bzgl. Studien- und Lehrorganisation/ Inhalte/ Verbesserungsmöglichkeiten etc.) zu äußern.	
Rts-Variablen	Dauer in sec, wie lange die TN auf den entsprechenden Seiten verbracht haben. Ich habe sie erst mal drin	

gelassen. Um sie sinnvoll auswerten zu können, müsste man aber eine genaue Zuordnung vornehmen, was recht aufwendig wäre.

Berechnung des Kongruenzwerts für Informatik-Studierende

Kurzbeschreibung des Konzepts



Hexagonales Modell von Holland (1997)

Bezeichnung Englisch	**Bezeichnung Deutsch**	**Code Erhebung**	**Beschreibung**
R realistic	Praktisch-technisch	M	Machen: technisch und/oder handwerklich arbeiten
I investigative	Intellektuell-forschend	D	Denken: lernen, forschen, Probleme lösen
A artistic	Künstlerisch, sprachlich	K	Kreativ sein: künstlerisch, innovativ arbeiten
S social	Sozial	H	Helfen: Menschen unterstützen, sich sozial engagieren
E Enterprising	Unternehmerisch	U	Unternehmen: Menschen führen, sich durchsetzen
C conventional	konventionell	V	Vorstrukturiert arbeiten: nach vorgegebenen Regeln, feste Arbeitsabläufe

Grundlage für Kongruenzberechnung ist das Hexagonale Modell von Holland (1997). Dabei hat man 6 Interessensbereiche, die im Modell so angeordnet sind, dass sich dadurch inhaltliche Nähe/Distanz ablesen lässt, z.B.: Ordnende Tätigkeiten (conventionell) sind näher an praktisch-technischen Interessen (realistic), aber weit entfernt von künstlerischen Interessen (artistic).

Des Weiteren benötigt man immer die Werte für die eigenen Interessen (Personenwert) und einer Umwelteinschätzung. Diese kann vorgenommen werden von einer Gruppe, die der Person ähnlich ist (z.B. Kommilitonen im Studiengang, Menschen im gleichen Beruf – psychosozial – bei uns Fremdwert) oder von Experten (z.B. Dozenten, die im Studiengang unterrichten – anforderungsbasiert).

Je näher die eigene Einschätzung an der Umwelteinschätzung liegt, desto höher ist die Kongruenz/Übereinstimmung.

Mit höheren Kongruenzwerten werden z.B. eine höhere Zufriedenheit, längere Verweildauer im Beruf bzw. geringere Abbruchquoten im Studium in Zusammenhang gebracht.

Berechnung generell

Zur Berechnung des typologischen C-Index (K_{typ}) nach Brown und Gore (1994) werden die 3 erstgenannten Interessen (der erste, zweite und dritte Buchstabe im Personen- und Umweltprofil) nacheinander anhand des hexagonalen Modells kodiert (von 0 = Inkongruenz, wenn Person- und Umwelttypus im Hexagon gegenüberliegen bis 3 = hohe Kongruenz, wenn Person- und Umwelttypus übereinstimmen); danach wird nach folgender Formel gewichtet: $K_{typ} = 3 (X_i) + 2 (X_i) + (X_i)$, wobei X_i für die Werte von 0 bis 3 steht, die aus den Einzelabgleichen resultieren. Für den typologischen CWert liegen symmetrische Normalverteilungen vor (M = 9; Min = 0; Max = 18; vgl. Brown & Gore, 1994).

Berechnung der typologischen Kongruenz (Brown & Gore) anforderungsbasiert nach Expertenindex aus Handbuch: DMV (Denken, machen, vorstrukturiert)

Umweltcode einstellig	Personencode einstellig		Kongruenzwert (Holland)	
Wert	**Var-Name**	**Wert**	**Var-Name**	**Wert**
D	Typ_Selbst_1	D K oder M V oder H U	X1	3 2 1 0
M	Typ_Selbst_2	M D oder V K oder U H	X2	3 2 1 0
V	Typ_Selbst_3	V U oder M D oder H K	X3	3 2 1 0

Berechnung der typologischen Kongruenz (Brown & Gore) psychosozial aufgrund der Stichprobe

- **Auszählung der Häufigkeiten für die ersten drei Stellen – der jeweils häufigste Buchstabe an dieser Stelle ist dann der Code**

TypFremd_1 An erster Stelle (am wichtigsten)

		Häufigkeit	Prozent	Gültige Prozente	Kumulierte Prozente
Gültig	D	153	69.2	77.3	77.3
	H	4	1.8	2.0	79.3
	K	22	10.0	11.1	90.4
	M	11	5.0	5.6	96.0
	V	8	3.6	4.0	100.0
	Gesamt	198	89.6	100.0	
Fehlend	-99	23	10.4		
Gesamt		221	100.0		

TypFremd_2 An zweiter Stelle

		Häufigkeit	Prozent	Gültige Prozente	Kumulierte Prozente
Gültig	D	27	12.2	13.6	13.6
	F	1	.5	.5	14.1
	H	9	4.1	4.5	18.7
	K	87	39.4	43.9	62.6
	M	46	20.8	23.2	85.9
	U	9	4.1	4.5	90.4
	V	19	8.6	9.6	100.0
	Gesamt	198	89.6	100.0	
Fehlend	-99	23	10.4		
Gesamt		221	100.0		

TypFremd_3 An dritter Stelle

		Häufigkeit	Prozent	Gültige Prozente	Kumulierte Prozente
Gültig	D	13	5.9	6.6	6.6
	H	28	12.7	14.1	20.7
	K	36	16.3	18.2	38.9
	M	57	25.8	28.8	67.7
	U	21	9.5	10.6	78.3
	V	43	19.5	21.7	100.0
	Gesamt	198	89.6	100.0	
Fehlend	-99	23	10.4		
Gesamt		221	100.0		

Umweltcode einstellig	Personencode einstellig		Kongruenzwert (Holland)	
Wert	**Var-Name**	**Wert**	**Var-Name**	**Wert**
D	Typ_Selbst_1	D K oder M V oder H U	X1	3 2 1 0
K	Typ_Selbst_2	K D oder H M oder U V	X2	3 2 1 0
M	Typ_Selbst_3	M D oder V U oder K H	X3	3 2 1 0

C. CV-Scores der Recursive Feature Elimination

C.1. Szenario 1

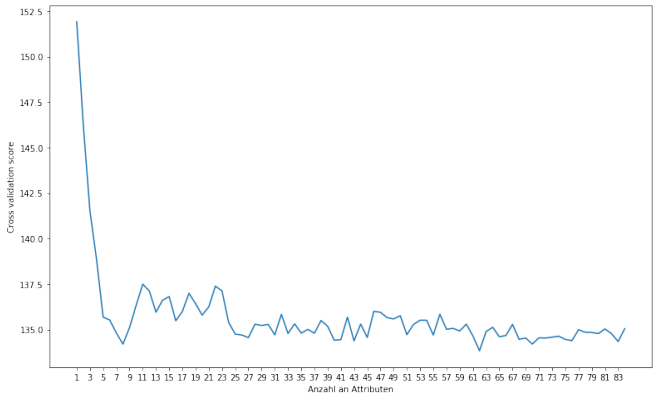


Abbildung 8.1.: Szenario 1, RMSE

C.2. Szenario 1a

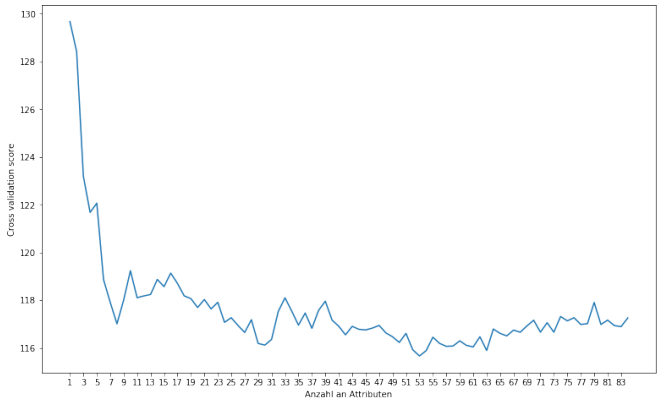


Abbildung 8.2.: Szenario 1a, RMSE

C.3. Szenario 2

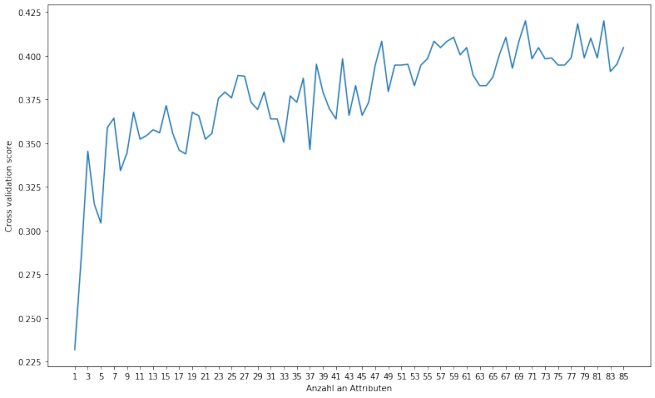


Abbildung 8.3.: Szenario 2, Genauigkeit

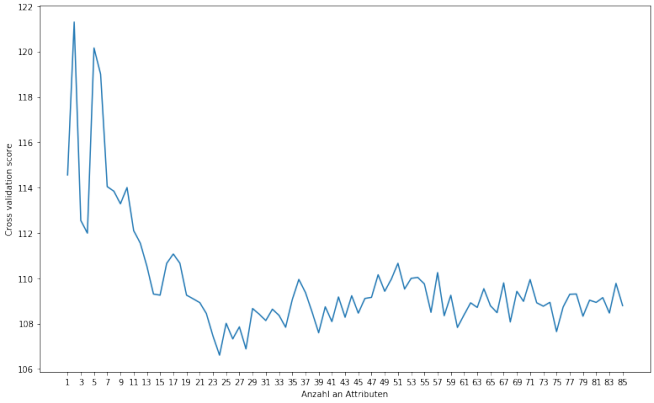


Abbildung 8.4.: Szenario 2, RMSE

C.4. Szenario 3

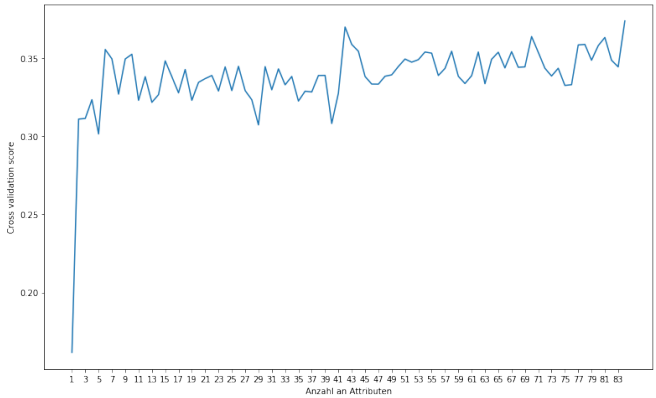


Abbildung 8.5.: Szenario 3, Genauigkeit

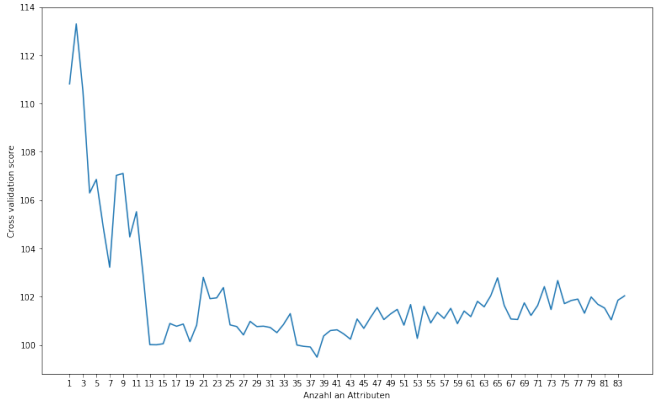


Abbildung 8.6.: Szenario 3, MAE

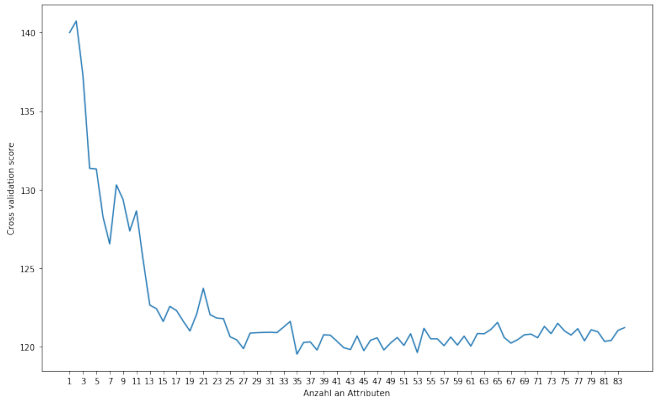


Abbildung 8.7.: Szenario 3, RMSE

C.5. Szenario 4

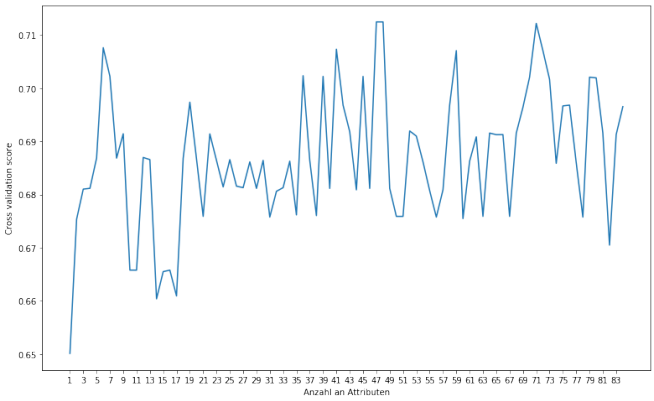


Abbildung 8.8.: Szenario 4, Genauigkeit

C.6. Szenario 5

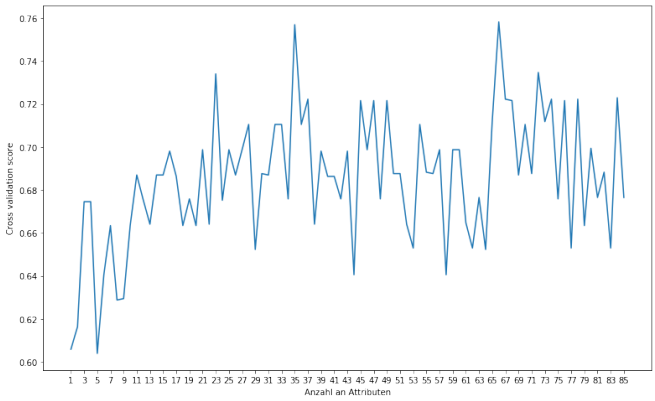


Abbildung 8.9.: Szenario 5, Genauigkeit

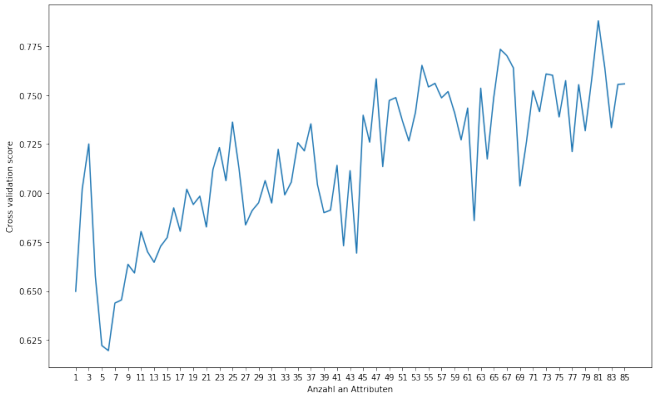


Abbildung 8.10.: Szenario 5, AUC

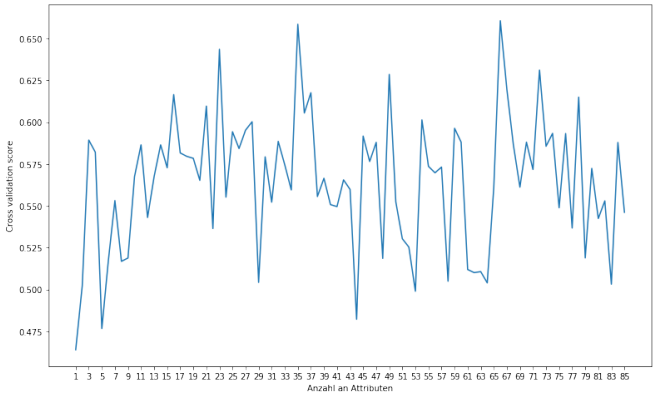


Abbildung 8.11.: Szenario 5, F1

C.7. Szenario 6

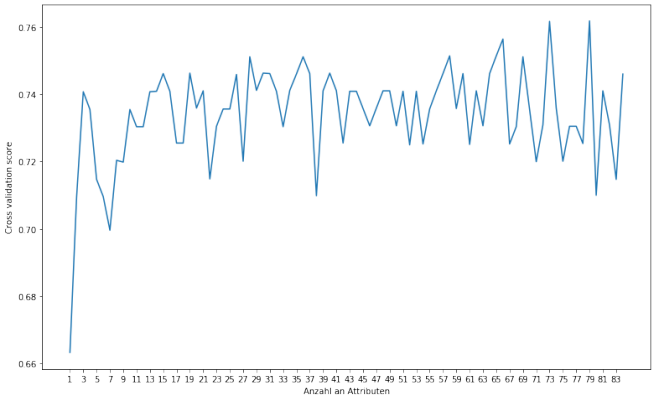


Abbildung 8.12.: Szenario 6, Genauigkeit

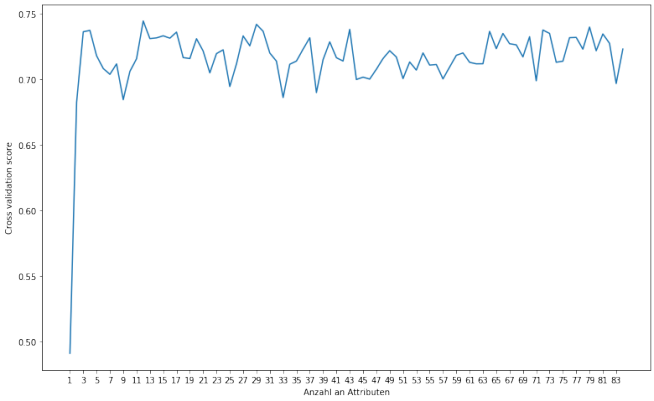


Abbildung 8.13.: Szenario 6, AUC

C.8. Szenario 7

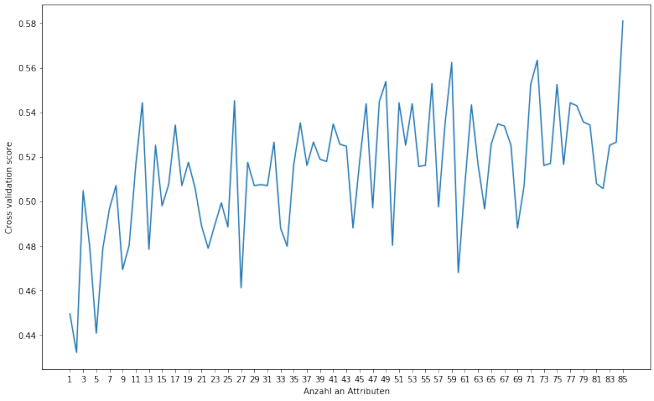


Abbildung 8.14.: Szenario 7, Genauigkeit

D. gewählte Hyperparameter

D.1. Szenario 1

	Splitkriterium	max. Tiefe	max. Attribute	max. Blätter	min. Daten Blatt	min. Daten Split
Genauigkeit	Entropie	3	0.75	5	20	2
RMSE	MSE	4	0.25	5	10	30
MAE	MAE	3	1.0	10	5	30

Tabelle 8.1.: Hyperparameter Szenario 1, alle Attribute

	Splitkriterium	max. Tiefe	max. Attribute	max. Blätter	min. Daten Blatt	min. Daten Split
Genauigkeit	Entropie	4	0.5	5	10	30
RMSE	MSE	7	0.5	10	3	10
MAE	MAE	4	1.0	10	5	30

Tabelle 8.2.: Hyperparameter Szenario 1, ausgewählte Attribute

D.2. Szenario 1a

	Splitkriterium	max. Tiefe	max. Attribute	max. Blätter	min. Daten Blatt	min. Daten Split
RMSE	MSE	3	0.25	10	5	20
MAE	MSE	10	0.25	20	5	20

Tabelle 8.3.: Hyperparameter Szenario 1a, alle Attribute

	Splitkriterium	max. Tiefe	max. Attribute	max. Blätter	min. Daten Blatt	min. Daten Split
RMSE	MSE	4	0.5	10	20	2
MAE	MSE	10	0.25	25	3	10

Tabelle 8.4.: Hyperparameter Szenario 1a, ausgewählte Attribute

D.3. Szenario 2

	Splitkriterium	max. Tiefe	max. Attribute	max. Blätter	min. Daten Blatt	min. Daten Split
Genauigkeit	Entropie	3	0.5	5	20	2
RMSE	MSE	3	0.75	5	10	30
MAE	MAE	4	1.0	10	3	10

Tabelle 8.5.: Hyperparameter Szenario 2, alle Attribute

	Splitkriterium	max. Tiefe	max. Attribute	max. Blätter	min. Daten Blatt	min. Daten Split
Genauigkeit	Entropie	3	0.25	5	5	2
RMSE	MAE	7	0.25	10	3	2
MAE	MAE	7	0.25	10	3	2

Tabelle 8.6.: Hyperparameter Szenario 2, ausgewählte Attribute

D.4. Szenario 3

	Splitkriterium	max. Tiefe	max. Attribute	max. Blätter	min. Daten Blatt	min. Daten Split
Genauigkeit	Entropie	4	0.75	5	20	2
RMSE	MSE	3	0.75	10	10	2
MAE	MAE	3	0.75	10	10	30

Tabelle 8.7.: Hyperparameter Szenario 3, alle Attribute

	Splitkriterium	max. Tiefe	max. Attribute	max. Blätter	min. Daten Blatt	min. Daten Split
Genauigkeit	Entropie	5	0.75	10	1	2
RMSE	MSE	5	0.75	10	10	40
MAE	MAE	5	0.75	20	3	2

Tabelle 8.8.: Hyperparameter Szenario 3, ausgewählte Attribute

D.5. Szenario 4

	Splitkriterium	max. Tiefe	max. Attribute	max. Blätter	min. Daten Blatt	min. Daten Split
Genauigkeit	Entropie	3	0.75	5	20	2
AUC	Entropie	5	0.75	10	10	40
F1	Entropie	4	0.75	10	20	2

Tabelle 8.9.: Hyperparameter Szenario 4, alle Attribute

	Splitkriterium	max. Tiefe	max. Attribute	max. Blätter	min. Daten Blatt	min. Daten Split
Genauigkeit	Entropie	3	0.75	5	20	2
AUC	Entropie	5	0.75	10	10	40
F1	Entropie	4	0.75	10	20	2

Tabelle 8.10.: Hyperparameter Szenario 4, ausgewählte Attribute

D.6. Szenario 5

	Splitkriterium	max. Tiefe	max. Attribute	max. Blätter	min. Daten Blatt	min. Daten Split
Genauigkeit	Entropie	5	0.5	20	1	5
AUC	Gini	4	0.5	10	3	10
F1	Entropie	5	0.5	20	1	5

Tabelle 8.11.: Hyperparameter Szenario 5, alle Attribute

	Splitkriterium	max. Tiefe	max. Attribute	max. Blätter	min. Daten Blatt	min. Daten Split
Genauigkeit	Gini	3	0.75	5	1	30
AUC	Gini	3	0.75	10	3	2
F1	Gini	3	0.75	5	1	30

Tabelle 8.12.: Hyperparameter Szenario 5, ausgewählte Attribute

D.7. Szenario 6

	Splitkriterium	max. Tiefe	max. Attribute	max. Blätter	min. Daten Blatt	min. Daten Split
Genauigkeit	Entropie	7	1.0	20	3	10
AUC	Gini	10	1.0	10	3	20
F1	Entropie	7	1.0	20	1	10

Tabelle 8.13.: Hyperparameter Szenario 6, alle Attribute

	Splitkriterium	max. Tiefe	max. Attribute	max. Blätter	min. Daten Blatt	min. Daten Split
Genauigkeit	Entropie	7	1.0	20	1	5
AUC	Gini	5	0.25	10	3	10
F1	Gini	15	0.25	30	1	2

Tabelle 8.14.: Hyperparameter Szenario 6, ausgewählte Attribute

D.8. Szenario 7

	Splitkriterium	max. Tiefe	max. Attribute	max. Blätter	min. Daten Blatt	min. Daten Split
Genauigkeit	Entropie	4	0.25	5	5	30

Tabelle 8.15.: Hyperparameter Szenario 7, alle Attribute

	Splitkriterium	max. Tiefe	max. Attribute	max. Blätter	min. Daten Blatt	min. Daten Split
Genauigkeit	Entropie	4	0.75	10	1	10

Tabelle 8.16.: Hyperparameter Szenario 7, ausgewählte Attribute

E. Notenverteilungen

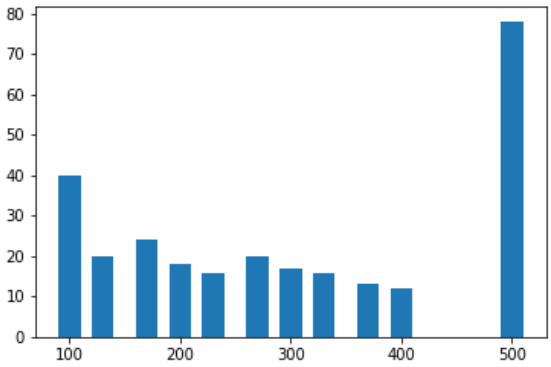


Abbildung 8.15.: Notenverteilung Programmierung 1

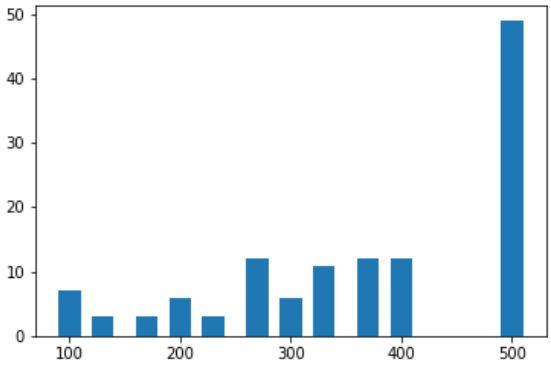


Abbildung 8.16.: Notenverteilung Mathematik für Informatiker 1

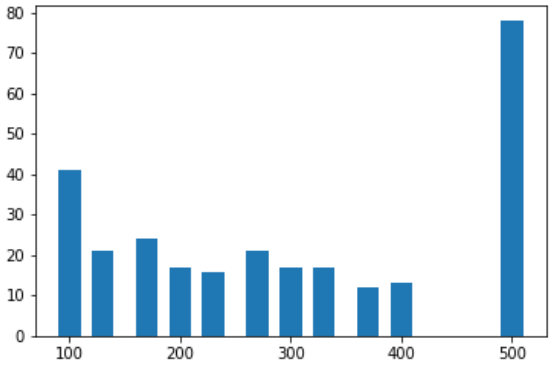


Abbildung 8.17.: Notenverteilung beste Note

F. Ergebnisse Künstliche Daten

Szenario	Genauigkeit Training	Genauigkeit Test	Genauigkeit CV
1.	52.85%	35.71%	46.23%
2.	55.44%	4.76%	50.21%
3.	48.19%	14.29%	47.44%
4.	93.78%	79.76%	83.38%
5.	95.34%	82.14%	87.65%
6.	93.78%	79.76%	83.38%
7.	96.37%	82.14%	85.05%

Tabelle 8.17.: Genauigkeiten künstliche Daten $n = 277$, $\bar{\rho} \approx 0.7$

Szenario	RMSE Training	RMSE Test	RMSE CV	MAE Training	MAE Test	MAE CV
1.	34.39	122.03	80.87	20.31	93.45	57.91
1a.	68.61	92.45	88.77	0.17	71.71	67.51
2.	22.52	96.77	79.07	14.61	57.14	52.23
3.	32.14	128.58	80.96	25.16	111.50	58.37

Tabelle 8.18.: Fehler künstliche Daten $n = 277$, $\bar{\rho} \approx 0.7$

Szenario	AUC Training	AUC Test	AUC CV	F1 Training	F1 Test	F1 CV
4.	0.9649	0.8785	0.8648	89.29%	54.05%	71.60%
5.	0.9966	0.8439	0.9166	94.34%	79.45%	84.41%
6.	0.9649	0.8785	0.8648	89.29%	54.05%	71.60%

Tabelle 8.19.: AUC und F1-Score künstliche Daten $n = 277$, $\bar{\rho} \approx 0.7$

Szenario	Genauigkeit Training	Genauigkeit Test	Genauigkeit CV
1.	51.81%	14.29%	45.94%
2.	55.44%	4.76%	50.73%
3.	48.19%	14.29%	47.44%
4.	98.45%	88.10%	90.13%
5.	97.41%	91.67%	93.76%
6.	98.45%	88.10%	90.13%
7.	95.85%	94.05%	89.72%

Tabelle 8.20.: Genauigkeiten künstliche Daten $n = 277$, $\bar{\rho} \approx 0.8$

Szenario	RMSE Training	RMSE Test	RMSE CV	MAE Training	MAE Test	MAE CV
1.	34.02	117.82	81.45	20.41	87.74	59.28
1a.	1.04	140.36	91.54	0.15	110.12	68.91
2.	22.52	96.77	77.85	14.61	57.14	52.23
3.	36.56	128.58	81.03	16.63	97.72	57.53

Tabelle 8.21.: Fehler künstliche Daten $n = 277$, $\bar{\rho} \approx 0.8$

Szenario	AUC Training	AUC Test	AUC CV	F1 Training	F1 Test	F1 CV
4.	0.9904	0.9444	0.9429	97.44%	73.68%	82.74%
5.	0.9955	0.9349	0.9584	96.86%	89.55%	92.64%
6.	0.9904	0.9444	0.9429	97.44%	73.68%	82.74%

Tabelle 8.22.: AUC und F1-Score künstliche Daten $n = 277$, $\bar{\rho} \approx 0.8$

G. Verlaufsplan

Zeit [min]	**Inhalt**	**Arbeitsform**	**Materialien**
5	Vorstellung Dozenten und Kennenlernen	Gespräch im Plenum	-
7	Abfragen Vorwissen und Vorstellungen zu KI/ML	Gespräch im Plenum, Einzelarbeit	Fragebogen, Stifte
5	Vortrag Beispiele und Arten von ML	Vortrag	Beamer, Präsentation, Browser
10	Anfertigen von Beispieldaten, Diskussion über Aufteilung der Daten	Einzelarbeit, fragend-entwickelndes Gespräch (feG)	AB „Trainings- und Testdaten“, Stifte, Scheren, Kartons
8	Erklärungen überwachtes, unüberwachtes, bestärkendes Lernen	Vortrag	Beamer, Präsentation, TicTacToe
10	Quiz zur Wiederholung	interaktives Quiz mit kompetitivem Charakter	Beamer, Browser, Smartphones
	kurze Pause		
3	Erklärung Baum in der Informatik	Vortrag	Beamer, Präsentation
5	Klassifizierung von Daten mit vorgegebenem Baum an Hund/Katzen-Beispiel	gemeinsame Erarbeitung	Whiteboard/Tafel, Marker/Kreide, Datenkarten, Magnete
10	Erstellen eines Baums zu vorgegebenen Daten	Einzelarbeit	AB „Decision Trees“, Stifte
5	Vergleich der Bäume	feG	AB „Decision Trees“, Beamer, Präsentation
2	Vorstellung Wetterdaten	Vortrag	Beamer, Präsentation
2	Entscheidungsfindung bei Wetterbeispiel	Vortrag	Beamer, Präsentation

Zeit [min]	**Inhalt**	**Arbeitsform**	**Materialien**
10	Erklärung Entstehungsprozess des Baums	Vortrag, feG	Beamer, Präsentation, Hund/Katzen-Beispiel
7	Erklärung Erstellen eines Baums zu vorgegebenen Daten	Erarbeitung im Plenum	Beamer, Präsentation, Whiteboard/Tafel, Marker/Kreide
	Pause		
10	Quiz zur Wiederholung und erneutem Einstieg	interaktives Quiz mit kompetitivem Charakter	Beamer, Browser, Smartphones
10	Lesen der Erklärung zu Overfitting und Besprechung	Einzelarbeit, Gespräch im Plenum	Laptops, Webseite
2	Zusammenfassung der bisherigen Schritte	Vortrag	Beamer, Präsentation
10	Erklärung Sensitivität/Spezifität mit Threshold-Beispiel	Vortrag, feG	Beamer, Präsentation
5	Erklärung Hyperparameter mit Beispielen	Vortrag	Beamer, Präsentation
15	Erarbeitung Einfluss der Hyperparameter	Partnerarbeit	Laptops, Webseite, AB „Parameter“, Stifte
7	Besprechung der Ergebnisse	Gespräch im Plenum	Whiteboard/Tafel, Marker/Kreide, AB „Parameter“
	Pause		
3	Überleitung zur ethischen Diskussion durch Sensitivität/Spezifität bei med. Daten	Vortrag	Beamer, Präsentation
5	erste Diskussion	moderiertes Gespräch im Plenum	-
5	Entdecken des Einfluss verschiedener Algorithmen	Partnerarbeit	Laptops, Webseite, AB „Brustkrebsdiagnose “

Zeit [min]	**Inhalt**	**Arbeitsform**	**Materialien**
10	vertiefende Diskussion und Ausweitung der Diskussion	moderiertes Gespräch im Plenum	-
7	Abschluss, mündliches Feedback, erneute Befragung nach Vorstellungen zu KI/ML	Gespräch im Plenum, Einzelarbeit	Fragebogen
2	Verabschiedung	-	-

Tabelle 8.23.: Verlaufsplan

H. Fragebogen zu Vorstellungen der Studierenden

Künstliche Intelligenz
Maschinelles Lernen

InfoLabSaar

Bitte beantworten Sie die folgenden Fragen spontan.
Es gibt keine richtigen oder falschen Antworten!

1) An was denken Sie bei dem Begriff 'Künstliche Intelligenz'?

2) Welche Beispiele fallen Ihnen zu Künstlicher Intelligenz ein?

3) Kreuzen Sie bitte an, wie sehr Sie den folgenden Aussagen zustimmen!

	Stimme überhaupt nicht zu	Stimme eher nicht zu	Weder noch	Stimme eher zu	Stimme vollkommen zu
a) In meinem Alltag spielt Maschinelles Lernen eine große Rolle					
b) es ist wichtig ein Grundverständnis für Maschinelles Lernen zu haben					

Um Ihre Antworten zu vergleichen, bitten wir Sie im Folgenden ein nach dem erklärten Muster erstelltes Pseudonym einzutragen. Uns ist es nicht möglich die Antworten bestimmten Personen zuzuordnen.
Muster: 1. Erster Buchstabe Ihres Vornamens, 2. Geburtsmonat als Zahl, 3. Letzter Buchstabe des Namens Ihrer Mutter
Beispiel: Karl ist im Dezember geboren und seine Mutter heißt Margret. Sein Code lautet **K12T**

Ihr Code: _____

UNIVERSITÄT DES SAARLANDES | SIC Saarland Informatics Campus

I. Arbeitsblätter

I.1. Arbeitsblatt Trainings- und Testdaten

InfoLabSaar

Trainings- und Testdaten

Um handgeschriebene Zahlen automatisch zu erkennen, benötigt der Computer viele Beispiele von verschiedenen Personen. Trage in die Felder jeweils die angegebene Zahl ein. Schneide dann die einzelnen Felder so aus, dass die Labels am jeweiligen Feld bleiben.

Label: 0 **Label: 1** **Label: 2**

Label: 3 **Label: 4** **Label: 5**

Label: 6 **Label: 7** **Label: 8**

Label: 9

I.2. Arbeitsblatt Decision Trees

Version A

Decision Trees

InfoLabSaar

Erstelle zu den angegebenen Daten jeweils einen Decision Tree der die Frage beantwortet.

a) Handelt es sich um einen guten Arbeitgeber?

= Label

hoher Lohn	gutes Arbeitsklima	Weihnachtsgeld	Überstunden	guter Arbeitgeber
ja	nein	ja	ja	nein
ja	ja	ja	nein	ja
ja	ja	nein	nein	ja
ja	nein	nein	nein	nein
nein	ja	nein	nein	ja
nein	nein	nein	ja	nein
nein	ja	ja	nein	ja
nein	ja	nein	ja	nein

b) Ist die Person fit?

= Label

Alter	Geschlecht	macht Sport	isst FastFood	fit
20	w	nein	nein	ja
45	w	ja	nein	ja
50	w	nein	ja	nein
15	w	nein	ja	nein
33	m	ja	ja	ja
64	m	ja	ja	nein
11	m	nein	ja	nein
25	m	nein	nein	ja

UNIVERSITÄT DES SAARLANDES | SIC Saarland Informatics Campus

Version B

Decision Trees

InfoLabSaar

Erstelle zu den angegebenen Daten jeweils einen Decision Tree der die Frage beantwortet.

a) Handelt es sich um einen guten Arbeitgeber?

= Label

gutes Arbeitsklima	Überstunden	Weihnachtsgeld	hoher Lohn	guter Arbeitgeber
nein	ja	ja	ja	nein
ja	nein	ja	ja	ja
ja	nein	nein	ja	ja
nein	nein	nein	ja	nein
ja	nein	nein	nein	ja
nein	ja	nein	nein	nein
ja	nein	ja	nein	ja
ja	ja	nein	nein	nein

b) Ist die Person fit?

= Label

Alter	Geschlecht	macht Sport	isst FastFood	fit
20	w	nein	nein	ja
45	w	ja	nein	ja
50	w	nein	ja	nein
15	w	nein	ja	nein
33	m	ja	ja	ja
64	m	ja	ja	nein
11	m	nein	ja	nein
25	m	nein	nein	ja

UNIVERSITÄT DES SAARLANDES | SIC Saarland Informatics Campus

I.3. Arbeitsblatt Hyperparameter

InfoLabSaar

Parameter Tuning

Ziel der Arbeitsphase:
Finde die Kombination an Parametern, mit der du die größte Genauigkeit auf den Testdaten erreichst. Finde heraus, welchen Einfluss die Parameter jeweils auf die Genauigkeit haben. Welche Zusammenhänge kannst du erkennen?

Hyperparameter:

Maximale Tiefe des Baums
Bestimmt die maximale Anzahl an Aufteilungen der Daten. Standardwert: So lange aufteilen, bis entweder alle Blätter nur eine Kategorie enthalten oder bis die minimale Anzahl Datensätze pro Split erreicht ist.

Minimale Anzahl Datensätze pro Split
Bestimmt die minimale Anzahl an Datensätzen die in einem inneren Knoten vorhanden sein müssen, damit er geteilt werden darf. Standardwert: 2.

Minimale Anzahl Datensätze pro Blatt
Bestimmt die minimale Anzahl an Datensätzen die in einem Blatt vorhanden sein müssen. Standardwert: 1.

Maximale Anzahl an Features
Bestimmt die maximale Anzahl an Features die bei der Suche nach dem besten Split berücksichtigt werden sollen. Standardwert: 1.

Tipps:

1. Verändere einzelne Werte. Ignoriere zunächst den Schwellenwert und die Featureauswahl.
2. Betrachte sowohl große als auch kleine Werte.
3. Arbeite mit deinem Nachbarn zusammen.
4. Notiere deine Ergebnisse.
5. Fallen dir Erklärungen für deine Beobachtungen ein?
6. Welchen Einfluss hat es, bestimmte Features an- und abzuwählen?
7. Wie ist der Einfluss des Schwellenwerts auf die Genauigkeit zu erklären?

Im Folgenden kannst du deine Parameterkombinationen und die zugehörigen Genauigkeiten eintragen. Du findest auch einige Vorschläge, die du ausprobieren kannst.

Max. Tiefe	
Min. Datens. pro Split	
Min. Datens. pro Blatt	
Max. Features	
Genauigkeiten	
Trainingsdaten	
Testdaten	

Max. Tiefe	
Min. Datens. pro Split	
Min. Datens. pro Blatt	
Max. Features	
Genauigkeiten	
Trainingsdaten	
Testdaten	

Max. Tiefe	
Min. Datens. pro Split	
Min. Datens. pro Blatt	
Max. Features	
Genauigkeiten	
Trainingsdaten	
Testdaten	

Max. Tiefe	
Min. Datens. pro Split	
Min. Datens. pro Blatt	
Max. Features	
Genauigkeiten	
Trainingsdaten	
Testdaten	

Max. Tiefe	
Min. Datens. pro Split	
Min. Datens. pro Blatt	
Max. Features	
Genauigkeiten	
Trainingsdaten	
Testdaten	

Max. Tiefe	
Min. Datens. pro Split	
Min. Datens. pro Blatt	
Max. Features	
Genauigkeiten	
Trainingsdaten	
Testdaten	

Max. Tiefe	
Min. Datens. pro Split	
Min. Datens. pro Blatt	
Max. Features	
Genauigkeiten	
Trainingsdaten	
Testdaten	

Max. Tiefe	
Min. Datens. pro Split	
Min. Datens. pro Blatt	
Max. Features	
Genauigkeiten	
Trainingsdaten	
Testdaten	

Max. Tiefe	
Min. Datens. pro Split	
Min. Datens. pro Blatt	
Max. Features	
Genauigkeiten	
Trainingsdaten	
Testdaten	

Max. Tiefe	
Min. Datens. pro Split	
Min. Datens. pro Blatt	
Max. Features	
Genauigkeiten	
Trainingsdaten	
Testdaten	

Max. Tiefe	
Min. Datens. pro Split	
Min. Datens. pro Blatt	
Max. Features	
Genauigkeiten	
Trainingsdaten	
Testdaten	

Max. Tiefe	
Min. Datens. pro Split	
Min. Datens. pro Blatt	
Max. Features	
Genauigkeiten	
Trainingsdaten	
Testdaten	

Max. Tiefe	
Min. Datens. pro Split	
Min. Datens. pro Blatt	
Max. Features	
Genauigkeiten	
Trainingsdaten	
Testdaten	

Max. Tiefe	
Min. Datens. pro Split	
Min. Datens. pro Blatt	
Max. Features	
Genauigkeiten	
Trainingsdaten	
Testdaten	

Max. Tiefe	
Min. Datens. pro Split	
Min. Datens. pro Blatt	
Max. Features	
Genauigkeiten	
Trainingsdaten	
Testdaten	

Max. Tiefe	
Min. Datens. pro Split	
Min. Datens. pro Blatt	
Max. Features	
Genauigkeiten	
Trainingsdaten	
Testdaten	

Max. Tiefe	1
Min. Datens. pro Split	2
Min. Datens. pro Blatt	1
Max. Features	1
Genauigkeiten	
Trainingsdaten	
Testdaten	

Max. Tiefe	15
Min. Datens. pro Split	2
Min. Datens. pro Blatt	1
Max. Features	1
Genauigkeiten	
Trainingsdaten	
Testdaten	

Max. Tiefe	20
Min. Datens. pro Split	2
Min. Datens. pro Blatt	1
Max. Features	1
Genauigkeiten	
Trainingsdaten	
Testdaten	

Max. Tiefe	-
Min. Datens. pro Split	2
Min. Datens. pro Blatt	1
Max. Features	11
Genauigkeiten	
Trainingsdaten	
Testdaten	

Max. Tiefe	-
Min. Datens. pro Split	50
Min. Datens. pro Blatt	1
Max. Features	11
Genauigkeiten	
Trainingsdaten	
Testdaten	

Max. Tiefe	-
Min. Datens. pro Split	100
Min. Datens. pro Blatt	1
Max. Features	11
Genauigkeiten	
Trainingsdaten	
Testdaten	

Max. Tiefe	-
Min. Datens. pro Split	2
Min. Datens. pro Blatt	10
Max. Features	11
Genauigkeiten	
Trainingsdaten	
Testdaten	

Max. Tiefe	-
Min. Datens. pro Split	2
Min. Datens. pro Blatt	50
Max. Features	11
Genauigkeiten	
Trainingsdaten	
Testdaten	

Max. Tiefe	-
Min. Datens. pro Split	2
Min. Datens. pro Blatt	100
Max. Features	11
Genauigkeiten	
Trainingsdaten	
Testdaten	

InfoLabSaar

Parameter Tuning

Max. Tiefe	
Min. Datens. pro Split	
Min. Datens. pro Blatt	
Max. Features	
Genauigkeiten	
Trainingsdaten	
Testdaten	

←

Ausgewählte Features
.
.
.
.
.

Max. Tiefe	
Min. Datens. pro Split	
Min. Datens. pro Blatt	
Max. Features	
Genauigkeiten	
Trainingsdaten	
Testdaten	

←

Ausgewählte Features
.
.
.
.
.

Max. Tiefe	
Min. Datens. pro Split	
Min. Datens. pro Blatt	
Max. Features	
Genauigkeiten	
Trainingsdaten	
Testdaten	

←

Ausgewählte Features
.
.
.
.
.

Max. Tiefe	
Min. Datens. pro Split	
Min. Datens. pro Blatt	
Max. Features	
Genauigkeiten	
Trainingsdaten	
Testdaten	

←

Ausgewählte Features
.
.
.
.
.

Max. Tiefe	
Min. Datens. pro Split	
Min. Datens. pro Blatt	
Max. Features	
Genauigkeiten	
Trainingsdaten	
Testdaten	

←

Ausgewählte Features
.
.
.
.
.

I.4. Arbeitsblatt Brustkrebsdiagnose

Brustkrebsdiagnose

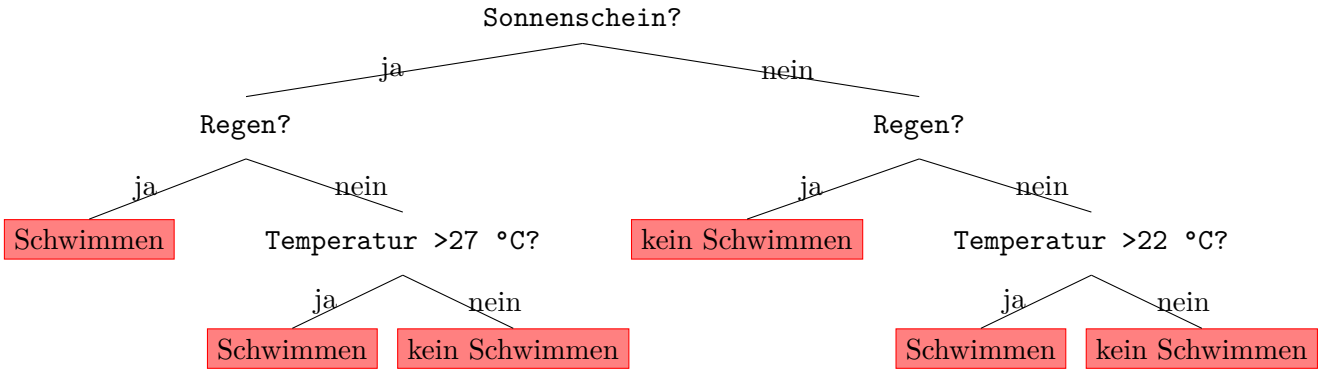
Hier findest du verschiedene Beispiele für Patientendaten.

Gib diese Daten auf der Webseite ein und beobachte die Vorhersagen.
Was fällt dir auf, wenn du verschiedene Algorithmen benutzt?

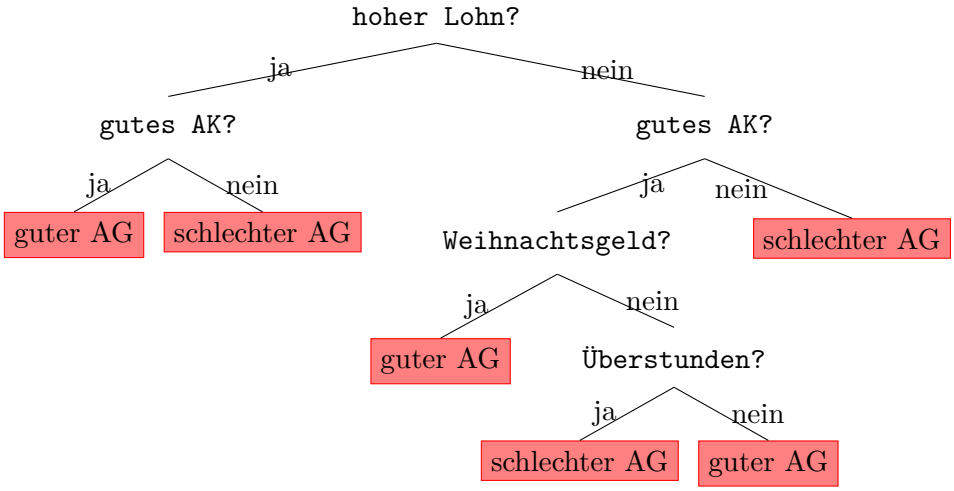
	Mittelwerte				Standardabweichungen		schlechtester Wert		
	Textur	Flächeninhalt	Gleichm. Radiusl.	Konkavität	Flächeninhalt	Konkavität	Gleichm. Radiusl.	Konkavität	Symmetrie
Patient 1	14.00	100.00	0.00	0.00	6.00	0.00	0.14	0.00	0.47
Patient 2	40.00	1003.00	0.20	0.00	550.00	0.50	0.23	1.25	0.67
Patient 3	22.00	2500.00	0.00	0.00	6.00	0.00	0.20	0.25	0.40
Patient 4	9.00	2500.00	0.00	0.00	6.00	0.35	0.15	0.25	0.26

J. Mögliche Entscheidungsbäume

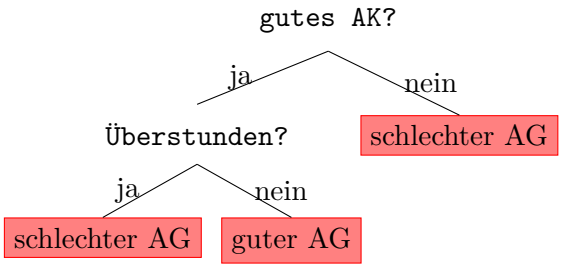
J.1. Entscheidungsbaum zu Beispiel 5.1



J.2. Entscheidungsbaum zu Arbeitsblatt I.2 a) Version A

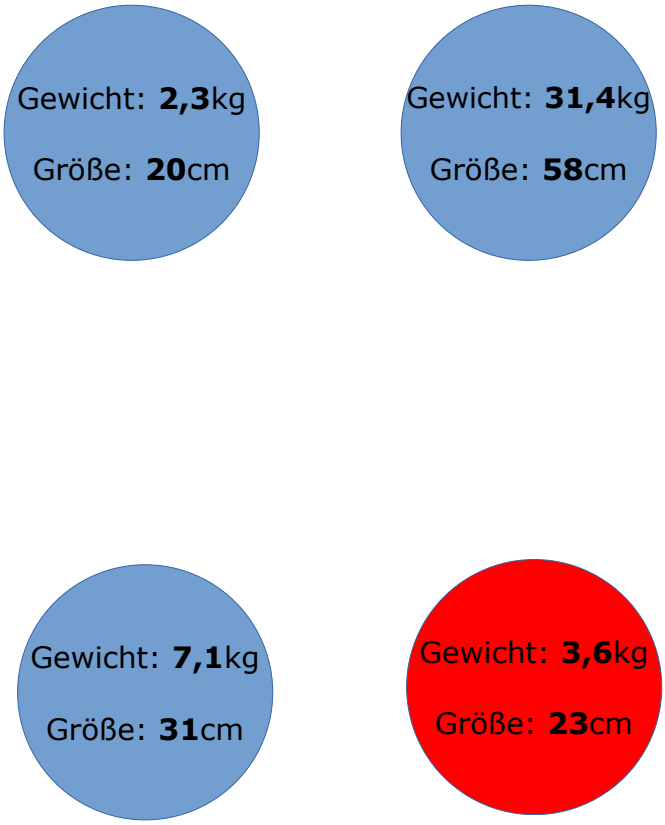


J.3. Entscheidungsbaum zu Arbeitsblatt I.2 a) Version B



K. Sonstige Materialien

K.1. Beispiel Decision Tree



zugehöriger Decision Tree