IMG60-LB-106

Klassifikation mit R

Laborbericht

eingereicht bei

Prof. Dr. Paul Kirchberg

AKAD

von

Justin Stange-Heiduk

Hengstrücken 132

37520 Osterode am Harz

Telefon: 015233817587

Studiengang: Data Science

3. Fachsemester

Matrikelnummer: 8149363

Datum: 18.08.2024

[1. Einleitung 1](#_Toc174865617)

[2. Theoretische Grundlagen und Hintergründe 2](#_Toc174865618)

[2.1 Klassifikationsaufgaben 2](#_Toc174865619)

[2.2 Klassifikationsverfahren: Entscheidungsbaum 2](#_Toc174865620)

[3. Entscheidungsbaum-Klassifikation mit R 5](#_Toc174865621)

[3.1 Knowledge Discovery in Database 5](#_Toc174865622)

[3.2 Datenexploration 6](#_Toc174865623)

[3.3 Datenvorbereitung und -verarbeitung 8](#_Toc174865624)

[3.4 Trainings- und Testdatensplit 13](#_Toc174865625)

[3.5 Entscheidungsbaum Implementierung 14](#_Toc174865626)

[3.6 Anwendung des Entscheidungsbaums 23](#_Toc174865627)

[3.7 Entscheidungsbaum Regeln 28](#_Toc174865628)

[3.8 Genauere Untersuchung der Ergebnisse 30](#_Toc174865629)

[4. Fazit 32](#_Toc174865630)

[Literaturverzeichnis 33](#_Toc174865631)

# Einleitung

In der modernen Geschäftswelt sind präzise Datenanalysen und fundierte Entscheidungsfindung entscheidend für den Erfolg eines Unternehmens. Dies gilt insbesondere für Unternehmen wie HaMa-Cycle, einen führenden Hersteller von Fahrrädern, der kontinuierlich seine Prozesse optimieren und sein Marketing effizienter gestalten möchte. Die Fallstudie IMG602 bietet einen umfassenden Einblick in die Big-Data-Umgebung von HaMa-Cycle und illustriert, wie Business-Intelligence-Analysen zur Verbesserung der Entscheidungsfindung beitragen können.

Ein zentrales Element dieser Analyse ist die Klassifikation von Kundendaten zur Vorhersage der Preiskategorie gekaufter Fahrräder. Die Klassifikation zielt darauf ab, anhand von Kundenmerkmalen Regeln zu identifizieren, die es ermöglichen, die Preiskategorie (niedrig, mittel, hoch) der gekauften Fahrräder präzise vorherzusagen. Hierbei wird ein Entscheidungsbaum als Klassifikationsverfahren verwendet, welches sich durch seine Einfachheit und Interpretierbarkeit auszeichnet.

In dieser Untersuchung werden zunächst die theoretischen Grundlagen und die Funktionsweise von Entscheidungsbäumen erläutert. Anschließend wird beschrieben, wie diese Methode mithilfe der Programmiersprache R auf die spezifischen Daten von HaMa-Cycle angewandt wird. Die Analyse umfasst die Datenexploration, -vorbereitung und -verarbeitung sowie das Training und die Testung des Modells. Ziel ist es, die Effektivität des Entscheidungsbaums zu bewerten und seine Fähigkeit zu überprüfen, stabile und verlässliche Vorhersagen für die Preiskategorie der Fahrräder zu liefern.

# Theoretische Grundlagen und Hintergründe

Zu Beginn dieses Projektberichts werden die notwendigen theoretischen Grundlagen und Hintergründe aufgearbeitet, um den Grundbaustein für das nachfolgende Kapitel zu legen. Dabei wird zunächst die Idee und der Relevanz hinter dem hier behandelten Thema, der Klassifikation im Bereich Maschinelles Lernen, dargestellt. Anschließend wird konkreter auf den hier verwendeten Klassifikationsverfahren den Entscheidungsbaum eingegangen.

## Klassifikationsaufgaben

Klassifikationen gehören zu den zentralen Klassen möglicher Ziele im Data Mining (Alpar & Niedereichholz, 2000, S. 9-12). Die Defintion der Klassifikation nach Müller und Lenz ist folgende 2013, S. 95):

* Die Klassifikation hat das Ziel, betrachtete Objekte anhand ihrer Merkmale in vorab definierten diskreten Klassen einzuordnen. Diese Klassen sind vorher festgelegt und basieren auf bestimmten Eigenschaften der Objekte. Die Methode der Klassifikation wird häufig in Form von Regeln ausgedrückt, die als Klassifikator bezeichnet werden. Der Klassifikator entscheidet, welche Klasse für ein bestimmtes Objekt am besten geeignet ist.

Wenden wir diese Definition für die Fallstudie an, dann lautet die Klassifikationsaufgabe, die Kundenmerkmale des Fahrradherstellers HaMa-Cycle mithilfe des Entscheidungsbaumalgorithmus in den jeweiligen vorhandenen Kategorien, hier die Preiskategorie, einzuteilen.   
Das Ziel ist es, durch die Verwendung von Kundenmerkmalen besseres Marketing betreiben zu können. Dadurch, dass anhand der Kundenmerkmale Rückschlüsse auf die Preiskategorie bzw. besser formuliert auf die Zahlungsbereitschaft der Kunden gegeben werden können, ist ein Unternehmen in der Lage, gezieltere und effektivere Marketingmaßnahmen zu entwickeln. Dies ist besonders wichtig, da es die Möglichkeit bietet, die Bedürfnisse und Präferenzen der Kunden besser zu verstehen und darauf abgestimmte Angebote zu erstellen.

## Klassifikationsverfahren: Entscheidungsbaum

Entscheidungsbäume können für Klassifikations- aber auch Regressionaufgaben verwednet werden. Bei der Regression wird im Gegensatz zur Klassifikation kein Klassifikator bestimmt, sondern ein konkreter Wert prognostiziert (z.B. die Vorhersage der Temperatur). Auch hier erfolgt das Training der Algorithmen anhand von Datensätzen, die das zugrunde liegende Problem repräsentieren. (Bertram & Paschek, 2020, S. 6ff)

Klassifikationsalgorithmen werden verwendet, um Klassifikatoren anhand von Trainingsdaten zu entwickeln. Die Trainingsdaten bestehen aus Objektmerkmalen, in diesem Fall den Kundenmerkmalen verschiedener Kunden, und den jeweiligen Kategorien, hier den Preiskategorien. Da Klassifikationsverfahren sowohl die Objektmerkmale als auch die zugehörigen Kategorien benötigen, um die Klassifikatoren zu lernen, spricht man von überwachtem Lernen. Auf Basis dieser gelernten Klassifikatoren können neue, unbekannte Objekte oder Kunden mit ihren Merkmalen in die wahrscheinlichste Kategorie eingeordnet werden. (Kirchberg, S. 51)

Entscheidungsbäume sind eine effektive Methode zur Klassifikation, die nach dem Prinzip "Teile-und-Herrsche" arbeiten. Dabei werden Objekte durch eine Abfolge von Tests aufgeteilt, wobei die Klassifikationsabfrage für jede Teilmenge wiederholt wird. Diese Prozedur setzt sich fort, bis durch die Tests eine präzise Vorhersage der Klasse ermöglicht wird.   
Die Struktur und Funktionsweise von Entscheidungsbäumen lassen sich gut grafisch darstellen: Als gerichtete, zyklenfreie Graphen bestehen sie aus Knoten und Kanten. Die Knoten repräsentieren die durchzuführenden Tests, während die Kanten, in diesem Kontext auch gennant Zweige, die Abfolge der Tests beschreiben. Besonders sind die Blattknoten, die keine weiteren Kanten haben und die endgültige Klassifikation angeben.

Entscheidungsbäume beginnen typischerweise mit de Wurzelknoten, die den Anfangstest darstellt, und wachsen dann von oben nach unten. Diese Baumstruktur ist besonders anschaulich und eignet sich gut, um komplexe Entscheidungsprozesse zu visualisieren. (Kirchberg, S. 54f)

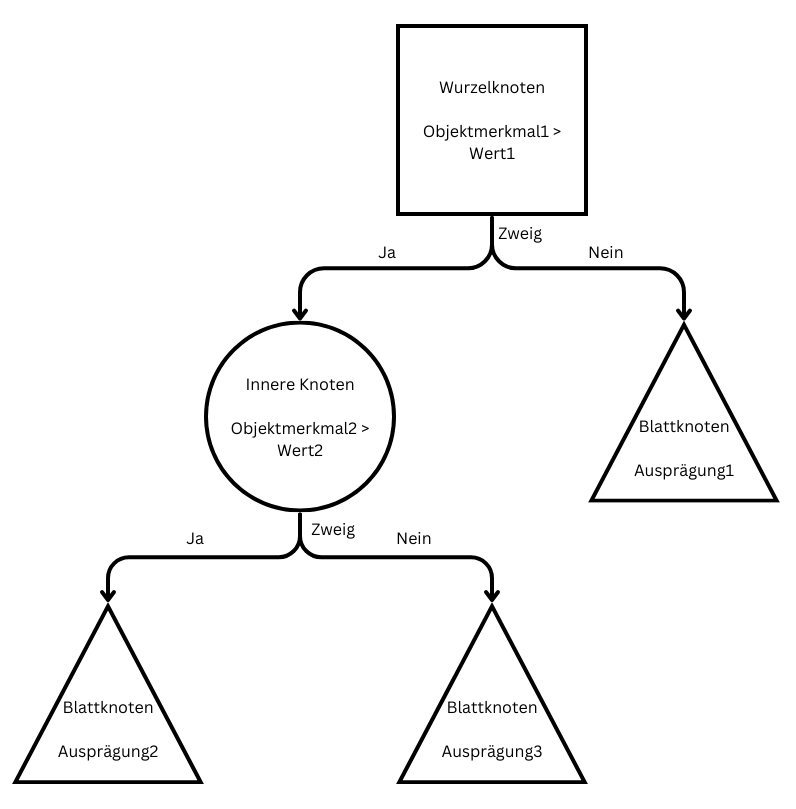


Abbildung 1: Exemplarischer Entscheidungsbaum

In Abbildung 1 wird ein exemplarischer Entscheidungsbaum dargestellt. Das Quadrat stellt die Wurzel bzw. den Startknoten dar. Es wird eine Klassifikationsabfrage für eine Teilmenge der Daten gestellt, ob das Objektmerkmal 1 bei jedem betrachteten Datensatz größer ist als Wert1. Die Kanten bzw. Zweige repräsentieren die möglichen Übergänge von einem Knoten zum nächsten, basierend auf den getroffenen Entscheidungen. Die Kanten werden oft durch Pfeile visualisiert, um die Richtung des Entscheidungsflusses zu verdeutlichen. Ist das Objektmerkmal 1 nicht größer als Wert1, gelangt man in das Rechteck, das hier den Blattknoten darstellt. In den Blattknoten angekommen, wird der Datensatz in den möglichen Ausprägungen klassifiziert. Ist jedoch das Objektmerkmal1 größer als Wert1, gelangt man in den Kreis, der hier einen inneren Knoten darstellt. Ein innerer Knoten in einem Entscheidungsbaum ist ein Knoten, der eine Entscheidungsregel oder Bedingung repräsentiert, anhand derer der Datenfluss weiter aufgeteilt wird. Diese Knoten stehen zwischen dem Wurzelknoten und den Blattknoten. Inneren Knoten folgen weitere innere Knoten oder Blattknoten, die durch Zweige verbunden sind, und sie sind entscheidend für die Strukturierung und Feinabstimmung der Entscheidungslogik innerhalb des Baums. In dieser Abbildung folgen diesem inneren Knoten zwei Blattknoten, die dann nach dem eben beschriebenen Schema den Datensatz in Ausprägung2 oder Ausprägung3 klassifizieren.

# Entscheidungsbaum-Klassifikation mit R

Im vorherigen Kapitel wurden die theoretischen Grundlagen von Klassifikationsaufgaben sowie die Einordnung der Entscheidungsbäume innerhalb dieser erläutert.

In diesem Kapitel die Implementierung des Entscheidungsbaumalgorithmus in R vorgestellt. Zur Vorgehensweise wird das Knowledge Discovery in Database als Entwicklungsschema genutzt. Zu Beginn erfolgt eine detaillierte Analyse der verwendeten Daten durch eine explorative Datenanalyse, um die Struktur und Merkmale der Daten besser zu verstehen. Anschließend werden die Daten falls nötig in relevanten Daten eingeteilt sowie bereinigt, vorverarbeitet, reduziert und verdichtet. Nachdem wird der Entscheidungsbaumalgorithmus schrittweise implementiert. Die Schritte umfassen unter anderem die Modellierung in R, die Optimierung der Modellparameter und die Evaluierung der Ergebnisse. Ziel ist es, die Leistungsfähigkeit des Entscheidungsbaums zu bewerten und zu beurteilen, wie gut dieses Modell für die spezifische Klassifikationsaufgabe geeignet ist. Dabei werden die verschiedenen Implementierungsschritte betrachtet und die verfügbaren Möglichkeiten sowie deren Effektivität evaluiert. Zum Schluss werden die Ergebnisse analysiert.

## Knowledge Discovery in Database

Der Begriff "Knowledge Discovery in Databases" (KDD) wurde erstmals 1989 geprägt und bildet die Grundlage für moderne datenanalytische Methoden (Fayyad, Piatetsky-Shapiro, & Smyth, 1996b, S. 28). KDD bezeichnet den komplexen Prozess der Identifikation von Mustern in Daten, die als gültig, neuartig, nützlich und verständlich gelten (Fayyad, Piatetsky-Shapiro, & Smyth, 1996a, S. 41). Diese Muster werden zu Wissen, wenn sie als interessant bewertet werden, basierend auf ihrer Gültigkeit, Neuartigkeit, Nützlichkeit und Einfachheit (Kurgan & Musilek, 2006, S. 3). Das Hauptziel von KDD ist es, Erkenntnisse aus Daten zu abstrahieren, um zukünftige Entwicklungen vorhersagen zu können (Fayyad, Piatetsky-Shapiro, & Smyth, 1996a, S. 37). Data Mining ist dabei nur ein Teil des KDD-Prozesses, der auch Datenaufbereitung und Ergebnisvisualisierung umfasst, um sinnvolle Resultate zu gewährleisten (Fayyad, Piatetsky-Shapiro, & Smyth, 1996a, S. 42). KDD ist kein konkretes Vorgehensmodell, sondern eine methodische Basis, die durch ein Framework mit neun Aufgaben strukturiert wird und als iterativer Prozess ähnlich dem PDCA-Zyklus verstanden werden kann. Dieses Vorgehen erlaubt flexible Sprünge zwischen den einzelnen Aufgaben.

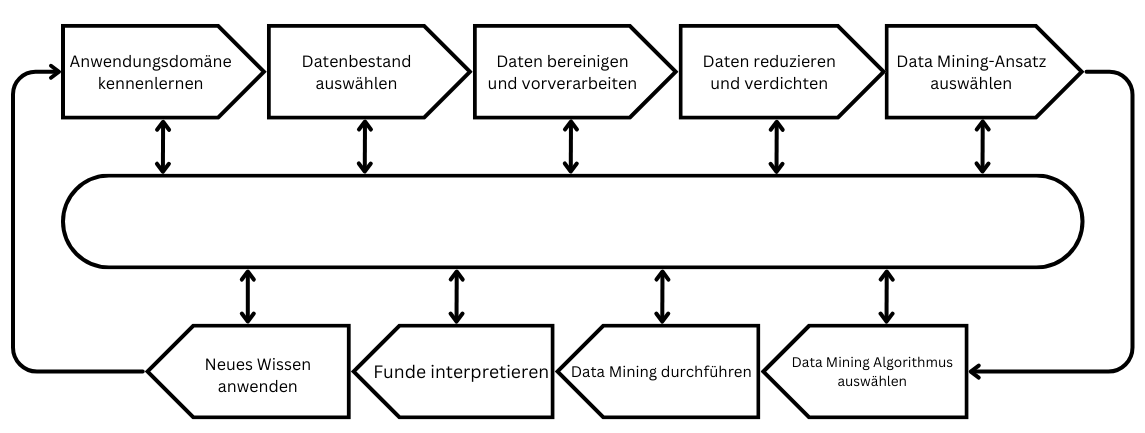


Abbildung 2: KDD im Überblick (in Anlehnung an Fayyad/ Piatetsky-Shapiro/Smyth 1996b, S.29)

In Abbildung 2 werden die neun Aufgaben von KDD dargestellt die im weiteren Verlauf des Kapitels genutzt und weiter erläutert werden.

## Datenexploration

Datenexploration ist ein wesentlicher Schritt im Data-Mining-Prozess, bei dem Daten visuell und statistisch untersucht werden, um Muster, Anomalien und Beziehungen zu erkennen. Dieser explorative Ansatz ermöglicht es, ein besseres Verständnis der Daten zu entwickeln und Hypothesen für die weitere Analyse zu generieren (Tukey, 1977). Eine gründliche Datenexploration kann dabei helfen, die Qualität der Daten zu beurteilen und notwendige Vorverarbeitungen zu identifizieren, die für die erfolgreiche Durchführung komplexer Analysemethoden erforderlich sind (Han, Pei, & Kamber, 2011).

Ziel ist es die mit der Datenexploration die erste Aufgabe des KDD zu behandeln und zwar die Anwendungsdomäne kennenzulernen und zu verstehen (Fayyad/ Piatetsky-Shapiro/Smyth 1996b, S.30).

Dafür wird zu beginn die verfügbaren Features bzw. die Kundenmerkmale angeschaut.

*# Ausgabe der Spaltennamen*

**colnames**(data)

1. 'Alter'
2. 'Einkommen'
3. 'Abschluss'
4. 'Geschlecht'
5. 'Wohngebiet'
6. 'Stellung'
7. 'Eigenheim'
8. 'Kinderanzahl'
9. 'Familienstand'
10. 'Bundesland'
11. 'MitgliedSportverein'
12. 'Krankheitstage'
13. 'Preiskategorie'

Es gibt insgesamt zwölf Kundenmerkmale und das zu klassifizierde Merkmal, die Preiskategorie. Also anhand des Alters, Einkommen usw. sollen Kunden vorab durch einen Klassifikator in einer Preiskategorie eingeteilt werden.

*# Anzahl der Zeilen der Daten*

**nrow**(data)

652

Dazu stehen in den Daten 652 Datensätze bzw. Kunden zur Verfügung. Die Daten haben eine Dimension von 652x13. Relevant ist die Größe der Daten bzw. die Dimension für den Trainingsprozess. Entscheidungsbäume sind anfällig für Overfitting. Overfitting in Entscheidungsbäumen tritt auf, wenn das Modell zu komplex wird und spezifische Details oder Rauschen in den Trainingsdaten erfasst. Dies führt dazu, dass das Modell auf neuen Daten schlecht generalisiert und zu ungenauen Vorhersagen neigt. Entscheidungsbäume sind besonders anfällig für Overfitting, wenn sie ohne Einschränkungen wachsen dürfen, da sie dann jeden möglichen Pfad in den Daten verfolgen können. Auf der anderen Seite gibt es auch Underfitting, wenn das Modell zu einfach ist und nicht in der Lage ist, die zugrunde liegenden Muster in den Daten zu erkennen. Dies kann passieren, wenn der Baum nicht tief genug ist oder wenn wichtige Merkmale nicht berücksichtigt werden. Um diese Probleme zu vermeiden, können Techniken wie Pruning angewendet werden, bei dem unwichtige Knoten entfernt werden, um die Komplexität des Baumes zu verringern und eine bessere Generalisierung auf neue Daten zu erreichen. (Xu, 2023, S.169)  
In den Artikel von Christanti et al. wird ein bestimmtes Prunning Verfahren, Bayes Risk Post-Pruning, vorgestellt um das Problem von Overfitting zu reduzieren. Dabei werden zwei verschiedene Datensätze mit verschiedenen Dimensionen genutzt um das Problem des Overfitting zu demonstieren und um zu zeigen das deren Pruning Verfahren funktioniert. (Christianti et al., 2020)

Für weitere Betrachtung werden die Kundenmerkmale in ihrem Messniveau oder in ihre Skalierung in verschiedenen Typen eingeteilt..

* Nominale Daten: Daten werden in Kategorien eingeordnet die keine natürliche Ordnung haben.
  + Dazu zählen diese Merkmale:
    - Stellung: Angestellter, Arbeiter, Arbeitslos, Beamter, Selbststaendig
    - Bundesland: Baden-Wuerttemberg, Bayern, Berlin, Hamburg, Hessen, Mecklemburg-Vorpommern, Mecklenburg-Vorpommern, Niedersachsen, Nordrhein-Westfalen, Rheinland-Pfalz, Saarland, Sachsen, Sachsen-Anhalt, Schleswig-Hollstein, Schleswig-Holstein, Thueringen
      * Hier fällt auf das sich Rechtschreibfehler in Mecklemburg-Vorpommern und schlesi sowie in Schleswig-Hollstein und Schleswig-Holstein sich einschliechen. Dieser Fehler wird später noch behoben.
    - Mitgliedverein: j, n
    - Familienstand: geschieden, ledig, verheiratet, verwitwet
    - Geschlecht: m,w
    - Eigenheim: j,n
* Ordinal Daten: sind ebenfalls kategorisch, aber im Gegensatz zu nominalen Daten gibt es eine klare, geordnete Beziehung zwischen den Kategorien.
  + Dazu zählen diese Merkmale:
    - Abschluss: keiner, Hauptschule, Realschule, Gymnasium, Studium, Promotion
    - Wohngebiet: Land, Dorf, Kleinstadt, Mittelstadt, Grossstadt
    - Preiskategorie: niedrig, mittel, hoch
* Intervall-Daten: Numerische Daten, bei denen die Abstände zwischen den Datenpunkten gleich sind, aber es gibt keinen natürlichen Nullpunkt.
  + Dazu zählen diese Merkmale:
    - Keine Merkmale in den Daten sind Intervall-Daten
* Verhältnis-Daten: Sind wie Intervall-Daten, aber mit einem natürlichen Nullpunkt.
  + Dazu zählen diese Merkmale:
    - Alter: Zwischen 21 und 49
    - Einkommen: Zwischen 12.545 und 191.469
    - Kinderanzahl: Zwischen 0 und 4
    - Krankheitstage: Zwischen 0 und 19

## Datenvorbereitung und -verarbeitung

Die zweite Aufgabe des KDD ist es den Datenbestand auszuwählen (Fayyad/ Piatetsky-Shapiro/Smyth 1996b, S.30). Die Daten werden hier in Rahmen dieser Arbeit vorgegeben.   
Der Zweck der Daten ist es herauszufinden, wie die Kundenmerkmale genutzt werden können, um Kunden in Preiskategorien einzuteilen. Dazu scheint es sinnvoll, Merkmale wie das Einkommen, Wohngebiet, Stellung, Kinderanzahl etc. für die Kategorisierung zur Verfügung zu haben. Ob Merkmale wie "Kunde ist im Sportverein" relevant sind, wird bei der Analyse der Ergebnisse des Entscheidungsbaums festgestellt, sowie alle bei allen anderen Kundenmerkmalen auch.

Als nächstes werden die Datenbereinigt und -vorverarbeitet (Fayyad/ Piatetsky-Shapiro/Smyth 1996b, S.30). Es wird in den Daten untersucht ob es Außreißer, Rauschen oder fehlende Daten gibt.

*# Ausgabe Ausprägungen von den Merkmalen*

for (col in **colnames**(data)) {

**print**(**paste**("Spalte:", col))

**print**(**table**(data[[col]]))

[1] "Spalte: Alter"

21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46

1 2 3 6 8 13 24 24 23 29 56 47 49 51 47 55 34 46 36 31 23 14 15 4 4 1

47 48 49

3 1 2

[1] "Spalte: Einkommen"

12545 13804 13827 14217 15195 15496 16376 16799 16888 16939 17129

2 1 1 1 2 1 1 1 1 1 1

17170 17463 17600 17928 18401 18476 18534 18615 18729 18761 19015

1 1 2 1 1 1 1 1 1 1 1

19094 19422 19665 20258 20460 20640 20747 20832 21275 21368 21497

1 2 1 1 1 1 1 1 2 1 1

21499 21659 21703 21807 22147 22369 22434 22904 22913 23080 23113

1 1 1 1 1 1 1 1 2 1 1

23127 23301 23524 23759 23939 24006 24140 24183 24264 24278 24415

1 1 1 1 1 1 1 1 2 1 1

24476 24548 24587 24604 24644 24653 24778 24804 24818 24960 24981

1 1 2 1 1 1 1 1 1 1 1

25106 25115 25123 25225 25276 25425 25477 25491 25599 25675 25712

2 1 1 1 1 1 2 1 1 2 1

25785 25892 25930 25991 25994 26096 26106 26264 26370 26528 26548

1 1 1 1 1 1 1 1 1 1 1

26553 26557 26585 26599 26681 26728 26766 26826 26827 26875 26934

2 1 1 1 2 1 1 1 1 1 1

26937 26953 26957 27081 27125 27159 27166 27250 27256 27264 27273

1 1 1 1 1 1 1 1 2 2 1

27354 27363 27615 27625 27632 27688 27783 27937 27999 28012 28129

1 1 1 1 1 1 3 1 1 1 1

28169 28306 28343 28391 28446 28477 28480 28486 28489 28553 28581

1 1 1 1 1 1 1 1 1 1 1

28636 28704 28761 28785 28820 28831 28970 28980 29000 29130 29151

1 1 1 1 1 1 1 1 1 1 1

29228 29365 29399 29452 29525 29541 29553 29646 29650 29667 29682

1 2 1 1 1 1 1 1 1 1 1

29692 29699 29702 29724 29739 29775 29847 29855 29862 29965 29966

1 1 1 2 1 1 1 2 1 1 1

29991 30000 30004 30072 30101 30157 30294 30438 30551 30635 30659

1 1 1 1 1 1 1 1 1 1 1

30674 30679 30685 30702 30717 30720 30761 30798 30883 30908 31013

1 1 1 1 1 1 1 1 1 1 1

31085 31108 31155 31161 31172 31205 31211 31298 31314 31385 31391

1 1 2 1 1 1 1 1 1 1 1

31460 31543 31573 31607 31608 31640 31652 31662 31675 31759 31819

1 1 1 1 2 1 1 1 1 2 1

31825 31831 31834 31874 31880 31900 31901 31909 31947 31959 31961

2 1 1 1 1 2 1 1 1 1 1

31984 32006 32103 32112 32139 32141 32162 32214 32325 32382 32405

1 1 1 1 1 1 1 1 1 1 1

32429 32446 32563 32719 32812 32856 32885 32896 32910 32919 32949

3 1 1 1 1 1 1 1 1 1 2

32969 33056 33099 33105 33162 33200 33221 33256 33284 33291 33316

1 1 1 1 1 1 1 1 1 1 1

33322 33349 33375 33380 33390 33410 33444 33462 33477 33493 33496

1 1 1 1 1 1 1 1 1 1 1

33589 33650 33761 33812 33850 33891 33920 33960 33963 34027 34028

1 2 1 1 1 2 1 2 1 1 1

34112 34141 34161 34163 34187 34193 34204 34227 34260 34369 34370

1 1 1 1 1 1 1 1 1 1 1

34396 34422 34456 34481 34513 34541 34581 34625 34695 34703 34707

1 1 1 1 1 1 1 2 1 1 1

34710 34742 34787 34846 34861 34871 34927 34967 35639 35730 36354

1 1 1 1 2 1 1 1 2 1 1

36404 36797 36848 37593 38036 38318 38348 38459 38650 38842 38993

1 1 1 1 1 1 1 1 1 1 2

39365 39406 39486 39557 39587 39597 39617 39899 40352 40393 40826

1 1 1 2 1 1 1 1 1 1 1

41259 41420 41480 41672 41792 42226 42759 43303 43414 43535 43635

1 1 1 1 1 1 1 2 1 1 1

43686 43917 43968 44069 44169 44673 44753 45045 45891 46284 46385

1 1 1 1 2 1 1 1 1 1 1

46435 46536 46566 47170 47211 47382 47724 48470 48530 48903 49527

1 2 1 1 1 1 1 2 1 1 1

49668 49718 49789 49819 49920 50645 50987 51008 51078 51179 51290

1 1 1 1 1 1 1 1 1 2 2

51340 51390 51612 52065 52266 52528 52558 52770 53556 53878 54422

2 1 1 1 1 1 1 1 1 2 1

54694 54714 54774 55207 55519 56406 56416 56718 56869 56970 57040

1 1 1 1 1 1 1 1 1 1 1

57282 57302 57453 57584 57594 57654 58098 58229 58299 58440 58500

1 1 1 1 2 1 1 2 1 1 2

58692 58752 59226 59447 59810 59971 60092 61069 61733 62015 62156

1 1 1 1 1 1 1 1 1 1 2

62186 62670 62680 62730 62811 62952 63053 63294 63677 63687 63959

1 1 1 1 2 1 1 1 1 1 1

64231 64311 64795 66114 66507 66900 67000 67564 68048 68249 68803

1 1 1 1 1 1 1 1 1 1 1

68904 69125 69377 69478 70112 70485 71120 71150 71381 71573 72117

1 1 1 1 1 1 1 1 1 1 1

73174 73285 73355 73678 74030 74091 74362 74433 74534 74634 75027

1 1 1 1 1 1 1 1 1 1 1

75460 75531 75601 76034 76075 76679 77313 77746 78290 79126 79670

1 1 1 1 1 1 1 1 1 1 1

79710 79922 80073 80355 80868 81080 81886 81936 82379 82570 82762

1 1 1 1 1 1 1 2 1 1 2

83134 83265 83598 83658 84202 84313 84393 85864 85904 86075 86095

1 2 1 1 1 1 1 1 1 1 1

86357 87012 87022 87606 87918 88402 88996 89389 89610 89640 89902

1 1 2 1 1 1 1 1 1 1 1

90436 90637 90849 91896 93306 93568 93931 94122 94162 94474 94686

2 1 1 1 1 1 1 1 2 1 1

94706 95633 98181 99560 100044 100467 100809 101121 101504 101564 101705

1 1 1 1 1 1 1 1 1 1 1

101846 102128 102803 102874 103629 104707 105845 106006 106721 107647 108050

1 1 1 1 1 1 1 2 1 1 1

110981 111112 111998 112089 113166 114163 114385 114566 115815 118544 118736

1 1 2 1 1 1 1 1 1 1 2

120327 121495 122170 122653 123288 123298 123771 124305 124607 125302 125645

1 1 1 1 1 1 1 1 2 1 1

127014 127216 127860 128223 129743 130569 133651 133661 134578 135645 135826

1 1 1 1 1 1 1 1 1 1 1

137085 139432 140278 140640 143178 143652 145988 147720 149966 150037 151346

1 1 1 2 1 2 1 2 1 1 1

152524 153461 156653 162626 182305 191469

2 1 1 1 1 1

[1] "Spalte: Abschluss"

Gymnasium Hauptschule Keiner Promotion Realschule Studium

115 124 141 16 122 134

[1] "Spalte: Geschlecht"

m w

328 324

[1] "Spalte: Wohngebiet"

Dorf Grossstadt Kleinstadt Land Mittelstadt

44 327 82 26 173

[1] "Spalte: Stellung"

Angestellter Arbeiter Arbeitslos Beamter Selbststaendig

259 211 56 60 66

[1] "Spalte: Eigenheim"

j n

56 596

[1] "Spalte: Kinderanzahl"

0 1 2 3 4

233 177 120 73 49

[1] "Spalte: Familienstand"

geschieden ledig verheiratet verwitwet

125 324 132 71

[1] "Spalte: Bundesland"

Baden-Wuerttemberg Bayern Berlin

51 57 30

Hamburg Hessen Mecklemburg-Vorpommern

40 41 41

Mecklenburg-Vorpommern Niedersachsen Nordrhein-Westfalen

2 44 57

Rheinland-Pfalz Saarland Sachsen

48 46 42

Sachsen-Anhalt Schleswig-Hollstein Schleswig-Holstein

45 5 41

Thueringen

62

[1] "Spalte: MitgliedSportverein"

j n

78 574

[1] "Spalte: Krankheitstage"

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19

2 14 53 107 145 46 51 45 41 37 25 23 20 13 13 5 4 6 1 1

[1] "Spalte: Preiskategorie"

hoch mittel niedrig

84 380 188

Rauschen bezeichnet zufällige oder unerwartete Variationen in den Daten, die das Signal verzerren und die Analyse erschweren können. Diese zufälligen Störungen resultieren oft aus Messfehlern oder anderen unkontrollierbaren Variablen. (Bibel, 2000, S.17). Es wird angenommen, dass in den Daten kein rauschen vorliegt, da es eindeutige Kundenmerkmale sind.   
In der Literatur gibt es viele Methoden um in den Daten Rauschen zu erkennen und zu entfernen. Wie die Technik Wavelet-Denoising, die Rauschen durch mehrstufige Analyse der Daten entfernt. (To et al., 2009).

Ausreißer: Ein Ausreißer ist ein Datenpunkt, der sich signifikant von den anderen Datenpunkten unterscheidet. (Bibel, 2000, S.17) Ausreißer erkennt man gut, indem unteranderem die Interquartilabstand Methode verwendet wird. Diese identifiziert Ausreißer indem der IQR berechnet wird, dieser um das 1.5 vergrößert wird und liegt ein Wert in diesen Daten außerhalb dieses Bereichs von Q1 oder Q3 aus wird dieser entfernt. (Vinutha et al., 2018) Eine andere Methode wäre die Z-Score Methode. Der Z-Score misst, wie viele Standardabweichungen ein Datenpunkt vom Mittelwert entfernt ist. Werte mit einem Z-Score größer als 3 oder kleiner als -3 gelten oft als Ausreißer. Diese Methode ist jedoch empfindlich gegenüber extremen Werten und setzt voraus, dass die Daten normal verteilt sind. (Zhang, 2011)  
Auch hier wird angenommen, dass keine Ausreißer vorliegen ohne einen spezifischen Test durchzuführen. Der Fokus dieser Arbeit liegt auf die Implementierung eines Entscheidungsbaum in R.

Ob fehlende Daten vorhanden sind wird folgender Code dazu genutzt.

*# Überprüfen ob fehlende Werte vorhanden sind*

**sum**(**is.na**(data))

0

Das fehlende Daten vorhanden sind wäre auch bereits in der Datenexploration aufgefallen. Aber es ist aufgefallen das Rechtschreibfehler in den Ausprägungen vorliegen.

*# Ändere den Namen der Ausprägung Mecklemburg-Vorpommern in Mecklenburg-Vorpommern sowie Schleswig-Hollstein in Schleswig-Holstein*

data$Bundesland[data$Bundesland == "Mecklemburg-Vorpommern"] <- "Mecklenburg-Vorpommern"

data$Bundesland[data$Bundesland == "Schleswig-Hollstein"] <- "Schleswig-Holstein"

Im Anschluss würde geguckt werden ob die Daten reduziert und verdichtet werden können (Fayyad/ Piatetsky-Shapiro/Smyth 1996b, S.30). Auch hier wird darauf verzichtet aber es wird noch erwähnt wie man das machen könnte.  
Eine typische Methode wäre die Principal Component Analysis (PCA). Das Ziel der PCA ist es, die aussagekräftigste Grundlage für die Neuformulierung eines gegebenen Datensatzes zu identifizieren. Es wird erwartet, dass diese neue Basis versteckte Strukturen im Datensatz aufdeckt und das Rauschen herausfiltert. Es gibt so viele Anwendungen wie Dimensionalitätsreduzierung, Datenkomprimierung, Merkmalsextraktion und Datenvisualisierung. (Kurita, 2019) Eine weitere methode wäre die Factor Analysis (FA). Ähnlich wie PCA, versucht FA die zugrunde liegenden Faktoren oder Dimensionen in den Daten zu identifizieren, die die Varianz erklären. (Leckliter et al., 1986)

## Trainings- und Testdatensplit

In der Entwicklung von Vorhersagemodellen werden die Daten, mit denen das Modell trainiert oder angepasst wird, als Trainingsdaten bezeichnet. Diese Daten dienen dazu, die Parameter des Modells zu optimieren und das Modell auf die vorhandenen Muster und Zusammenhänge zu sensibilisieren. Ein wichtiger Aspekt in der praktischen Anwendung ist jedoch, wie gut das Modell bei der Vorhersage von Klassen für neue, bisher nicht gesehene Daten funktioniert. Um diese Leistungsfähigkeit zu überprüfen, verwendet man sogenannte Testdaten – ein separater Datensatz, der zur Modellanpassung nicht verwendet wurde und daher als „neu“ gilt. Alternativ kann auch Kreuzvalidierung eingesetzt werden, bei der die vorhandenen Daten mehrfach in Trainings- und Testdatensätze aufgeteilt werden. In dieser Betrachtung konzentrieren wir uns jedoch auf die direkte Klassifizierung der Beobachtungen im Testdatensatz mit der entwickelten Klassifikationsmethode. Eine gute Methode sollte dabei in der Lage sein, nahezu alle Beobachtungen korrekt zu klassifizieren. (Runkler, 2015, S.44)

*# Funktion, die Daten in 80% Trainingsdaten und 20% Testdaten aufteilt*

**split\_data** <- function(data, train\_ratio = 0.8) {

**set.seed**(13)  *# Für Reproduzierbarkeit*

    total\_rows <- **nrow**(data)

    train\_indices <- **sample**(1:total\_rows, size = total\_rows \* train\_ratio)

    train\_data <- data[train\_indices, ]

    test\_data <- data[-train\_indices, ]

    return(list(train\_data = train\_data, test\_data = test\_data))

}

*# Beispielverwendung*

split <- split\_data(data)

train\_data <- split$train\_data

test\_data <- split$test\_data

*# Anzahl der Zeilen der Trainings- und Testdaten drucken*

**print**(**paste**("Anzahl der Zeilen der Trainingsdaten:", **nrow**(train\_data)))

**print**(**paste**("Anzahl der Zeilen der Testdaten:", **nrow**(test\_data)))

[1] "Anzahl der Zeilen der Trainingsdaten: 521"

[1] "Anzahl der Zeilen der Testdaten: 131"

## Entscheidungsbaum Implementierung

Die nächste Aufgabe laut dem KDD wäre es den richtigen Data-Mining Ansatz auszuwählen (Fayyad/ Piatetsky-Shapiro/Smyth 1996b, S.31). Im Kapitel 2. wurde bereits geklärt welcher Ansatz hier gewählt werden wird: die Klassifikation. Dadurch dass geklärt ist das die Kunden in Preiskategorien eingeteilt werden sollen, ist das Label bzw. die Klasse bekannt in dem die Kunden klassifiziert werden soll. Im gegensatz zum Clustering, indem wenn die Kunden in Gruppen eingeteilt werden sollen, die innnerhalb der Gruppen homogen sind und zu anderen Gruppen heterogen sind ohne vorab die Kunden in bestimmten Gruppen einzuteilen, wie hier in Preiskategorien. (Schendera, C. F. G., 2010)

Anschließend daran soll der geeignete Data Mining Algorithmus gewählt werden. In Rahmen dieser Arbei soll der Entscheidungsbaum Algorithmus genutzt werden.   
Es gibt natürlich neben diesen weiteren Algorithmen. Der Native Bayes Klassifkator zum Beispieö, der ein probabilistischer Klassifikator ist, der auf dem Bayesschen Theorem basiert und davon ausgeht, dass die Merkmale unabhängig voneinander sind. (Ren et al., 2009) Aber in unseren Daten könnte vermutet werden, dass es einen zusammenhang zwischen Stellung und Einkommen gibt. Dafür könnte ein geeigneter statitscher Abhängigkeitstest verwendet werden.   
Einen weiterer wäre der Support Vector Maschine. Ein leistungsfähiger Klassifikator, der versucht, die optimale Trennlinie (oder -ebene) zwischen verschiedenen Klassen zu finden. (Noble, 2006).

Der Vorteil von Entscheidungsbäume ist, dass dieser leicht zu verstehen und zu interpretieren ist, sowie keine Datenvorverarbeitung benötigt wie manch andere. Es benötigt keine Normalisierung oder Standartisierung der Daten.

Welcher spezifischer Entscheidungsbaum-Algorithmus zur Implementierung genutzt wird, wird anschließend erläutert.   
In Rahmen dieser Arbeit wird sich auf drei bekannte spezifische Algorithmen beschränkt die zur Auswahl stehen. Ein zentrales Unterscheidungsmerkmal ist, welche Berechnungsmethodik zur Bestimmung der Abhängigkeit von Merkmalen und Zielklasse für die Auswahl des nächsten Tests verwendet wird und ob Pruning eingesetzt wird. (Kirchberg, S. 61)

C4.5, CHAID und CART sind drei spezifische Algorithmen zur Erstellung von Entscheidungsbäumen, die unterschiedliche Methoden zur Auswahl von Attributen und zur Baumkonstruktion verwenden. C4.5 verwendet die Entropie und den daraus abgeleiteten Informationsgewinn, um Entscheidungsbäume zu erstellen, die numerische, fehlende und fehlerhafte Werte verarbeiten können. Chi-Squared Automatic Interaction Detector (CHAID) hingegen nutzt den Chi-Quadrat-Test zur Auswahl von Attributen und ist besonders effizient bei der Handhabung mehrwertiger Attribute, während CART auf binäre Bäume fokussiert ist und den Gini-Index zur Attributauswahl verwendet, was zu besonders tiefen Bäumen führt, die nachträglich beschnitten werden müssen, um Überanpassung zu vermeiden. (Petersohn, 2005, S. 106f) (Behnisch, 2008 S. 93f)

Beim Vergleich dieser Algorithmen ist C4.5 ideal, wenn es um die Verarbeitung komplexer und unvollständiger Daten geht. CHAID eignet sich besonders gut für die Analyse multivariater Daten, bei denen mehrere Attribute gleichzeitig untersucht werden und die Datenmenge nicht allzu groß ist, da dieser Algorithmus rechenintensiv ist. CART ist die beste Wahl, wenn eine einfache, binäre Baumstruktur bevorzugt wird oder wenn Regression erforderlich ist. Die Auswahl des passenden Algorithmus hängt daher stark von den spezifischen Datenanforderungen und dem gewünschten Ergebnis ab. (Petersohn, 2005, S. 106f) (Behnisch, 2008 S. 93f)

Auf Grundlage dieser Erkenntnisse wird der C4.5-Algorithmus gewählt. C4.5 ist besonders geeignet, da er sowohl mit numerischen als auch mit kategorialen Daten umgehen kann, ohne dass eine vorherige Diskretisierung notwendig ist. In unseren Datensatz gibt es numerische Daten wie das Einkommen aber auch kategorische Daten wie die Stellung. Der Algorithmus verwendet Entropie und Informationsgewinn, um das beste Attribut für den nächsten Split zu bestimmen, und erstellt den Entscheidungsbaum in einem rekursiven Top-down-Verfahren. Nachdem der Baum vollständig erstellt wurde, wird ein Pruning durchgeführt, um übermäßiges Wachstum zu verhindern und die Generalisierungsfähigkeit des Modells zu verbessern. (Petersohn, 2005, S. 106f)  
Pruning bezieht sich auf das Zurückschneiden eines Entscheidungsbaums, um dessen Komplexität zu reduzieren und das Risiko von Overfitting zu minimieren. Dabei gibt es zwei Hauptarten: Pre-Pruning, bei dem das Wachstum des Baums während des Aufbaus durch vordefinierte Abbruchkriterien begrenzt wird, und Post-Pruning, bei dem nach dem vollständigen Aufbau des Baums überflüssige oder schwache Verzweigungen entfernt werden, um die Generalisierungsfähigkeit des Modells auf neue Daten zu verbessern. (Kirchberg, S. 59f) In welcher Form Pruning notwendig und wenn durchgeführt wird, wird während der implementierung bestimmt.

Es gibt vier relevante KPIs für den Entscheidungsbaum (Dai & Ji, 2014, S. 50ff):

Entropie:

* Dabei sei *C* die Anzahl der Klassen und *p*(*s,j)* der Anteil der Instanzen in den Attributen S , die der *j-ten* Klasse zugeordnet sind. Daher berechnet sich die Entropie des Attributs *S* wie folgt:

*Entropie(S) =*

* Die Entropie misst die Ungewissheit oder Unordnung in einem System, wobei eine höhere Entropie darauf hinweist, dass die Daten gleichmäßig verteilt sind und es schwieriger ist, eine Vorhersage zu treffen.

*# Berechnung der Entropie*

**entropy** <- function(x) {

  freq <- **table**(x) / **length**(x)

  freq <- freq[freq > 0]  *# Entferne Null-Frequenzen*

  -**sum**(freq \* **log2**(freq))

  }

Gain:

* Dementsprechend ist der Informationsgewinn durch einen Trainingsdatensatz T wie folgt definiert mit als die verschiedenen Ausprägungen von :
* Der Informationsgewinn (Gain) misst die Verringerung der Entropie, die durch das Teilen eines Datensatzes nach einem bestimmten Attribut erzielt wird. Er zeigt an, wie gut ein Attribut die Daten in Bezug auf ihre Zielvariable trennt, wobei ein höherer Gain auf eine größere Verbesserung der Vorhersagekraft hindeutet.

*# Berechnung des Informationsgewinns (Gain)*

**gain** <- function(dataX, dataY, attribute, threshold = NULL) {

  if (**is.null**(threshold)) {

    values <- **unique**(dataX[[attribute]])

    weighted\_entropy <- 0

    for (val in values) {

    subset\_dataY <- dataY[dataX[[attribute]] == val]

    weighted\_entropy <- weighted\_entropy + (**length**(subset\_dataY) / **length**(dataY)) \* entropy(subset\_dataY)

    }

  } else {

    subset\_dataY\_leq <- dataY[dataX[[attribute]] <= threshold]

    subset\_dataY\_gt <- dataY[dataX[[attribute]] > threshold]

    weighted\_entropy <- (**length**(subset\_dataY\_leq) / **length**(dataY)) \* entropy(subset\_dataY\_leq) +

              (**length**(subset\_dataY\_gt) / **length**(dataY)) \* entropy(subset\_dataY\_gt)

  }

  entropy(dataY) - weighted\_entropy

  }

Split-Information:

* Es wird wie folgt berechnet:
* Die Split-Information misst die Entropie des Splits selbst, also die Verteilung der Datenpunkte auf die verschiedenen Zweige nach einem Split. Sie quantifiziert, wie gleichmäßig der Split die Daten aufteilt, unabhängig von der Zielvariable. Eine höhere Split-Information bedeutet, dass der Split die Datenpunkte auf viele verschiedene Teile aufteilt, was eine geringere Präferenz für stark verzweigte Attribute in Entscheidungsbäumen begünstigt.

*# Berechnung der Split-Information*

**split\_info** <- function(dataX, dataY, attribute, threshold = NULL) {

  if (**is.null**(threshold)) {

    freq <- **table**(dataX[[attribute]]) / **nrow**(dataX)

  } else {

    freq <- **c**(**length**(dataX[dataX[[attribute]] <= threshold, ]) / **nrow**(dataX),

**length**(dataX[dataX[[attribute]] > threshold, ]) / **nrow**(dataX))

  }

  freq <- freq[freq > 0]  *# Entferne Null-Frequenzen*

  -**su**

Gain Raio:

* Es wird wie folgt berechnet:
* Der Gain Ratio ist ein Maß, das den Informationsgewinn (Gain) eines Splits durch die Split-Information (SplitInfo) normalisiert, um Verzerrungen zugunsten von Attributen mit vielen möglichen Werten zu vermeiden; er bevorzugt Splits, die sowohl einen hohen Informationsgewinn bieten als auch die Daten gleichmäßig aufteilen.

**gain\_ratio** <- function(dataX, dataY, attribute, threshold = NULL) {

  g <- gain(dataX, dataY, attribute, threshold)

  si <- split\_info(dataX, dataY, attribute, threshold)

  if (si == 0) {

    return(0)

  }

  gr <- g / si

  return(gr)

  }

Der C4.5-Algorithmus berechnet für jedes Attribut den Gain Ratio und wählt das Attribut mit dem höchsten Wert für den nächsten Split aus. Anschließend überprüft er, ob ein Abbruchkriterium (z. B. maximale Tiefe erreicht oder geringe Verbesserung) erfüllt ist. Wenn kein Abbruchkriterium erfüllt ist, teilt er die Daten basierend auf dem gewählten Attribut und wiederholt den Prozess rekursiv für die entstandenen Teilmengen, bis alle Bedingungen erfüllt sind oder die Daten vollständig klassifiziert sind. (Dai & Ji, 2014 S.51)

Der Entscheidungsbaum kann durch Pre-Pruning während des Aufbaus gestoppt werden, um Überanpassung zu vermeiden. Dies geschieht in folgenden Situationen: Erstens, wenn alle Datensätze in einem Knoten dieselbe Klassenzugehörigkeit haben. Zweitens, wenn alle Datensätze in einem Knoten in ihren Merkmalsausprägungen identisch sind, aber nicht unbedingt derselben Klasse zugeordnet sind. Drittens, wenn die Menge der Datensätze in einem Knoten eine bestimmte Mindestanzahl unterschreitet. Schließlich, wenn die Verbesserung durch eine weitere Verzweigung, gemessen anhand von Kriterien wie der Entropie, zu gering ist. (Kirchberg, S. 59f)

*# Funktion, um die besten Splits zu finden und den Entscheidungsbaum zu erstellen*

**find\_best\_splits** <- function(dataX, dataY, depth = 0, max\_depth, min\_samples, min\_gain, split\_history = "") {

**cat**(**sprintf**("Tiefe: %d, Anzahl der Stichproben hier1: %d\n", depth, **nrow**(dataX)))

*# Abbruchkriterien*

    if (depth >= max\_depth) {

**cat**("Abbruchbedingung erfüllt: Maximale Tiefe erreicht\n")

      decision <- **names**(**which.max**(**table**(dataY)))

      probability <- **max**(**table**(dataY)) / **length**(dataY)

      return(list(leaf = TRUE, decision = decision, probability = probability))

    }

    if (**nrow**(dataX) < min\_samples) {

**cat**("Abbruchbedingung erfüllt: Teilmenge unter min\_samples\n")

      decision <- **names**(**which.max**(**table**(dataY)))

      probability <- **max**(**table**(dataY)) / **length**(dataY)

      return(list(leaf = TRUE, decision = decision, probability = probability))

    }

    if (**length**(**unique**(dataY)) == 1) {

**cat**("Abbruchbedingung erfüllt: Alle Datensätze haben dieselbe Klassenzugehörigkeit\n")

      decision <- **unique**(dataY)

      return(list(leaf = TRUE, decision = decision, probability = 1.0))

    }

    if (**all**(**sapply**(dataX, function(col) **length**(**unique**(col))) == 1)) {

**cat**("Abbruchbedingung erfüllt: Alle Datensätze sind in ihren Merkmalsausprägungen identisch\n")

      decision <- **names**(**which.max**(**table**(dataY)))

      probability <- **max**(**table**(dataY)) / **length**(dataY)

      return(list(leaf = TRUE, decision = decision, probability = probability))

    }

    attributes <- **names**(dataX)

    gain\_ratios\_list <- **lapply**(attributes, function(attr) {

**cat**(**strrep**("#", 20), "\n")

      if (**is.numeric**(dataX[[attr]])) {

**cat**(**sprintf**("Finde besten Split für %s\n", attr))

        best\_split(dataX, dataY, attr, min\_samples)

      } else {

**cat**(**sprintf**("Finde besten Split für kategoriales Attribut %s\n", attr))

        best\_split\_categorical(dataX, dataY, attr)

      }

    })

    gain\_ratios\_list <- **Filter**(**Negate**(is.null), gain\_ratios\_list)  *# Entferne NULL-Einträge*

    if (**length**(gain\_ratios\_list) == 0) {

**cat**("Keine gültigen Splits gefunden\n")

      decision <- **names**(**which.max**(**table**(dataY)))

      probability <- **max**(**table**(dataY)) / **length**(dataY)

      return(list(leaf = TRUE, decision = decision, probability = probability))

    }

    gain\_ratios <- **sapply**(gain\_ratios\_list, function(x) x$gain\_ratio)

    if (**max**(gain\_ratios) < min\_gain) {

**cat**("Abbruchbedingung erfüllt: Verbesserung durch weitere Verzweigung zu gering\n")

      decision <- **names**(**which.max**(**table**(dataY)))

      probability <- **max**(**table**(dataY)) / **length**(dataY)

      return(list(leaf = TRUE, decision = decision, probability = probability))

    }

    best\_attribute <- attributes[**which.max**(gain\_ratios)]

    best\_attribute\_info <- gain\_ratios\_list[[**which.max**(gain\_ratios)]]

**cat**(**strrep**("#", 20), "\n")

    if (**is.numeric**(dataX[[best\_attribute]])) {

**cat**(**sprintf**("Bestes Attribut für Split: %s mit Schwellenwert: %s\n", best\_attribute, best\_attribute\_info$threshold))

    } else {

**cat**(**sprintf**("Bestes Attribut für Split: %s mit Ausprägung: %s\n", best\_attribute, best\_attribute\_info$value))

    }

**cat**(**strrep**("#", 30), "\n")

    if (**is.numeric**(dataX[[best\_attribute]])) {

      subset\_leq <- dataX[dataX[[best\_attribute]] <= best\_attribute\_info$threshold, ]

      subset\_gt <- dataX[dataX[[best\_attribute]] > best\_attribute\_info$threshold, ]

      subset\_dataY\_leq <- dataY[dataX[[best\_attribute]] <= best\_attribute\_info$threshold]

      subset\_dataY\_gt <- dataY[dataX[[best\_attribute]] > best\_attribute\_info$threshold]

      left\_child <- find\_best\_splits(subset\_leq, subset\_dataY\_leq, depth + 1, max\_depth, min\_samples, min\_gain,

**paste0**(split\_history, best\_attribute, "<=", best\_attribute\_info$threshold, " -> "))

      right\_child <- find\_best\_splits(subset\_gt, subset\_dataY\_gt, depth + 1, max\_depth, min\_samples, min\_gain,

**paste0**(split\_history, best\_attribute, ">", best\_attribute\_info$threshold, " -> "))

      return(list(attribute = best\_attribute, threshold = best\_attribute\_info$threshold,

                  left = left\_child, right = right\_child))

    } else {

      children <- list()

      for (val in **unique**(dataX[[best\_attribute]])) {

        subset\_val <- dataX[dataX[[best\_attribute]] == val, ]

        subset\_dataY\_val <- dataY[dataX[[best\_attribute]] == val]

**cat**(**sprintf**("Tiefe: %d, Anzahl der Stichproben hier2: %d %s:%s\n", depth + 1, **nrow**(subset\_val), split\_history, **paste0**(best\_attribute, ":", val)))

        children[[val]] <- find\_best\_splits(subset\_val, subset\_dataY\_val, depth + 1, max\_depth, min\_samples, min\_gain,

**paste0**(split\_history, best\_attribute, ":", val, " -> "))

      }

      return(list(attribute = best\_attribute, values = children))

    }

  }

In der Implementierung wurde das Reduced-Error-Pruning genutzt, um den Entscheidungsbaum nachträglich zu vereinfachen und Überanpassungen zu vermeiden. Dabei wurde für jeden inneren Knoten des Baums überprüft, ob das Ersetzen dieses Knotens durch ein Blattknoten die Fehlklassifikationsrate auf den Validierungsdaten nicht erhöht. Die Entscheidung basiert auf einem Bottom-up-Ansatz: Wenn die Reduzierung des Baums die Fehlklassifikationsrate nicht verschlechtert, wird der Teilbaum des Knotens abgeschnitten. Dieser Prozess wird fortgesetzt, bis jede weitere Kürzung die Klassifikationsgüte verschlechtern würde. (Rokach/Maimon 2014, S.70f.)

*# Funktion zur Berechnung der Fehlklassifikationsrate*

**calculate\_error\_rate** <- function(tree, validation\_data, validation\_labels) {

  predictions <- **sapply**(1:**nrow**(validation\_data), function(i) {

    row\_as\_list <- **as.list**(validation\_data[i, , drop = FALSE])

    classify(tree, row\_as\_list)

  })

  error\_rate <- **sum**(predictions != validation\_labels) / **length**(validation\_labels)

  return(error\_rate)

}

*# Funktion zum Prunen des Entscheidungsbaums*

**prune\_tree** <- function(tree, validation\_data, validation\_labels) {

*# Wenn der Knoten ein Blatt ist, gib den Knoten zurück*

  if (!**is.null**(tree$leaf) && tree$leaf) {

    return(tree)

  }

*# Rekursiv die linken und rechten Teilbäume prunen*

  if (!**is.null**(tree$left)) {

    tree$left <- prune\_tree(tree$left, validation\_data, validation\_labels)

  }

  if (!**is.null**(tree$right)) {

    tree$right <- prune\_tree(tree$right, validation\_data, validation\_labels)

  }

*# Berechne die Fehlklassifikationsrate für den aktuellen Baum*

  current\_error\_rate <- calculate\_error\_rate(tree, validation\_data, validation\_labels)

*# Ersetze den Knoten durch ein Blatt und berechne die Fehlklassifikationsrate*

  majority\_class <- ifelse(**sum**(validation\_labels == "j") >= **sum**(validation\_labels == "n"), "j", "n")

  pruned\_tree <- list(leaf = TRUE, decision = majority\_class)

  pruned\_error\_rate <- calculate\_error\_rate(pruned\_tree, validation\_data, validation\_labels)

*# Wenn die Fehlklassifikationsrate nicht steigt, schneide den Teilbaum ab*

  if (pruned\_error\_rate <= current\_error\_rate) {

    return(pruned\_tree)

  } else {

    return(tree)

  }

}

## Anwendung des Entscheidungsbaums

Nach der Implementation wird der Entscheidungsbaum erstellt und die Vorhersagegenauigkeit wird getestet um den nächsten Schritt des KDD, Data Mining durchführen, zu erfüllen. (Fayyad/ Piatetsky-Shapiro/Smyth 1996b, S.30).

Mit diesem Befehl wird der Entscheidungsbaum erstellt:

decision\_tree <- create\_decision\_tree(features\_train, target\_train, max\_depth = 4, min\_samples = 30, min\_gain = 0.1)

Es werden die Trainingsdaten benutzt die Parameter für den Pre-Pruning sind das die maximale Tiefe vier, jeder Zwei mindestens 30 Datensätze enthält und eine Abzweigung nur erstellt wird wenn der Gain Radio über 0.1 liegt.

Nun wird der Entscheidungsbaum genutzt um die Testdaten zu klassifizieren und um zu gucken wie gut die Vorhersagegenauigkeit ist.

*# Berechne die Genauigkeit des Entscheidungsbaums*

accuracy <- calculate\_accuracy(decision\_tree, features\_test, target\_test)

**print**(**sprintf**("Genauigkeit: %.2f%%", accuracy \* 100))

[1] "Genauigkeit: 76.34%"

Im Vergleich zu den Testdaten:

*# Berechne die Genauigkeit des Entscheidungsbaums*

accuracy <- calculate\_accuracy(decision\_tree, features\_train, target\_train)

**print**(**sprintf**("Genauigkeit: %.2f%%", accuracy \* 100))

[1] "Genauigkeit: 88.87%"

Es ist ein Unterschied von über 12% Vorhersagegenauigkeit. An der Stelle wird keine Modelloptimierung vorgenommen indem die Parameter so angepasst werden, dass die Vorhersagegenauigkeit besser wird.

Es wird aber geguckt wie sich die Vorhersagegenauigkeit mit unterschiedlichem Parameter verhält um die Stabilität des Models zu testen.

decision\_tree <- create\_decision\_tree(features\_train, target\_train, max\_depth = 5, min\_samples = 50, min\_gain = 0.1)

[1] "Train Genauigkeit: 88.10%"

[1] "Test Genauigkeit: 78.63%"

decision\_tree <- create\_decision\_tree(features\_train, target\_train, max\_depth = 4, min\_samples = 20, min\_gain = 0.1)

[1] "Train Genauigkeit: 88.68%"

[1] "Test Genauigkeit: 77.10%"

decision\_tree <- create\_decision\_tree(features\_train, target\_train, max\_depth = 2, min\_samples = 30, min\_gain = 0.1)

[1] " Train Genauigkeit: 66.99%"

[1] "Test Genauigkeit: 64.89%"

decision\_tree <- create\_decision\_tree(features\_train, target\_train, max\_depth = 5, min\_samples = 30, min\_gain = 0.1)

[1] "Train Genauigkeit: 88.68%"

[1] "Test Genauigkeit: 77.10%"

decision\_tree <- create\_decision\_tree(features\_train, target\_train, max\_depth = 4, min\_samples = 30, min\_gain = 0.2)

[1] "Train Genauigkeit: 82.73%"

[1] "Test Genauigkeit: 75.57%"

decision\_tree <- create\_decision\_tree(features\_train, target\_train, max\_depth = 4, min\_samples = 30, min\_gain = 0.05)

[1] "Train Genauigkeit: 88.87%"

[1] "Test Genauigkeit: 76.34%"

Tendenziell lässt sich aus diesem kleinen Test schließen, dass sich das Modell so verhält wie es sich verhalten soll. Je weniger Beschränkungen für die Erstellung des Entscheidungsbaum vorliegen desto besser werden die Train Vorhersagen wegen Overfitting und desto schlechter die Ergebnisse für Test.

Anschließend wird noch der Post-Pruning durchgeführt dazu wird der Entscheidungsbaum mit diesen Parametern verwendet:

decision\_tree <- create\_decision\_tree(features\_train, target\_train, max\_depth = 5, min\_samples = 50, min\_gain = 0.1)

[1] "Train Genauigkeit: 88.10%"

[1] "Test Genauigkeit: 78.63%"

pruned\_tree <- prune\_tree(decision\_tree, features\_test, target\_test)

*# Berechne die Genauigkeit des geprunten Entscheidungsbaums*

accuracy <- calculate\_accuracy(pruned\_tree, features\_test, target\_test)

**print**(**sprintf**("Genauigkeit nach Pruning: %.2f%%", accuracy \* 100))

[1] "Genauigkeit nach Pruning: 78.63%"

Die Genauigkeit ist genauso gut wie der Originale, da kein Post Pruning stattgefundet hat, da es sonst zu größeren Genauigkeit Einbussen käme.

Es wird noch getestet ob unterschiedliche Test- und Trainingsdaten unterschiedliche Entscheidungsbäume entstehen lassen.

|  |  |  |  |
| --- | --- | --- | --- |
| Random Seed | Train Genauigkeit | Test Genauigkeit | Genauigkeit Post-Pruning |
| 13 | 88.10% | 78.63% | 78.63% |
| 12 | 85.60% | 88.55% | 88.55% |
| 14 | 87.14% | 83.97% | 83.97% |

Tabelle 1: Genauigkeiten mit unterschiedlichen Random Seeds

Dieser kleine Test zeigt das unterschiedliche Trainigs- und Testdaten einen erheblichen unterschied machen. In den Versuch mit Random Seed 12 ist sogar die Test Genauigkeit geringer als die des Trainigs.

Der Entscheidungsbaum für den Random Seed 12:

MitgliedSportverein

├── j

│ └── Stellung

│ ├── Angestellter: hoch (85.71%)

│ ├── Arbeitslos: hoch (100.00%)

│ ├── Arbeiter: hoch (65.00%)

│ ├── Selbststaendig: hoch (100.00%)

│ └── Beamter: hoch (75.00%)

└── n

└── Einkommen <= 67232

├── Familienstand

│ ├── ledig: mittel (90.36%)

│ ├── verwitwet: niedrig (81.82%)

│ └── geschieden

│ └── Bundesland

│ ├── Saarland: niedrig (75.00%)

│ ├── Hamburg: mittel (80.00%)

│ ├── Sachsen-Anhalt: niedrig (100.00%)

│ ├── Schleswig-Holstein: niedrig (88.89%)

│ ├── Hessen: niedrig (100.00%)

│ ├── Sachsen: niedrig (75.00%)

│ ├── Nordrhein-Westfalen: niedrig (100.00%)

│ ├── Niedersachsen: niedrig (83.33%)

│ ├── Thueringen: niedrig (100.00%)

│ ├── Rheinland-Pfalz: niedrig (66.67%)

│ ├── Mecklemburg-Vorpommern: niedrig (60.00%)

│ ├── Berlin: niedrig (100.00%)

│ ├── Bayern: mittel (75.00%)

│ └── Baden-Wuerttemberg: mittel (100.00%)

└── verheiratet

└── Bundesland

├── Saarland: niedrig (100.00%)

├── Niedersachsen: niedrig (100.00%)

├── Hessen: niedrig (100.00%)

├── Mecklemburg-Vorpommern: niedrig (83.33%)

├── Thueringen: niedrig (87.50%)

├── Schleswig-Holstein: niedrig (66.67%)

├── Bayern: mittel (100.00%)

├── Baden-Wuerttemberg: mittel (75.00%)

├── Rheinland-Pfalz: niedrig (75.00%)

├── Nordrhein-Westfalen: niedrig (80.00%)

├── Hamburg: mittel (100.00%)

├── Sachsen: niedrig (100.00%)

├── Sachsen-Anhalt: niedrig (100.00%)

├── Schleswig-Holstein: niedrig (100.00%)

└── Berlin: niedrig (100.00%)

└── Einkommen > 67232: mittel (81.36%)

Und so für den für Random Seed 13:

MitgliedSportverein

├── j

│ └── Blatt: hoch (84.62%)

└── n

└── Einkommen <= 73516.5

├── Familienstand

│ ├── ledig: mittel (89.14%)

│ ├── verwitwet: niedrig (86.05%)

│ └── geschieden

│ └── Bundesland

│ ├── Sachsen: niedrig (100.00%)

│ ├── Nordrhein-Westfalen: niedrig (80.00%)

│ ├── Saarland: niedrig (100.00%)

│ ├── Hessen: niedrig (83.33%)

│ ├── Baden-Wuerttemberg: mittel (88.89%)

│ ├── Bayern: mittel (100.00%)

│ ├── Mecklemburg-Vorpommern: mittel (50.00%)

│ ├── Schleswig-Hollstein: niedrig (100.00%)

│ ├── Sachsen-Anhalt: niedrig (100.00%)

│ ├── Thueringen: niedrig (83.33%)

│ ├── Rheinland-Pfalz: niedrig (80.00%)

│ ├── Hamburg: mittel (100.00%)

│ ├── Niedersachsen: niedrig (100.00%)

│ ├── Berlin: niedrig (100.00%)

│ └── Schleswig-Holstein: niedrig (100.00%)

└── verheiratet

└── Bundesland

├── Sachsen: niedrig (100.00%)

├── Nordrhein-Westfalen: niedrig (80.00%)

├── Saarland: niedrig (100.00%)

├── Hessen: niedrig (83.33%)

├── Baden-Wuerttemberg: mittel (88.89%)

├── Bayern: mittel (100.00%)

├── Mecklemburg-Vorpommern: niedrig (80.00%)

├── Schleswig-Hollstein: niedrig (100.00%)

├── Sachsen-Anhalt: niedrig (100.00%)

├── Thueringen: niedrig (83.33%)

├── Rheinland-Pfalz: niedrig (80.00%)

├── Hamburg: mittel (100.00%)

├── Niedersachsen: niedrig (100.00%)

├── Berlin: niedrig (100.00%)

└── Schleswig-Holstein: niedrig (100.00%)

Beim besseren Entscheidungsbaum wird bei dem Split nach Mitglied im Sportverein bei der Ausprägung ja, nach der Stellung klassifiziert statt keiner mehr. Bei keinem Sportverein wird zwar auch nach Einkommen gesplittet aber nach einem anderen Schwelenwert.

Die Implementierung wurde durch vielzähliche Test mit einen kleinen überschaubaren Datensatz verifziert und validiert.

## Entscheidungsbaum Regeln

Für die letzten beiden Aufgaben des KDD, Funde interpretieren und das neue Wissen anwenden werden die Regeln des Entscheidungsbaum genau beschrieben (Fayyad/ Piatetsky-Shapiro/Smyth 1996b, S.31). Es wird der Entscheidungsbaums für den aus Random Seed 12 für die weitere Betrachtung genutzt. Zubeginn wird die Klassifikation eines Mitglieds des Sportvereins wird anhand von zwei Hauptkriterien bestimmt: j oder n, sowie weitere Faktoren wie Einkommen, Familienstand und Bundesland.

Wenn das Mitglied als j klassifiziert ist, hängt die Klassifikation von der Stellung des Mitglieds ab:

* Angestellte haben eine hohe Wahrscheinlichkeit von 85.71%.
* Arbeitslose haben eine hohe Wahrscheinlichkeit von 100.00%.
* Arbeiter haben eine hohe Wahrscheinlichkeit von 65.00%.
* Selbstständige haben eine hohe Wahrscheinlichkeit von 100.00%.
* Beamte haben eine hohe Wahrscheinlichkeit von 75.00%.

Für Mitglieder, die als n klassifiziert sind, erfolgt die Klassifikation basierend auf dem Einkommen:

* Wenn das Einkommen ≤ 67232 Euro beträgt, wird die Klassifikation zusätzlich durch den Familienstand und in einigen Fällen durch das Bundesland beeinflusst:
  + Ledige Mitglieder haben eine mittlere Klassifikationswahrscheinlichkeit von 90.36%.
  + Verwitwete Mitglieder haben eine niedrige Klassifikationswahrscheinlichkeit von 81.82%.
  + Geschiedene Mitglieder werden weiter nach ihrem Bundesland unterschieden:
    - Bundesländer wie Sachsen, Saarland und Hessen haben eine niedrige Klassifikationswahrscheinlichkeit von bis zu 100.00%.
    - Andere Bundesländer wie Hamburg oder Mecklenburg-Vorpommern können eine mittlere oder niedrigere Wahrscheinlichkeit haben.
  + Verheiratete Mitglieder werden ebenfalls nach Bundesland differenziert, wobei Bundesländer wie Bayern, Hamburg und Berlin tendenziell höhere Klassifikationswahrscheinlichkeiten aufweisen.
* Wenn das Einkommen > 67232 Euro beträgt, erhalten die Mitglieder eine mittlere Klassifikation mit einer Wahrscheinlichkeit von 81.36%.

Diese Regeln ermöglichen eine differenzierte Klassifikation basierend auf Einkommen, Familienstand, und regionalen Unterschieden innerhalb des Sportvereins.

## Genauere Untersuchung der Ergebnisse

Zum Schluss wird eine Heatmap zur besseren Darstellung der Ergebnisse präsentiert.

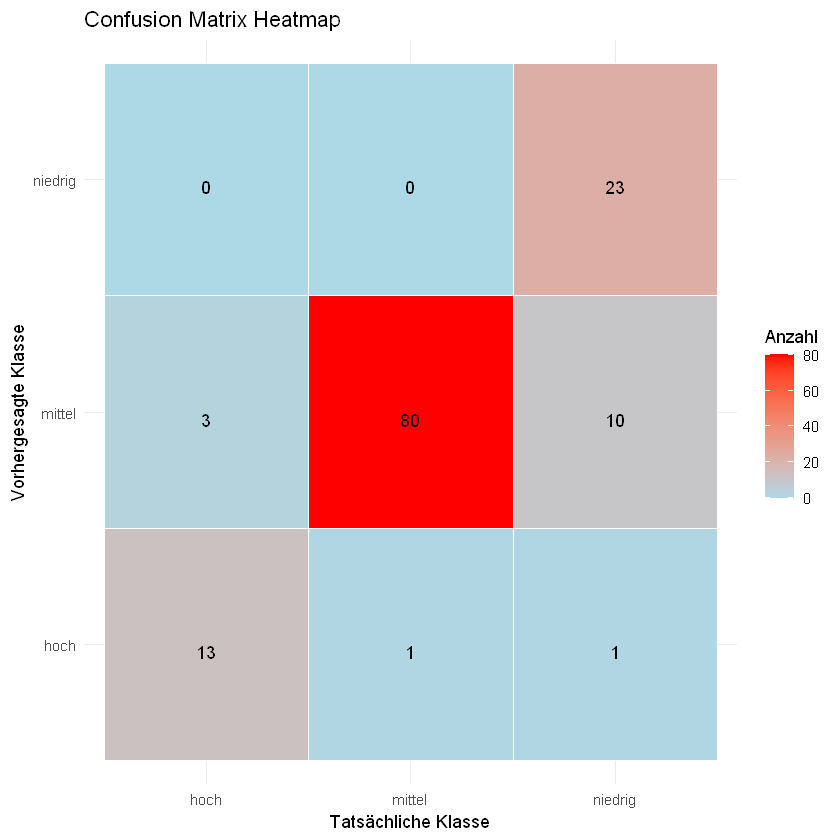


Tabelle 2: Heatmal der Ergebnisse

Das ist die Headmap der Ergebnisse des Entscheidungsbaums für den aus Random Seed 12. Die Diagnole von links Unten nach rechts Oben zeigen die richtig klassifizeirten Klassen. Die Klasse niedrig wird nie falsch klassifziert und die Klasse hoch wird 2 von 15-mal nur falsch klassifiziert. Die Klasse mittel mit den meisten auftretten mit 93 wird 80-mal richtig klassifiziert.

Ein Ausblick zu den falsch klassifizierten Kunden kann mit diesem Code angeguckt werden.

*# Funktion zur Identifizierung falsch klassifizierter Datensätze*

**find\_misclassified** <- function(predictions, actuals, data) {

  misclassified\_indices <- **which**(predictions != actuals)

  misclassified\_data <- data[misclassified\_indices, , drop = FALSE]

  return(misclassified\_data)

}

*# Berechne die Vorhersagen*

predictions <- **sapply**(1:**nrow**(features\_test), function(i) {

  row\_as\_list <- **as.list**(features[i, , drop = FALSE])

  classify(decision\_tree, row\_as\_list)

})

*# Finde die falsch klassifizierten Datensätze*

misclassified\_data <- find\_misclassified(predictions, target\_test, data)

*# Ausgabe der falsch klassifizierten Datensätze*

if (**nrow**(misclassified\_data) > 0) {

**cat**("Falsch klassifizierte Datensätze:\n")

**print**(misclassified\_data)

} else {

**cat**("Alle Datensätze wurden korrekt klassifiziert.\n")

}

Ein kleiner Ausblick:

Falsch klassifizierte Datensätze:

Alter Einkommen Abschluss Geschlecht Wohngebiet Stellung

2 30 33284 Studium w Grossstadt Arbeiter

3 40 32112 Realschule w Grossstadt Angestellter

4 29 26957 Hauptschule w Kleinstadt Angestellter

# Fazit

Im Rahmen dieser Arbeit wurde die Effektivität eines Entscheidungsbaums zur Klassifikation der Preiskategorie von Fahrrädern bei HaMa-Cycle untersucht. Die Klassifikation basierte auf Kundenmerkmalen, die durch externe Datenlieferanten bereitgestellt wurden, um präzisere Marketingstrategien zu entwickeln. Die Ergebnisse zeigen, dass Entscheidungsbäume ein nützliches Werkzeug für die Vorhersage von Preiskategorien darstellen. Durch die Anwendung des Entscheidungsbaums auf 80 Prozent der Daten und die anschließende Testung auf den verbleibenden 20 Prozent konnte die Vorhersagegenauigkeit ermittelt und die Stabilität der Klassifikationsregeln überprüft werden.

Die Analyse begann mit der gründlichen Exploration und Vorbereitung der Daten, um eine fundierte Trainingsbasis zu schaffen. Die Aufteilung der Daten in Trainings- und Testsets ermöglichte eine objektive Bewertung des Modells. Die Implementierung des Entscheidungsbaums in R zeigte, dass das Modell effektiv Muster in den Kundenmerkmalen identifizieren konnte, die auf die Preiskategorie der Fahrräder hinweisen.

Insgesamt bestätigt die Arbeit, dass Entscheidungsbäume nicht nur für die Klassifikation, sondern auch für die Optimierung von Geschäftsentscheidungen durch präzise und stabile Vorhersagen nützlich sind. Die gewonnenen Erkenntnisse können HaMa-Cycle dabei unterstützen, ihre Marketingstrategien gezielt anzupassen und effizienter auf Kundenbedürfnisse einzugehen.

Literaturverzeichnis

Alpar, P. & Niedereichholz, J. (Hrsg.). (2000). *XBusiness Computing*. *Data Mining im praktischen Einsatz: Verfahren und Anwendungsfälle für Marketing, Vertrieb, Controlling und Kundenunterstützung*. Vieweg+Teubner Verlag. https://doi.org/10.1007/978-3-322-89950-7

Behnisch, M. (2008). *Urban Data Mining : Operationalisierung der Strukturerkennung und Strukturbildung von Ähnlichkeitsmustern über die gebaute Umwelt [mit CD].* https://doi.org/10.5445/KSP/1000008458

Bertram, T. & Paschek, S. (2020). Methoden der Künstlichen Intelligenz. In M. Feiner & M. Schöllhorn (Hrsg.), *KI4Industry KI für den Mittelstand: Der Online-Kongress zum praktischen Einstieg in die Künstliche Intelligenz für KMU* (S. 6–14). Hochschule Karlsruhe.

Bibel, W. (2000). *Information Mining: Methoden, Algorithmen und Anwendungen Intelligenter Datenanalyse*. *Computational Intelligence Ser*. Springer Vieweg. in Springer Fachmedien Wiesbaden GmbH.

Christianti, D., Abdullah, S. & Nurrohmah, S. (2020). Bayes Risk Post-Pruning in Decision Tree to Overcome Overfitting Problem on Customer Churn Classification. In *Proceedings of the 1st International Conference on Statistics and Analytics, ICSA 2019, 2-3 August 2019, Bogor, Indonesia*. https://eudl.eu/doi/10.4108/eai.2-8-2019.2290487

Dai, W. & Ji, W. (2014). A MapReduce Implementation of C4.5 Decision Tree Algorithm. *International Journal of Database Theory and Application*, *7*(1), 49–60. https://doi.org/10.14257/ijdta.2014.7.1.05

Kirchberg, P. *Business Intelligence: Methoden der statistsichen Auswertung.*

Kurita, T. (2019). *Computer Vision: A Reference Guide* (Springer eBook Collection). Springer.

Leckliter, I. N., Matarazzo, J. D. & Silverstein, A. B. (1986). A literature review of factor analytic studies of the WAIS-R. *Journal of Clinical Psychology*, *42*(2), 332–342. https://doi.org/10.1002/1097-4679(198603)42:2<332::AID-JCLP2270420220>3.0.CO;2-2

Müller, R. M. & Lenz, H. J. (2013). *Business intelligence*. *eXamen.press*. Springer Vieweg.

Noble, W. S. (2006). What is a support vector machine? *Nature Biotechnology*, *24*(12), 1565–1567. https://doi.org/10.1038/nbt1206-1565

Petersohn, H. (2005). *Data Mining: Verfahren, Prozesse, Anwendungsarchitektur*. Zugl.: Leipzig, Univ., Habil.-Schr., 2004. Oldenbourg. https://www.degruyter.com/isbn/9783486593334 https://doi.org/10.1524/9783486593334

Ren, J., Lee, S. D., Chen, X., Kao, B., Cheng, R. & Cheung, D. (2009). Naive Bayes Classification of Uncertain Data. In *2009 Ninth IEEE International Conference on Data Mining.* IEEE. https://doi.org/10.1109/icdm.2009.90

Rokasc, L.; Miamon, O. Data Mining with Decision Trees: Theory and Applications. World Scientific 2014.

Runkler, T. A. (2015). *Data Mining: Modelle und Algorithmen intelligenter Datenanalyse* (2., aktualisierte Aufl. 2015). Springer Fachmedien Wiesbaden. https://doi.org/10.1007/978-3-8348-2171-3

Schendera, C. F. G. (2010). *Clusteranalyse mit SPSS: Mit Faktorenanalyse* (Wirtschaftsmathematik- und Statistik 8-2011). Oldenbourg.

To, A. C., Moore, J. R. & Glaser, S. D. (2009). Wavelet denoising techniques with applications to experimental geophysical data. *Signal Processing*, *89*(2), 144–160. https://doi.org/10.1016/j.sigpro.2008.07.023

Vinutha, H. P., Poornima, B. & Sagar, B. M. (2018). Detection of Outliers Using Interquartile Range Technique from Intrusion Dataset. *Information and Decision Sciences*, *701*, 511–518. https://doi.org/10.1007/978-981-10-7563-6\_53

Xu, J. (2023). Systematic Analysis and Application Prospect of Decision Tree. *Highlights in Science, Engineering and Technology*, *71*, 163–170. https://doi.org/10.54097/hset.v71i.12687

Zhang, X. D. (2011). Illustration of SSMD, z score, SSMD\*, z\* score, and t statistic for hit selection in RNAi high-throughput screens. *Journal of biomolecular screening*, *16*(7), 775–785. https://doi.org/10.1177/1087057111405851

Ich versichere, dass ich das beiliegende Assignment selbstständig verfasst, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie alle wörtlich oder sinngemäß übernommenen Stellen in der Arbeit gekennzeichnet habe.