Begleitskript zum Unterrichtsmodul: „Lernende Algorithmen“

Liebe Kollegin, lieber Kollege,

das vorliegende Skript ist als unterstützendes Begleitmaterial zu den von uns entwickelten Arbeitsblättern gedacht und richtet sich an Lehrpersonen, die eine Unterrichtseinheit zum Thema *Lernen Algorithmen* durchführen wollen. Im Skript stellen wir ergänzende fachliche und fachdidaktische Informationen bereit, legen Intentionen und Lösungsvorschläge zu den Aufgaben offen und skizzieren mögliche Wege zu deren Einbindung im Unterricht. Ausdrücklich handelt es sich um Vorschläge, von denen Sie natürlich abweichen können.

Über Hinweise, Erfahrungsberichte und Verbesserungsvorschläge freuen wir uns.

Ziele und Zielgruppe

Das Unterrichtsmodul richtet sich an Schülerinnen und Schüler in Informatikkursen der Sekundarstufe II. Folgende Kenntnisse werden vorausgesetzt:

- (binäre) Bäume
  - Begriffe: Wurzel, Knoten, Kanten, Blätter, Höhe, Vorgänger, Nachfolger (Kinder)
  - Baumtraversierung
- Algorithmen: Begriffsbedeutung, Darstellung in Form von Pseudocode o.ä.; imperative Programmierung
- Programmierkenntnisse in Java bzw. Python (zur Bearbeitung des plugged-Teils)

Das Unterrichtsmodul verfolgt das Ziel, Schülerinnen und Schülern am Beispiel von *lernenden Algorithmen* einen ersten Einblick in die Welt *künstlicher Intelligenz* zu ermöglichen. Sie lernen dabei wesentliche Unterschiede zwischen lernenden Algorithmen und klassischen, nicht-lernenden Algorithmen kennen. Dadurch sollen sie in die Lage versetzt werden, Einsatzmöglichkeiten, Chancen und Risiken von Entwicklungen im Bereich der künstlichen Intelligenz zu erfassen und grob zu beurteilen. Somit versteht sich das Modul als Beitrag zur Allgemeinbildung im Zeitalter der fortschreitenden und alle Lebensbereiche erfassenden Digitalisierung.

Inhalte

Im Rahmen der Unterrichtseinheit werden folgende Inhalte behandelt:

1. Grundlegendes Datenmodell (Merkmale, Ausprägungen)
2. Klassifikation mithilfe von Entscheidungsbäumen
3. Intuitive Konstruktion und Vergleich von Entscheidungsbäumen
4. Attributauswahl: Gini-Index
5. Induktive Baumerzeugung mithilfe des gewichteten Gini-Indexes
6. Implementierung

Saarbrücken, im August 2019

Stefan Strobel
Gymnasium am Rotenbühl, Saarbrücken
Universität des Saarlandes
strobel@cs.uni-saarland.de

Pascal Schmidt
Gymnasium Saarpfalz, Homburg
Universität des Saarlandes
pascal.schmidt@uni-saarland.de

Begleitskript zum Unterrichtsmodul: „Lernende Algorithmen“

1. Grundlegendes Datenmodell

Bei den in diesem Unterrichtsmodul verwendeten Daten handelt es sich um *Tupel*, welche die *Ausprägungen* eines Objekts in den betrachteten *Merkmalen* angeben. Möglicherweise kennen die Schülerinnen und Schüler solche Daten aus dem Bereich der Objekt- oder datenorientierten Modellierung (z.B. ER-Modellierung), wo Objekte ebenfalls durch die Ausprägungen ihrer Attribute beschrieben werden. Wir halten es dennoch für sinnvoll, das verwendete Datenmodell an einem konkreten Beispiel einzuführen.

Beispiel
Die Tabelle zeigt einen Ausschnitt aus einem Kalender für den Monat Februar 2019.

1	Fr	
2	Sa	
3	So	
4	Mo	
5	Di	
6	Mi	
7	Do	
8	Fr	
9	Sa	
10	So	
11	Mo	
12	Di	
13	Mi	

Die Tage des **Wochenendes**, Samstag und Sonntag, sind farbig hervorgehoben. Für einige Tage seien zusätzliche Informationen hinsichtlich des **Wetters** und der **Schneesituation** bekannt:

- Wetter: Ist es an dem Tag *sonnig*?
- Schneesituation: Ist das nächstgelegene Skigebiet mit guten Schneeverhältnissen weniger oder mehr als 100 Kilometer vom Wohnort einer Person entfernt?

Die berechtigte und natürliche Frage, wie die Subsumtion konkreter Tage unter die Begriffe *„gute Schneeverhältnisse“* (Schneehöhe? Wetter? Wind? Nebel?) oder *„sonnig“* (Sonnenscheindauer?) erfolgt bzw. erfolgen könnte, sollte unseres Erachtens im Unterrichtsgespräch nicht übergangen werden. Zugleich halten wir es für sinnvoll, den Schülerinnen und Schülern transparent zu machen, dass Fragen nach der „Entstehung“ von Daten bzw. der durchgeführten Aggregation im Verlauf des Unterrichtsmoduls nicht im Vordergrund stehen. Zur Vermeidung wiederkehrender Diskussionen – auch über Sinn und Unsinn von Regelwerken/Entscheidungsbäumen – sind viele Übungsaufgaben abstrakt gehalten und gerade nicht aus der Lebenswelt der Schülerinnen und Schüler gewählt. Das Mittel der Abstraktion erlaubt hier eine Fokussierung auf die wesentlichen Aspekte der Konstruktion und Analyse von Entscheidungsbäumen.

Der bereits gezeigte kalendarische Ausschnitt könnte mit zusätzlichen Informationen dann wie folgt aussehen:

1	Fr	s, <100 km
2	Sa	s, >100 km
3	So	ns, >100 km
4	Mo	
5	Di	ns, <100 km
6	Mi	s, < 100 km
7	Do	s, > 100 km
8	Fr	
9	Sa	
10	So	s, < 100 km
11	Mo	
12	Di	
13	Mi	ns, > 100 km

Somit liegen für 8 Tage die Ausprägungen der Merkmale Wetter, Wochenende und Entfernung vor, die wir in Form einer *Tabelle* oder kurz als *Tupel* notieren können. Zusätzlich ergänzen wir das künstliche Attribut **ID**, um auch solche Datenelemente voneinander unterscheiden zu können, die in allen betrachteten Merkmalen übereinstimmen. Wir bevorzugen die ID aus Gründen der Handhabbarkeit gegenüber dem kalendarischen Datum (ID: 1 vs. Datum: 01.02.2019)

Darstellung als *Tabelle*:

	Merkmale & Ausprägungen		
ID	**Sonne?**	**WE?**	**Entf?**
1	ja	nein	< 100
2	ja	ja	> 100
3	nein	ja	> 100
4	nein	nein	< 100
5	ja	nein	< 100
6	ja	nein	< 100
7	ja	ja	< 100
8	nein	nein	> 100

Darstellung als *Tupel*:

```
(ja,        nein,       <100)
(ja,        ja,         >100)
(nein,      ja,         >100)
(nein,      nein,       <100)
(ja,        nein,       <100)
(ja,        nein,       <100)
(ja,        ja,         <100)
(nein,      nein,       >100)
```

Im Unterrichtsmodul werden durchgehend Merkmale verwendet, die über lediglich zwei Ausprägungen verfügen (*binäre Merkmale*). Dies ist der Einfachheit geschuldet, da binäre Bäume leichter zu handhaben sind. Grundsätzlich ist eine Erweiterung auf mehrwertige Merkmale jedoch denkbar.

2. Klassifikation mithilfe von Entscheidungsbäumen

Zur grundlegenden Einführung von Entscheidungsbäumen dient Arbeitsblatt 1.
Die Schülerinnen und Schüler lernen hier Entscheidungsbäume als *Regelwerke* kennen, die gegebene Datenelemente in eine von mehreren Klassen einordnen (*klassifizieren*). Jeder Entscheidungsbaum beantwortet genau eine Frage. Die Frage, die der auf Arbeitsblatt 1 beschriebenen Situation zugrunde liegt, lautet: *Lohnt es sich unter den gegebenen Voraussetzungen, Skifahren zu gehen*? Als Input werden Datenelemente der in Kapitel 1 erläuterten Art verwendet.
Die Schülerinnen und Schüler lernen bei der Bearbeitung der Aufgaben die grundlegende Funktionsweise von Entscheidungsbäumen kennen (Traversierung von der Wurzel bis zu einem Blatt, welches die Klasse liefert) und gewinnen die Einsicht, dass eine vorliegende Menge von Daten von mehreren unterschiedlichen Bäumen gleich klassifiziert werden kann. Damit werden sie auch auf die Mehrdeutigkeit vorbereitet, der sie bei der Konstruktion eines Entscheidungsbaumes zu gegebenen Daten begegnen.

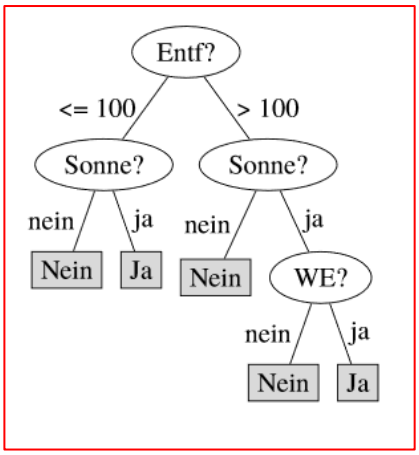
Lösungen zu Arbeitsblatt 1:

a) Bestimme Björns Entscheidung in folgende Situationen, in Abhängigkeit von dem von ihm gewählten Entscheidungsbaum.

	Merkmale			*Entscheidung*	
ID	**Sonne?**	**WE?**	**Entf?**	*e1*	*e2*
1	nein	ja	<= 100	Nein	Nein
2	ja	ja	<= 100	Ja	Ja
3	ja	nein	> 100	Nein	Nein
4	nein	nein	<= 100	Nein	Nein
5	ja	ja	> 100	Ja	Ja

b) (*Zusatzaufgabe*) Erstelle einen weiteren Entscheidungsbaum, der in *allen* Situationen die gleiche Antwort liefert, aber in der Wurzel das Merkmal *Entf(ernung)* abfragt.

Mehrere Lösungen denkbar, beispielsweise



3. Intuitive Konstruktion und Vergleich von Entscheidungsbäumen

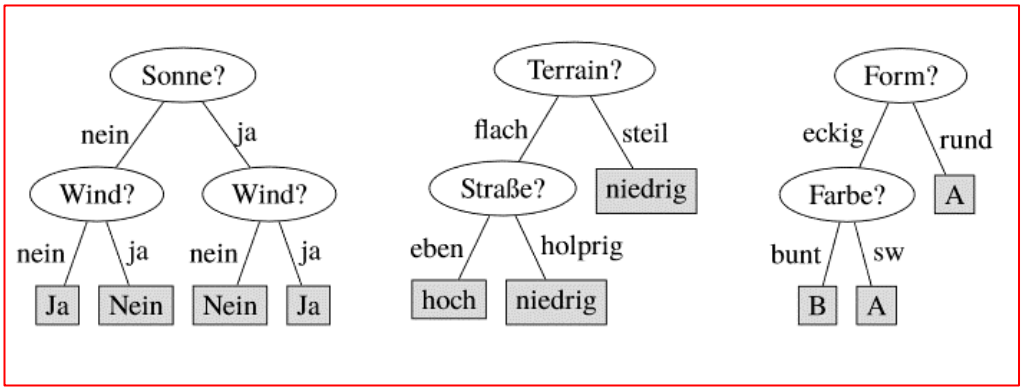
Die Arbeitsblätter 2 und 3 leiten die Schülerinnen und Schüler zur Erstellung von Entscheidungsbäumen an. Das Vorgehen soll in dieser Phase noch frei von Regeln erfolgen, um die Mehrdeutigkeit der Konstruktion erfahren zu können. Es ist hier (noch) nicht festgelegt, nach welchen Kriterien die Merkmalauswahl in den einzelnen Knoten erfolgt.
Somit können viele unterschiedliche Bäume entstehen, die sich in ihrer Gestalt unterscheiden. Diese Erkenntnis leitet später zu der zentralen Frage über: *Wie finde ich einen möglichst kompakten Baum, ohne zuvor alle denkbaren Bäume konstruieren zu müssen*?

Lösungen zu Arbeitsblatt 2:

a) Welche Merkmale werden in den verschiedenen Situationen betrachtet? Welche Entscheidung soll getroffen werden?

- Situation 1: Anhand der Wetterbedingungen (Sonne und Wind) wird entschieden, ob man Sport machen kann.
- Situation 2: Anhand der Beschaffenheit des Terrains und der Straße wird entschieden, ob man schnell fahren kann oder langsam fahren muss.
- Situation 3: Anhand von Form, Farbe und Größe werden Objekte den (abstrakten) Klassen A bzw. B zugeordnet.

b) Erstelle zu jeder Situation einen Entscheidungsbaum, der alle angegebenen Daten korrekt klassifiziert.



Die Lösungen sind beispielhaft; insbesondere zu Situation 3 sind andere Lösungen denkbar. Die dargestellte Lösung ist optimal hinsichtlich der Größe des Baumes und ergibt sich aus der Beobachtung, dass die Größe offenbar für die Klassifikation keine Rolle spielt. Arbeitet man die Merkmale bei der Baumerzeugung hingegen in der angegebenen Reihenfolge ab, so erhält man einen größeren Baum.

Ergänzung: Am Beispiel des linken bzw. rechten (unvollständigen) Binärbaumes sollte mit den Schülerinnen und Schülern besprochen werden, dass ein Pfad „vorzeitig“ in einem Blatt enden kann, wenn die bis dahin abgefragten Merkmale bereits ausreichen, um die Daten zu klassifizieren. Dies vermeidet unnötige Aufblähungen des Baumes.

Lösungen zu Arbeitsblatt 3:

a) Erstellt (arbeitsteilig) vier Entscheidungsbäume, die zu genau diesen Entscheidungen führen. Verwendet hierbei jeweils unterschiedliche Startmerkmale für die Abfrage in der Wurzel.

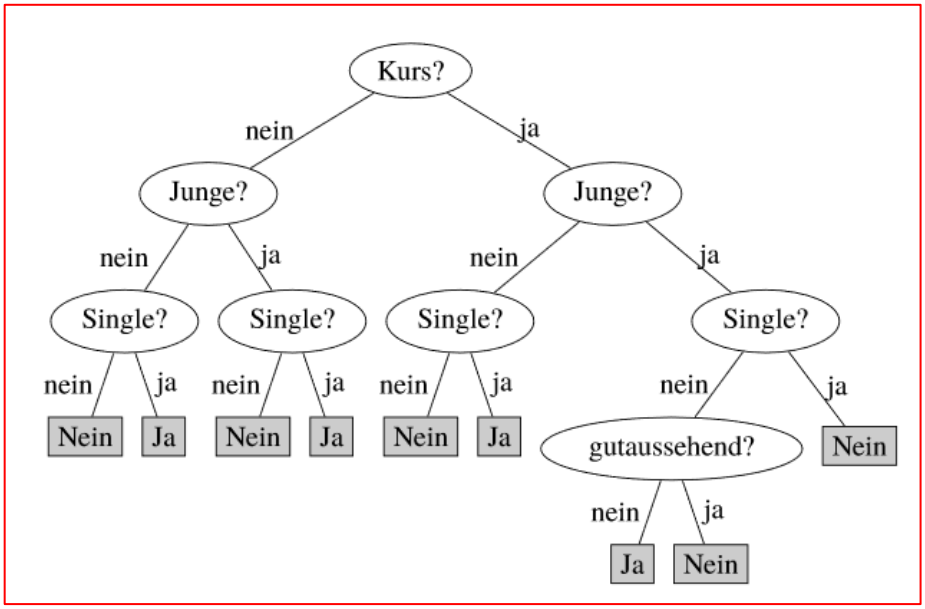
Da trotz vorgegebenem Anfangsmerkmal verschiedene Testreihenfolgen möglich sind, wird beispielhaft nur eine Lösung angegeben (s.u.). Die Tests auf einer Ebene sind hier jeweils identisch gewählt; ihre Reihenfolge entspricht der Reihenfolge der Merkmale in der Tabelle.

b) Vergleicht die in eurer Gruppe entstandenen Bäume hinsichtlich Höhe und Knotenanzahl.

Die Bäume unterscheiden sich ggf. in der Höhe, zumindest aber in der Anzahl der Knoten (Tests).

c) Gibt es noch weitere Entscheidungsbäume, die alle Daten korrekt klassifizieren? Begründet eure Antwort.

Tatsächlich gibt es noch sehr viele weitere Entscheidungsbäume. Nachdem das Merkmal für die Wurzel festgelegt ist, gibt es noch 3! Möglichkeiten, die anderen drei Merkmale in eine Reihenfolge zu bringen und dementsprechend auf den folgenden Ebenen abzufragen. Hinzu kommen zahlreiche weitere Möglichkeiten, da nicht in allen Knoten einer Ebene des Baumes das gleiche Merkmal abgefragt werden muss.



Ergänzung: Ein Beispiel für einen Entscheidungsbaum, bei dem auf einer Ebene unterschiedliche Merkmale abgefragt werden, findet sich auf Arbeitsblatt 4. Zugleich handelt es sich bei dem dort dargestellten Baum um einen Baum möglichst geringer Höhe und Knotenanzahl. Die Verwendung von Arbeitsblatt 4 ist nicht erforderlich, wenn die Schülerinnen und Schüler von sich aus auf die Idee kommen, die Daten zu analysieren/interpretieren und einen möglichst kompakten Baum zu erzeugen. In unseren bisherigen Unterrichtsversuchen war dies jedoch stets der Fall, sodass die auf dem Arbeitsblatt beschriebenen Fragestellungen anschließend mündlich im Unterrichtsgespräch geklärt werden können. Hierbei steht die Erarbeitung der Vorteile kleinerer Entscheidungsbäume im Zentrum:

- Weniger Knoten bedeutet weniger Tests zur Klassifikation eines Datenelements und damit höhere *Effizienz.* (Streng genommen gilt dies nur unter der Annahme, dass sich die Daten bei den einzelnen Tests gleichmäßig auf die beiden Alternativen verteilen)
- Kleinere Strukturen sind besser *interpretierbar*: Dies zeigt der auf Arbeitsblatt 4 abgebildete Entscheidungsbaum – gerade auch im Kontrast zu den zuvor konstruierten Bäumen – in anschaulicher Weise.

Lösungen zu Arbeitsblatt 4:

a) Zeige exemplarisch anhand von 2-3 Beispielen, dass dieser Baum die Datenelemente von Aufgabenblatt 3 korrekt klassifiziert.

individuelle Lösungen

b) Vergleiche diesen Baum mit den zuvor erstellen Entscheidungsbäumen (vgl. Arbeitsblatt 3). Welche Unterschiede fallen dir auf? Welche(n) Vorteil(e) bietet dieser Entscheidungsbaum gegenüber den anderen Bäumen?

Der Baum enthält weniger Knoten und hat eine geringere Höhe. Zur Klassifikation von Datenelementen ist daher im Durchschnitt eine geringere Zahl an Tests erforderlich.

c) Interpretiere den Baum als *Regelwerk*: Formuliere eine Bedingung, die eine Person erfüllen muss, um zu Peters Party eingeladen zu werden *(Formulierungshilfe: „Wenn …, dann …“)*.

Wenn eine Person weiblich und Single ist oder männlich und nicht gutaussehend, dann wird sie zu Peters Party eingeladen. Umgekehrt: Wenn eine Person weiblich und verpartnert oder männlich und gutaussehend ist, dann wird sie nicht eingeladen.

d) Leite aus dem Entscheidungsbaum eine begründete Vermutung über Peters Motiv bei der Auswahl seiner Gäste her.

Peter ist möglicherweise Single und auf der Suche nach einer Partnerin. Daher lädt er nur Frauen ein, die noch nicht vergeben sind. Zudem lädt er keine gutaussehenden Männer ein, da er diese als mögliche Konkurrenten betrachtet.

4. Attributauswahl: Gini-Index

Die Schülerinnen und Schüler haben inzwischen erfahren, dass zu einer gegebenen Situation verschiedene Entscheidungsbäume konstruiert werden können, die sich in ihrer Größe und damit in ihrer *Effizienz* und *Interpretierbarkeit* unterscheiden. Die daraus abgeleitete, zentrale Frage lautet: *Kann man auf systematische Weise – also regelbasiert - einen möglichst kleinen Baum finden, ohne alle denkbaren Bäume erzeugen zu müssen?*
Die Arbeitsblätter 5 und 6 verfolgen das Ziel, einen *gierigen* Algorithmus zu motivieren, der zu meist guten Ergebnissen führt. Er fußt auf folgender Beobachtung: Bei der Klassifikation passieren alle Datenelemente den Wurzelknoten und traversieren den Baum, bis sie schließlich Blätter erreichen, in denen sich nur noch Daten der gleichen Klasse sammeln. Der gierige Algorithmus besagt nun: Wähle in jedem Schritt der Baumerzeugung *ein* Merkmal, welches die Unordnung (bzgl. der Klassenzuordnung) in den Kinderknoten am stärksten verringert. (Wir schreiben „ein“ und nicht „das“ Merkmal, da es mehrere gleichermaßen geeignete Merkmale geben kann.)
Daraus ergibt sich zugleich die Notwendigkeit, (Un-)Ordnung zu quantifizieren. Hierfür eignet sich der *Gini-Index*, da er einfach zu berechnen und einer anschaulichen Erläuterung bzw. Interpretation zugänglich ist. Wir liefern in der folgenden Infobox einige fachliche und didaktische Hinweise zum Gini-Index und zu dessen Einführung im Unterricht. Nach der Einführung durch die Lehrperson kann mithilfe von Arbeitsblatt 7 der *gewichtete Gini-Index* als Attributauswahlmaß erarbeitet werden.

FACHLICHE ERGÄNZUNG: *Gini-Index*

Für eine Datenliste *L,* deren Elemente sich auf die beiden Klassen c_0 und c_1 aufteilen und mit relativer Häufigkeit p_0 bzw. p_1 vorkommen, berechnet sich der *Gini-Index* wie folgt:

$$\begin{aligned} \text{Gini}(L) &:= p_0 \cdot (1-p_0) + p_1 \cdot (1-p_1) && \text{[Defintion für zwei Klassen]} \\ &= p_0 \cdot p_1 + p_1 \cdot p_0 && [p_0 + p_1 = 1] \\ &= 2 \cdot p_0 \cdot p_1 \end{aligned}$$

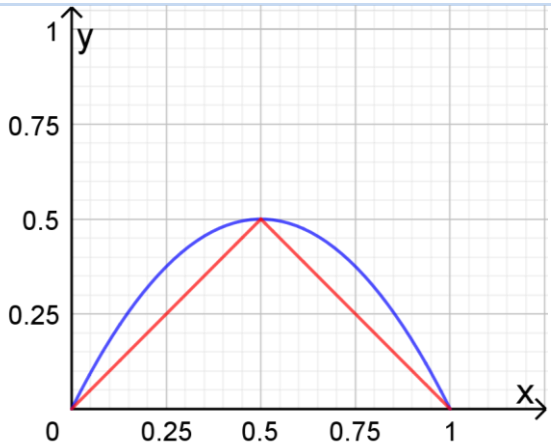
Die rechte Seite der ersten Gleichung beschreibt die Wahrscheinlichkeit, mit der ein zufällig ausgewähltes Datenelement falsch klassifiziert wird, wenn die Klassifikation zufällig entsprechend der Klassenverteilung erfolgt. So erhält man bei zufälliger Wahl mit einer Wahrscheinlichkeit von p_0 ein Datenelement der Klasse c_0 und klassifiziert dies anschließend mit der Wahrscheinlichkeit $p_1 = 1 - p_0$ fälschlicherweise als zur Klasse c_1 gehörend. Die Veranschaulichung dieser Interpretation mittels des aus der Mathematik bekannten Urnenmodells und konkreter Zahlen, z.B. in Form von farbigen Kugeln, bietet sich an und hat sich im Unterricht bewährt.
Die Fehlklassifikationswahrscheinlichkeit ist umso geringer, je homogener die Datenliste ist: Wenn alle Datenelemente der gleichen Klasse angehören, ist sie 0. Liegen beide Klassen gleichverteilt vor, so erfolgt eine Fehlklassifikation mit einer Wahrscheinlichkeit von 0,5. Somit ist der Wertebereich des Gini-Index [0; 0.5].
Eine andere Interpretation dieses Terms ist die folgende, formuliert in der Sprache des Urnenmodells: Zieht man nacheinander (mit Zurücklegen) aus der Liste *L* zufällig zwei Datenelemente, so beschreibt der Term die Wahrscheinlichkeit, Elemente verschiedener Klassen zu ziehen.
Beide Überlegungen können genutzt werden, um den Schülerinnen und Schülern den Gini-Index zur Quantifizierung von (Un-)Ordnung plausibel zu machen.

Durch Ausmultiplizieren lässt sich die erste Gleichung in folgende äquivalente Form bringen:

$$\text{Gini}(L) = (p_0 - p_0^2) + (p_1 - p_1^2) = (p_0 + p_1) - (p_0^2 + p_1^2) = 1 - (p_0^2 + p_1^2)$$

Der Term $1 - (p_0^2 + p_1^2)$ lässt sich mit obigen Überlegungen begründen, indem man die Wahrscheinlichkeit über das Gegenereignis berechnet: $p_0^2 + p_1^2$ beschreibt gerade die Wahrscheinlichkeit, zwei Elemente aus c_0 oder c_1 zu ziehen.

Nach Elimination einer Variablen (es gilt $p_0 + p_1 = 1$) kann der Gini-Index als gewöhnliche einstellige Funktion dargestellt werden. Die folgende Abbildung zeigt den Graphen der Funktion $f(p_0) = 2 \cdot p_0 \cdot (1-p_0)$.



Lage und Art der Extrema wurden bereits durch die stochastische Interpretation geklärt. Möglicherweise stören sich Schülerinnen und Schüler aber an dem nichtlinearen Verlauf zwischen den Extrema: Ausgehend von einer maximal heterogenen Datenliste (p_0 = 0,5) wirkt sich eine Zu- oder Abnahme von p_0 zunächst nur geringfügig auf den Gini-Index aus. Wir ergänzen daher den Graphen einer zweiten Funktion mit identischen Extrema und linearem Verlauf.
Mathematisch handelt es sich bei dem roten Graphen um die zweistellige Minimumfunktion von p_0 und p_1 oder, in Abhängigkeit von p_0 formuliert, um folgende abschnittsweise definierte Funktion:

$$g(p_0) = \begin{cases} p_0 & \text{falls} \quad p_0 \leq 0,5 \\ 1 - p_0 & \text{sonst} \end{cases}$$

Sofern im Unterricht – wie in diesem Modul vorgesehen – nur Situationen mit zwei Klassen behandelt werden, spricht wenig gegen diese Alternative. Sie ist aufgrund der Fallunterscheidung lediglich etwas umständlicher zu berechnen. Allerdings ist sie im Gegensatz zum Gini-Index nicht ohne weiteres auf den Fall mit mehreren Klassen übertragbar. Bei nur zwei Klassen könnte es interessant sein, die beiden Maße und die damit erzielten Resultate miteinander zu vergleichen.

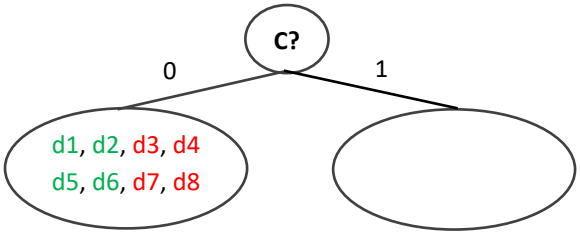
Der Gini-Index ist auch für Mengen mit mehr als nur zwei Klassen definiert. Er kann daher auch bei nicht-binären Klassifikationen benutzt werden. Allerdings empfiehlt sich dann die Verwendung der zweiten Variante (Gegenereignis), da nur auf diese Art ein übersichtlicher Term entsteht:
Teilen sich die Element der Datenliste L auf n Klassen c_0, …, c_{n-1} auf und kommen jeweils mit den relativen Häufigkeiten p_0, …, p_{n-1} vor, so ist die Wahrscheinlichkeit, ein zufällig ausgewähltes Datenelement korrekt zu klassifizieren, wenn die Klassifikation zufällig entsprechend der Klassenverteilung erfolgt, $p_0^2 + p_1^2 + \cdots + p_{n-1}^2$. Die Wahrscheinlichkeit einer falschen Klassifikation ist somit

$$\text{Gini}(L) = 1 - (p_0^2 + p_1^2 + \cdots + p_{n-1}^2)$$

Da wir uns in diesem Modul auf binäre Klassifikationen beschränken, bevorzugen wir im weiteren Verlauf zur Berechnung des Gini-Index einer Datenliste L die Formel Gini(L) = $2 \cdot p_0 \cdot p_1$.

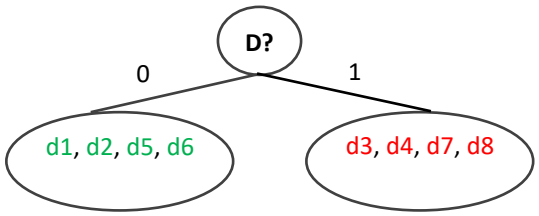
Lösungen zu Arbeitsblatt 5:

a) Stelle die Situation für einen initialen Test anhand von Merkmal C dar und begründe, dass sich dieses Merkmal nicht eignet.



Ein Split anhand von Merkmal C führt zu einem leeren rechten Kinderknoten, sodass sich die Anzahl der Datenelemente im linken Kinderknoten im Vergleich zum Vaterknoten nicht verringert. Somit führt der Split nicht zu einer Verbesserung. Mehr noch: Fortlaufende Splits dieser Art können zu einer Endlosschleife führen (wichtig für den plugged-Teil!).

b) Stelle die Situation für einen initialen Test anhand von Merkmal D dar und begründe, dass es sich besonders eignet.



Ein Split anhand von Merkmal D führt zu Aufteilung der Daten in zwei homogene Teillisten, sodass der Algorithmus nach diesem Schritt mit der entsprechenden Klassifizierung enden kann.

c) Während die Splits anhand von Merkmal C und D gewissermaßen Extreme darstellen, zeigt folgende Abbildung (Splits nach A bzw. B) Situationen, die in dieser oder ähnlicher Form häufig vorkommen.

Nimm an, du müsstest dich für eine dieser beiden Varianten entscheiden: Welche wählst du? Begründe deine Antwort!

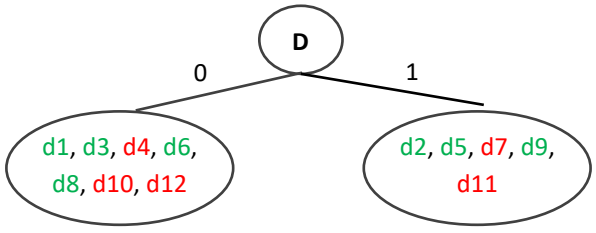
Der Split nach Attribut A bewirkt keinen Fortschritt: Beide Teillisten enthalten – wie auch die Ausgangsliste – jeweils 50% Daten mit Klasse „Ja“ bzw. „Nein“.
Der Split nach Attribut B generiert hingegen je eine Teilliste, in der eine Sorte von Elementen überwiegt. Der Split nach B kommt damit dem Ziel – Separierung der Daten nach ihrer Klasse – ein Stück näher und sollte bevorzugt werden.

Ergänzung: Ein Split nach Attribut A ist dennoch deutlich besser als ein Split nach C, denn zumindest verringert sich hier die Anzahl der Datenelemente pro Knoten. Bei Splits dieser Art besteht nicht die Gefahr einer Endlosschleife.

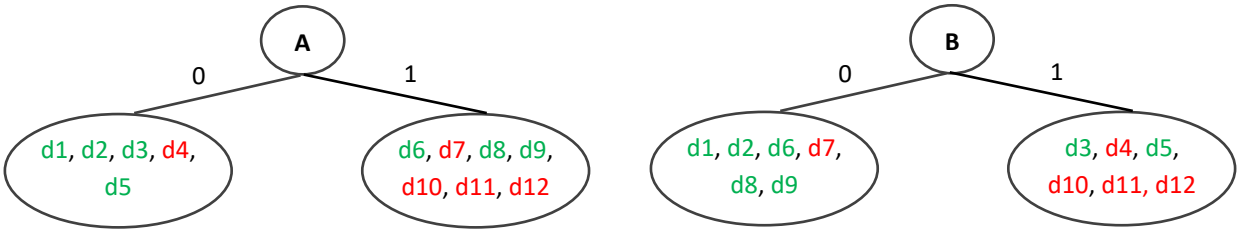
Lösungen zu Arbeitsblatt 6:

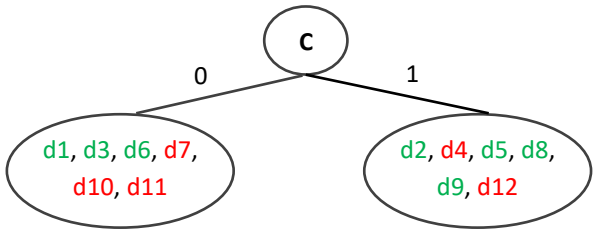
Aufgabe 1

a) Die folgende Abbildung zeigt die Situation für einen Split nach Merkmal D. Stelle die Situation jeweils dar, wenn für den ersten Split eines der anderen Merkmale verwendet wird.



Weitere Lösungen:





b) Welches dieser vier „Anfangsstücke“ würdest du zur Erstellung eines Entscheidungsbaumes nutzen? Begründe deine Entscheidung!
Die Splits nach D bzw. C führen zu zwei Teillisten, in denen jeweils Elemente beider Klassen relativ gleichmäßig verteilt sind, wohingegen die Splits A und B eine erkennbare „Vorsortierung“ bewirken: in beiden Teillisten dominieren jeweils Datenelemente einer der beiden Klassen. Beim Split nach B (5:1 und 4:2) ist dieses Übergewicht am deutlichsten. Daher ist ein Split nach Merkmal B zu bevorzugen.

Aufgabe 2

a) Berechne den *Gini-Index* der Datenliste aus Aufgabe 1.

$p_0 = \frac{7}{12}, p_1 = \frac{5}{12}$

$\Rightarrow gini(L) = \frac{35}{72} \approx 0{,}4861$

Der nahe am Maximum liegende Wert erklärt sich dadurch, dass die beiden Klassen in der Liste nahezu gleichverteilt vorkommen.

b) Bestimme für die beiden Kinderknoten, die in Aufgabe 1 bei einem Split nach Merkmal D bzw. Merkmal B entstehen, jeweils den *Gini-Index* der enthaltenen Daten.

Split nach D:
$L_{D=0}$ = { d1, d3, d4, d6, d8, d10, d12}
$gini(L_{D=0}) = 2 \cdot \frac{4}{7} \cdot \frac{3}{7} = \frac{24}{49} \approx 0{,}4898$

$L_{D=1}$ = { d2, d5, d7, d9, d11 }
$gini(L_{D=1}) = 2 \cdot \frac{3}{5} \cdot \frac{2}{5} = \frac{12}{25} \approx 0{,}4800$

Split nach B:
$L_{B=0}$ = { d1, d2, d6, d7, d8, d9 }
$gini(L_{B=0}) = 2 \cdot \frac{5}{6} \cdot \frac{1}{6} = \frac{5}{18} \approx 0{,}2778$

$L_{B=1}$ = { d3, d4, d5, d10, d11, d12 }
$gini(L_{B=1}) = 2 \cdot \frac{2}{6} \cdot \frac{4}{6} = \frac{8}{18} \approx 0{,}4444$

Lösungen zu Arbeitsblatt 7:

a) Berechne den *Gini-Index* für jeden der vier dargestellten Kinderknoten.

$gini(L_{A=0}) = 0$

$gini(L_{A=1}) = 2 \cdot \frac{35}{85} \cdot \frac{50}{85} \approx 0{,}4844$

$gini(L_{B=0}) = 2 \cdot \frac{45}{60} \cdot \frac{15}{60} \approx 0{,}3750$

$gini(L_{B=1}) = 2 \cdot \frac{5}{40} \cdot \frac{35}{40} \approx 0{,}2188$

b) Wie könnte man Julias bzw. Peters Standpunkt unter Verwendung des *Gini-Index* begründen? Liefere für jeden der beiden (mindestens) ein Argument.

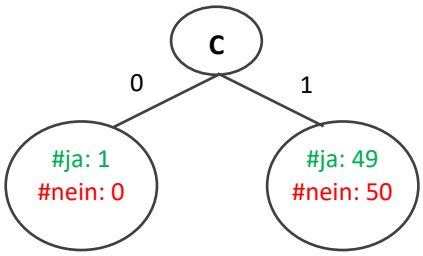
- Argument für Peter (Split A): Bei diesem Split entsteht der Knoten, der den kleinsten als Gini-Indizes hat.
- Argument für Peter (Split A): Der Mittelwert der beiden Gini-Indizes ist kleiner als bei einem Split nach B.
- Argument für Julia (Split B): Bei einem Split nach A entsteht der Knoten, der den größten alle Gini-Indizes hat.
- Argument für Julia (Split B): Es entstehen zwei Knoten, die beide deutlich homogener als der ursprüngliche Knoten sind, wohingegen bei einem Split nach A nur in einem Knoten deutliche Fortschritte erzählt werden (in dem aber verhältnismäßig wenige Elemente landen).

c) Wie würdest du den *Gini-Index* verwenden, um ein Merkmal unter mehreren Alternativen auszuwählen?

Individuelle Antworten

FACHLICHE ERGÄNZUNG: **gewichteter** *Gini-Index*

Gegen die Verwendung des Mittelwertes der Gini-Indizes spricht, dass hierbei die quantitative Aufteilung der Datenelemente nicht berücksichtigt wird. Dieses Argument kann den Schülerinnen und Schülern durch die Betrachtung eines noch deutlicheren Extrembeispiels verdeutlicht werden. Dies erreicht man, indem man die bei einem Split nach A vorliegende Situation noch weiter auf die Spitze treibt. Die folgende Abbildung zeigt einen derartigen Split:



Vergleicht man nun anstelle des Splits nach A den neuen Split nach C mit dem Split nach B, so fällt auf:

- Der Mittelwert der Gini-Indizes beträgt beim Split nach C ungefähr 0,25. Er fällt damit niedriger aus als bei einem Split nach B, wo der Mittelwert ungefähr 0,297 ist.
- Ein Split nach B erscheint allerdings deutlich sinnvoller als ein Split nach C, da bei letzterem nur ein einziges Datenelement abgespalten wird, wodurch sich die (Un-)Ordnung im anderen Knoten praktisch nicht verändert.

Die quantitative Aufteilung der Datenelemente kann man berücksichtigen, indem man die Gini-Indizes der beiden Kinderknoten, die bei einem Split entstehen, mit dem Anteil der Daten gewichtet, die in diesen Knoten laden. Formal kann man dies wie folgt darstellen:

Bei der binären Teilung einer Datenliste L mithilfe eines Merkmals X (mit den Ausprägungen 0 und 1) entstehen die zwei Teillisten $L_{X=0}$ und $L_{X=1}$. Als **gewichteten Gini-Index** des Splits von L nach X, $\mathrm{Gini}(L, X)$, bezeichnen wir den gewichteten Mittelwert der Gini-Indizes der beiden Teillisten von L:

$$\mathrm{Gini}(L, X) := \frac{|L_{X=0}|}{|L|} \cdot \mathrm{Gini}(L_{X=0}) + \frac{|L_{X=1}|}{|L|} \cdot \mathrm{Gini}(L_{X=1})$$

In obigem Beispiel ist also

$$\mathrm{Gini}(L, C) = \frac{|L_{C=0}|}{|L|} \cdot \mathrm{Gini}(L_{C=0}) + \frac{|L_{C=1}|}{|L|} \cdot \mathrm{Gini}(L_{C=1}) \approx \frac{1}{100} \cdot 0 + \frac{99}{100} \cdot 0{,}4999 \approx 0{,}4949.$$

Der gewichtete Gini-Index bei einem Split nach B ist hingegen

$$\mathrm{Gini}(L, B) = \frac{|L_{B=0}|}{|L|} \cdot \mathrm{Gini}(L_{B=0}) + \frac{|L_{B=1}|}{|L|} \cdot \mathrm{Gini}(L_{B=1}) = \frac{60}{100} \cdot 0{,}375 + \frac{40}{100} \cdot 0{,}21875 = 0{,}3125.$$

Damit ist der gewichtete Gini-Index des Splits nach B deutlich besser, als der des Splits nach C.

5. Induktive Baumerzeugung mithilfe des gewichteten Gini-Indexes

Der *gierige Algorithmus* zur Baumerzeugung lässt sich mithilfe des gewichteten Gini-Indexes präzisieren: Erzeuge den Entscheidungsbaum, indem du bei jedem Split ein Merkmal verwendest, das den gewichteten Gini-Index des Splits minimiert. Diese Vorschrift ist immer noch mehrdeutig, da in einer Situation mehrere Splits gleich gut sein können. Für die Implementierung muss festgelegt werden, wie dann verfahren wird. Zudem handelt es sich „nur“ um eine Heuristik, um eine Strategie der lokalen Optimierung, die global nicht das beste Ergebnis liefern muss.

Das Ziel der Bearbeitung von Arbeitsblatt 8 besteht zunächst darin, dass die Schülerinnen und Schüler das Verfahren verinnerlichen und durch wiederholte Anwendung des beschriebenen Kriteriums einen Entscheidungsbaum erzeugen. Fragen nach der Optimalität und der Implementierung eignen sich für eine vertiefte Behandlung zu einem späteren Zeitpunkt.

Lösungen zu Arbeitsblatt 8:

a) Zeige rechnerisch, dass Merkmal B für den ersten Test herangezogen werden sollte.

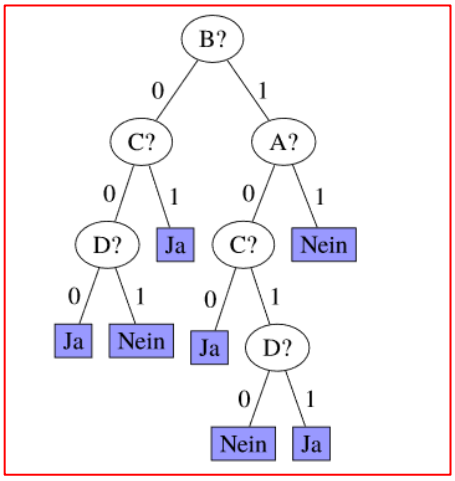
Wir berechnen für jedes der vier Merkmal den gewichteten Gini-Index:

- Gini(L, A) = $\frac{5}{12} \cdot 2 \cdot \frac{4}{5} \cdot \frac{1}{5} + \frac{7}{12} \cdot 2 \cdot \frac{3}{7} \cdot \frac{4}{7} = \frac{2}{15} + \frac{2}{7} = 0{,}419$
- Gini(L, B) = $\frac{1}{2} \cdot 2 \cdot \frac{5}{6} \cdot \frac{1}{6} + \frac{1}{2} \cdot 2 \cdot \frac{2}{6} \cdot \frac{4}{6} = \frac{5}{36} + \frac{8}{36} = 0{,}361$
- Gini(L, C) = $\frac{1}{2} \cdot 2 \cdot \frac{3}{6} \cdot \frac{3}{6} + \frac{1}{2} \cdot 2 \cdot \frac{4}{6} \cdot \frac{2}{6} = \frac{1}{4} + \frac{2}{9} = 0{,}472$
- Gini(L, D) = : $\frac{7}{12} \cdot 2 \cdot \frac{4}{7} \cdot \frac{3}{7} + \frac{5}{12} \cdot 2 \cdot \frac{3}{5} \cdot \frac{2}{5} = \frac{2}{7} + \frac{1}{5} = 0{,}486$

Der Split nach Merkmal B minimiert offenbar den gewichteten Gini-Index.

b) Setze das Verfahren fort und entwickele damit den vollständigen Entscheidungsbaum.

Lösung:



6. Implementierung

An die Bearbeitung der unplugged-Aufgaben schließt sich die Implementierung des rekursiven Baumerzeugungsalgorithmus und des Algorithmus zur Klassifizierung von Datenelementen an. Wir sind uns darüber im Klaren, dass hierbei hohe kognitive Anforderungen an die Schülerinnen und Schüler gestellt werden. Zur Entlastung stellen wir eine JAVA-Vorlage zur Verfügung, die nur an geeigneten Stellen ergänzt werden muss und von vielen technischen Details entlastet.

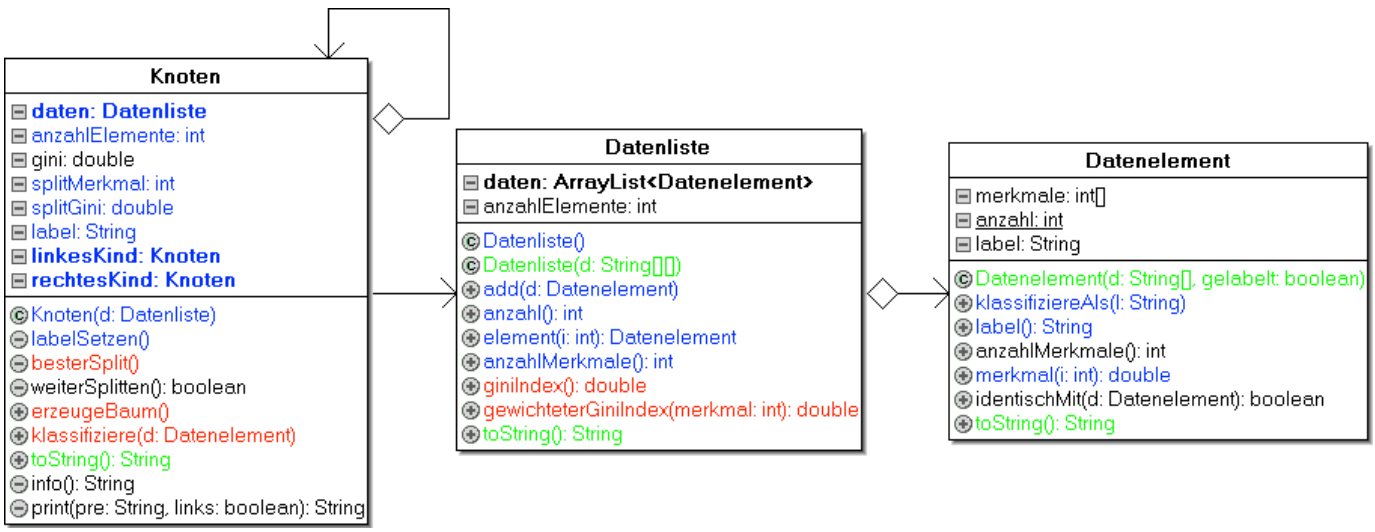
Es erscheint uns hilfreich, die zu implementierenden Algorithmen zunächst in Form von Pseudocode (o.ä.) mit den Schülerinnen und Schülern zu erarbeiten. Anschließend können die JAVA-Vorlage analysiert, besprochen und einzelne Bausteine des zuvor besprochenen Algorithmus identifiziert werden.

Pseudocode der beiden zentralen Algorithmen:

```
erzeugeBaum(Datenliste)
  Wenn alle Elemente der Datenliste der gleichen Klasse angehören,
    dann
          erzeuge ein Blatt und beschrifte es mit der Klasse,
    sonst
          wähle ein Merkmal, welches die Datenelemente auf zwei nichtleere
             Teillisten mit minimalem gewichtetem Gini-Index aufteilt,
          erstelle einen Knoten und beschrifte ihn mit diesem Merkmal,
          teile die Datenliste anhand des Merkmales in zwei Teillisten und
          erstelle zwei Kinderknoten:
             linkes Kind  := erzeugeBaum(Teilliste_1)
             rechtes Kind := erzeugeBaum(Teilliste_2).


klassifiziere(Datenelement)
  Wenn der aktuelle Knoten ein Blatt ist,
    dann
          markiere das Datenelement mit der Beschriftung des Knotens
    sonst
          wenn das Datenelement bzgl. des Merkmals, mit dem der Knoten be-
               schriftet ist, in die linke Teilliste gehört,
            dann
                  klassifiziere das Datenelement mit dem linken Kindknoten
            sonst
                  klassifiziere das Datenelement mit dem rechten Kindknoten
```

In der JAVA-Vorlage sind mehrere Klassen vorbereitet, die die Arbeit erleichtern sollen. Sie sind in folgendem UML-Diagramm dargestellt. Ausführliche Erklärungen und Details zur Implementierung folgen weiter unten. Die vollständige Implementierung sowie einige Testbeispiele finden Sie im beiliegenden Quellcode.



Begleitskript zum Unterrichtsmodul: „Lernende Algorithmen“

Die Schülerinnen und Schüler sollen die rot markierten Methoden *giniIndex* und *gewichteterGiniIndex* der Klasse *Datenelement* sowie die Methoden *besterSplit*, *erzeugeBaum* und *klassifiziere* der Klasse *Knoten* implementieren. Für diese Implementierungen benötigen sie die blau markierten Attribute und Methoden, deren Sinn und Verwendung daher im Vorfeld besprochen werden sollte. Die grün markierten Methoden werden zum Testen der Implementierung benötigt. Die restlichen Methoden dienen der formatierten Ausgabe der Bäume und für mögliche Erweiterungen, die bisher nicht thematisiert wurden.

Klasse `Datenelement`

Die Klasse dient der Repräsentation der verwendeten Datenelemente und kann den Schülerinnen und Schülern komplett vorgegeben werden. Auf einige Besonderheiten möchten wir an dieser Stelle erläuternd eingehen:

- Die Beschränkung auf **ganzzahlige Attributwerte** erleichtert die weitere Implementierung. Da im unplugged-Teil sogar ausschließlich binäre Attribute verwendet wurden, verwenden wir in der Implementierung (zunächst) nur die Zahlen 0 und 1, um die beiden Ausprägungen eines Merkmals darzustellen. Soll die Implementierung auf kontinuierliche Merkmale (z.B. Länge, …) erweitert werden, bietet es sich an, diese als double statt als int zu speichern.
- Die **Klassifizierung** wird in einem String gespeichert. Für die bisher verwendeten binären Klassifizierungen verwenden wir nur die Labels „ja“ und „nein“, eine Erweiterung auf nicht binäre Klassifizierungen ist aber möglich.
- Die **Erzeugung von Datenelementen** erfolgt mithilfe eines String-Arrays, in dem die einzelnen Merkmalsausprägungen stehen, evtl. gefolgt von einem Label, das die Klassenzugehörigkeit angibt. Der boolsche Parameter des Konstruktors gibt an, ob es sich um gelabelte Daten handelt, also ob der letzte Eintrag ein Merkmal oder ein Klassifikationslabel ist.
- Durch **Abfrage-Methoden** können die einzelnen Merkmalsausprägungen, die Anzahl der Merkmale sowie das Klassifikationslabel abgefragt und Datenelemente verglichen werden. Zwei Datenelemente gelten dabei als gleich, wenn sie in ihren Merkmalsausprägungen übereinstimmen, unabhängig von ihrer Klassifizierung. Mit der Methode *klassifiziereAls* kann einem Datenelement ein Klassifikationslabel zugewiesen werden.

Klasse `Datenliste`

Die Klasse dient der Repräsentation von Datenlisten, wie sie als Eingabe zur Erstellung von Entscheidungsbäumen verwendet werden.

- Mittels der beiden **Konstruktoren** können wahlweise eine leere Datenliste erstellt oder über ein zweidimensionales String-array bereits Datenelemente eingefügt werden. Weitere Datenelemente können mit der Methode *add* hinzugefügt werden.
- Für den Zugriff auf die einzelnen Datenelemente sowie die Anzahl der Elemente und Merkmale stehen **Abfrage-Methoden** zur Verfügung.
- Die Methoden *giniIndex* und *gewichterGiniIndex* berechnen die Indizes wie oben beschrieben. Sie sollen von den Schülerinnen und Schülern implementiert werden. Die Aufteilung der Datenliste in Teillisten zur Berechnung des gewichteten Gini-Indexes ist bei nicht-binären Merkmalen schwieriger als bei binären. Soll die Implementierung auf solche Merkmale erweitert werden, muss der Methode ein weiterer Parameter (*Schwellwert*) hinzugefügt werden, der angibt, wie die Datenelemente aufzuteilen sind.

Klasse `Knoten`

Der Entscheidungsbaum wird aus Instanzen der Klasse *Knoten* aufgebaut.

- **Attribute** der Klasse: In jedem Knoten werden in einer *Datenliste* die Datenelemente gespeichert, die diesen Knoten bei einer Klassifizierung passieren (würden). Zudem werden die Anzahl der Datenelemente und der Gini-Index der Datenliste gespeichert. Letzterer wird direkt im Konstruktor berechnet. Im Attribut *splitMerkmal* wird während des Aufbaus des Baumes das Merkmal gespeichert, nach dem die Datenliste auf die beiden Kindknoten aufgeteilt wird. Diese Information wird beim Klassifizieren neuer Datenelemente benötigt. Im Attribut *splitGini* wird der gewichtete Gini-Index dieses Splittes gespeichert um ihn bei der Ausgabe des Baumes anzeigen zu können.
- In der Methode *erzeugeBaum* sollen die Schülerinnen und Schüler den oben beschriebenen, rekursiven Algorithmus zur Erzeugung des Entscheidungsbaumes implementieren. Hierbei wird die Methode *weiterSplitten* benötigt, die letztlich die Rekursion beendet, indem sie überprüft, ob weitere Splits sinnvoll sind. Hierfür reicht es im einfachen Fall (keine widersprüchlichen Daten) zu überprüfen, ob alle Datenelemente der Liste der gleichen Klasse angehören (Gini-Index = 0). Soll der Algorithmus auch bei widersprüchlichen Daten funktionieren, so muss die Bedingung der if-Anweisung angepasst werden, da ein Knoten in diesem Fall nur dann weiter gesplittet werden darf, wenn zusätzlich

nicht alle Datenelemente gleich sind (abgesehen vom Label), also nicht in allen Merkmalsausprägungen übereinstimmen.
Soll ein Knoten nicht weiter gesplittet werden, also zu einem Blatt werden, wird er mit der Klasse seiner Datenelemente gelabelt. Bei widersprüchlichen Daten erfolgt hier ein Mehrheitsentscheid. Dafür kann die Methode *labelSetzen* verwendet werden. Andernfalls muss das beste Attribut für einen Split ermittelt werden, was die Methode *besterSplit* macht. Anschließend werden gemäß dieses Splits zwei Teillisten erzeugt und aus ihnen rekursiv zwei weitere (Teil-)Bäume erstellt.

- Die Methode *besterSplit* berechnet, bzgl. welchen Merkmals die Datenliste des Knotens am besten aufgeteilt wird. In der bisher besprochenen Variante muss dazu über alle Merkmale iteriert und jeweils der gewichtete Gini-Index berechnet werden. Der beste (kleinste) gewichtete Gini-Index sowie das zugehörige Merkmal werden im Knoten gespeichert. Soll der Algorithmus auf nicht-binäre Attribute erweitert werden, muss in dieser Methode zusätzlich über alle möglichen Schwellwerte jedes Merkmals iteriert werden und dieser Schwellwert für den besten Split im Knoten gespeichert werden.
- Mit der rekursiven Methode *klassifiziere* kann ein Datenelement unter Verwendung eines erzeugten Entscheidungsbaums klassifiziert werden: Ist der Knoten ein Blatt, so wird dem Datenelement das im Knoten gespeicherte Label zugewiesen. Andernfalls wird das Datenelement abhängig von der Ausprägung des Merkmals, das im Knoten zum-Splitten verwendet wird, mithilfe des linken oder rechten Teilbaums rekursiv klassifiziert. Bei nicht binären Merkmalen ist hier zusätzlich der im Knoten gespeicherte Schwellwert zu berücksichtigen.

In der vollständigen Implementierung, die Sie im beigefügten Material finden, verfügt die Klasse *Knoten* zusätzlich über eine Methode *erzeugeBaumMitReihenfolge*. Sie ist nützlich, um bestimmte Bäume unabhängig vom Gini-Index erzeugen und anzeigen zu lassen. Im ersten Parameter werden hier die zum Splitten verwendeten Merkmale in einem integer-Array in Level-Order angegeben, der zweite Parameter ist bei jedem Aufruf 1. Er wird nur für die Rekursion benötigt.

Beispiele für die Verwendung der Klassen finden Sie im beigefügten Material.