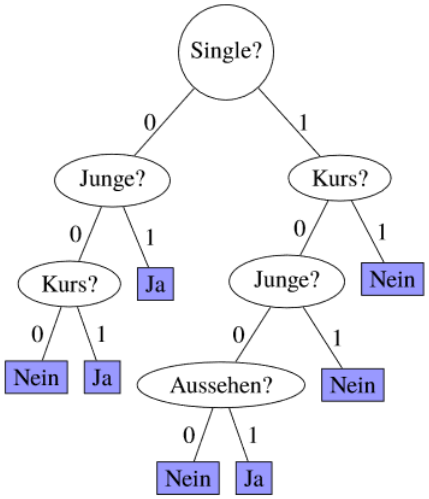
Unterrichtsmodul Lernende Algorithmen

Pascal Schmidt
Stefan Strobel
Prof. Dr. Verena Wolf



UNIVERSITÄT DES SAARLANDES
Ministerium für Bildung und Kultur
Landesinstitut für Pädagogik und Medien
SaLUt Saarland
GEFÖRDERT VOM
Bundesministerium für Bildung und Forschung

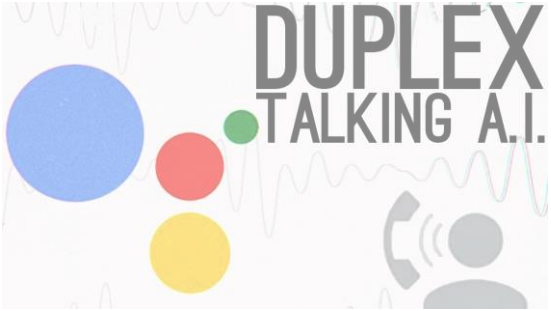
Ablauf des Workshops

- Motivation (Wolf)
- Einführung in Entscheidungsbäume (Schmidt)
- Bearbeitung verschiedener Aufgabenstellungen (unplugged/plugged)
- Feedback & Diskussion (Schmidt/Wolf)

Motivation

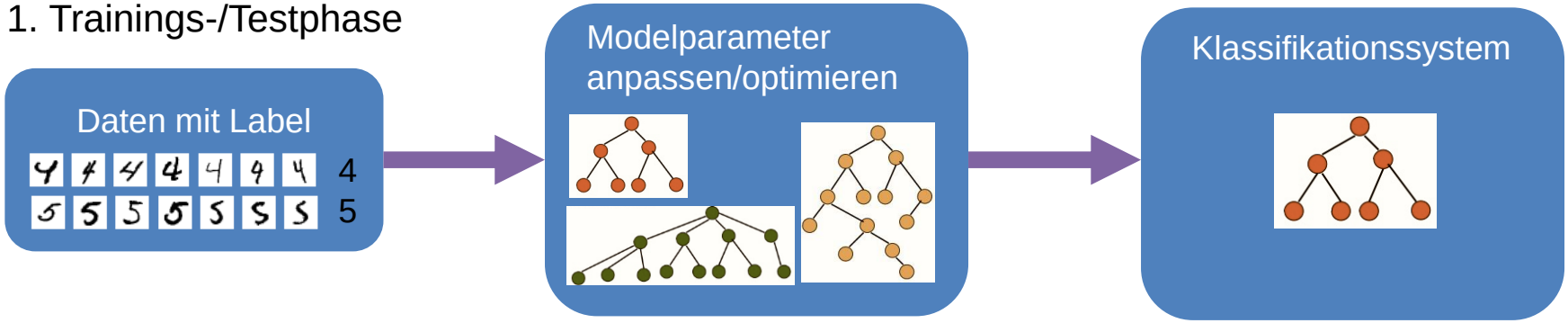
Zunehmender Einsatz von **Lern-Systemen**:

- Sprach-/Texterkennung und -übersetzung
- Bild- /Videoverarbeitung (z.B. Gesichts-/ Objekterkennung/-tracking, etc.)
- Empfehlungssysteme
- Medizin/Bioinformatik
- Planungssysteme/Robotik
- bald: selbstfahrende Autos
- Landwirtschaft, Maschinenbau, Verwaltung, Kundenservice, …

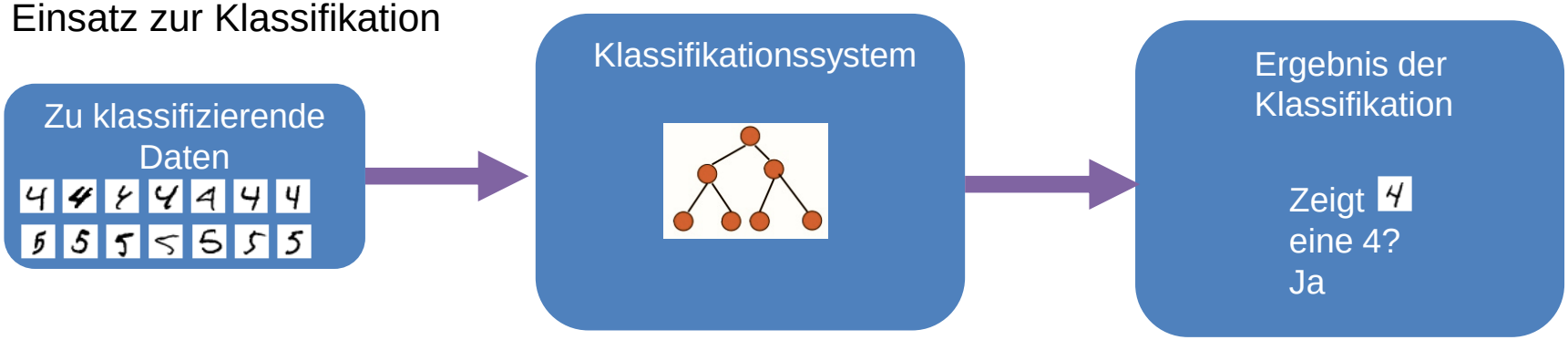


Funktionsweise eines Lernsystems

1. Trainings-/Testphase



2. Einsatz zur Klassifikation

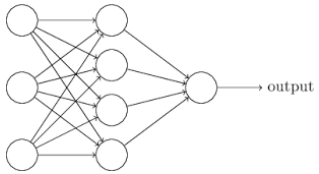


Problemlösen mit und ohne Lernen

Ohne Lernen

- Folge von Handlungsvorschriften
- Strategie zur Lösung vom Entwickler vorgegeben
- Bsp.: Such-/Sortieralgorithmen oder „selbst erdachtes Regelwerk“

Mit Lernen:



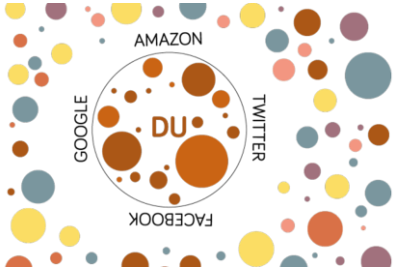
- „maschinelles Lernen“
- Generisches Modell wird mit Hilfe von Beispiellösungen angepasst
- Strategie wird „erlernt“

-> Wird weder in den Empfehlungen der GI noch in nationalen Lehrplänen berücksichtigt.

Warum Lernsysteme im Schulunterricht?

Ziel: **Entmystifizierung** der digitalen Welt;
Verstehen von **Chancen & Risiken** der KI

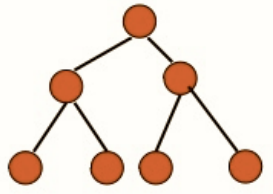
- Implikationen von Empfehlungssystemen: filter bubbles, fake news, fake videos und dark ads, ...
- Ethische Fragen (z.B. im Zusammenhang mit autonomem Fahren) diskutieren
- Algorithmic bias verstehen (z.B. Bewerbungen, Kredite, …)
- Umgang mit persönlichen Daten
- viel Zeit für "klassische" Art des Problemlösens (imperative Programmierung!), aber keine Lernsysteme im Unterricht



https://blog.bundjugend.de


https://www.propublica.org

Das Unterrichtsmodul

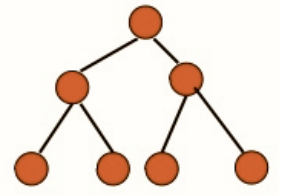


- Bisher "nur": Unterrichtsmodul zur grundlegenden Funktionsweise anhand einer Modellklasse (Entscheidungsbäume)
- unerprobt (!) → Feedback erwünscht
- (noch) keine generelle Diskussion von KI(-Implikationen) oder Teilbereich "Lernen"

E-Bäume als beispielhaftes Lernverfahren

Vorteile:

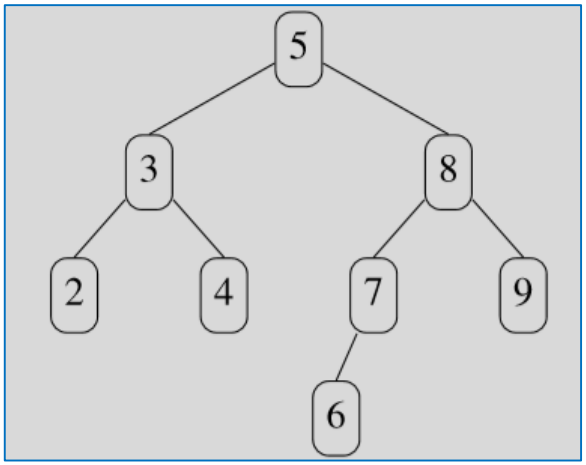
- nur geringe mathematische Vorkenntnisse nötig
- **einfache Funktionsweise**: Klassifikation durch Traversieren des Baumes (white box model)



> „Other classifiers **do not offer explanations** of this kind. Especially the *neural network* is a real *black box*."
>
> Kubat M.: *An Introduction to Machine Learning*, Springer (2017)

- rekursive Modellerstellung: Merkmal wählen, Split durchführen
- Optimierung durch einfache Heuristik
- Nähe zu Oberstufen-Themen wie binäre Suchbäume und divide&conquer-Prinzip

Geordnete binäre Bäume



Ein Binärbaum heißt **binärer Suchbaum**, wenn für jeden Knoten *k* gilt, dass

- alle Schlüssel im linken Teilbaum von *k* kleiner als der Schlüssel von *k* sind und
- alle Schlüssel im rechten Teilbaum von *k* größer (oder gleich) als der Schlüssel von *k* sind.

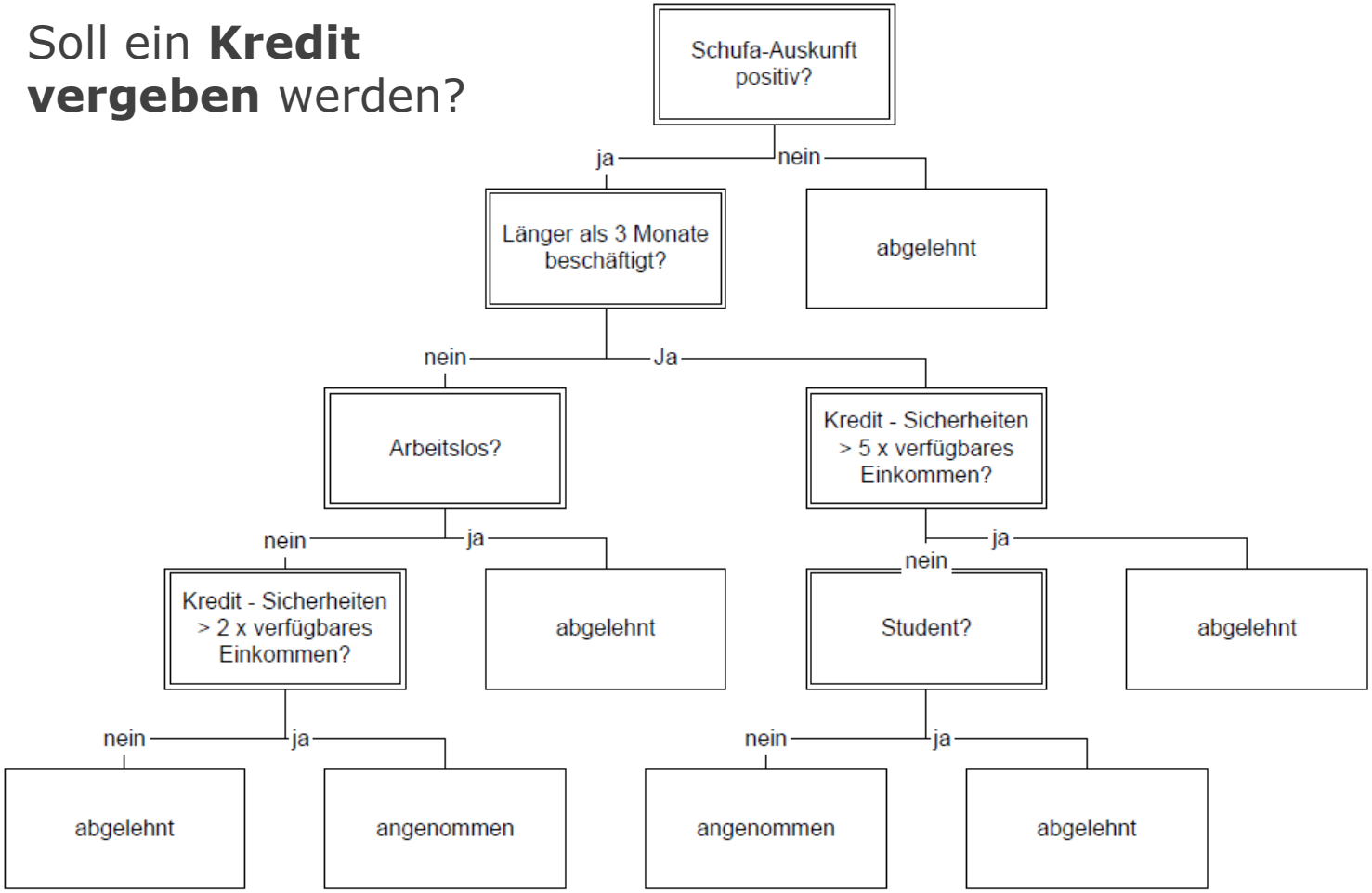
Ernst et al.: „*Grundkurs Informatik*“, Springer 2016

Aus dem Lehrplan (GOS, 4-stündig):

Geordnete binäre Bäume
Speicherung von Schlüsseln in einem binären Baum
Die elementaren Operationen Suchen, Einfügen und Entfernen am graphischen Modell

Was sind Entscheidungsbäume?

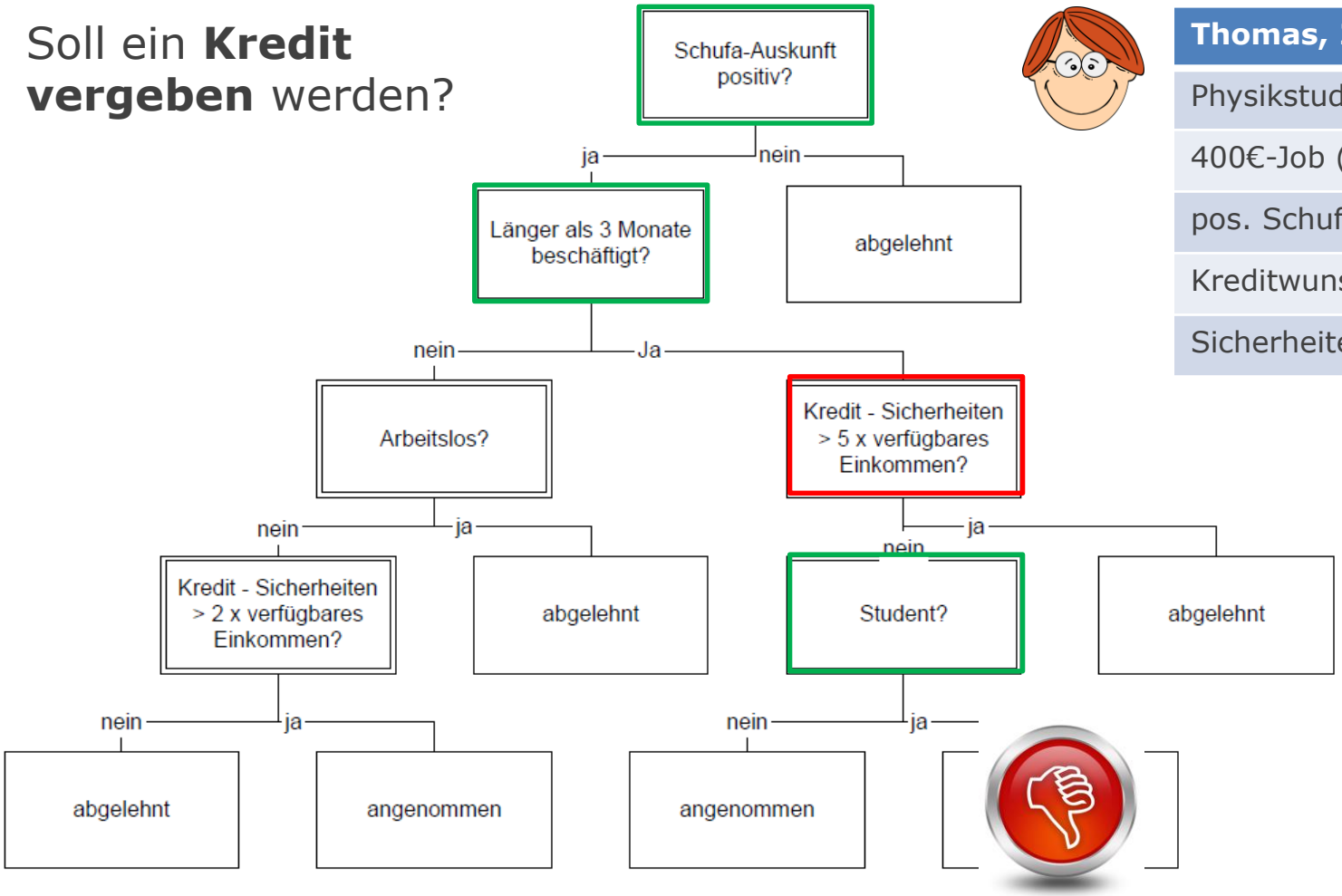
Soll ein **Kredit vergeben** werden?



Sawade et al: „*Entscheidungsbäume*“, Lehrstuhl Maschinelles Lernen, Universität Potsdam

Was sind Entscheidungsbäume?

Soll ein **Kredit vergeben** werden?



Thomas, 23
Physikstudent
400€-Job (seit 1 Jahr)
pos. Schufa-Auskunft
Kreditwunsch: 2500€
Sicherheiten: 1000€

Sawade et al: „*Entscheidungsbäume*“, Lehrstuhl Maschinelles Lernen, Universität Potsdam

Induktion von Entscheidungsbäumen

Wie entsteht aus Daten Wissen?

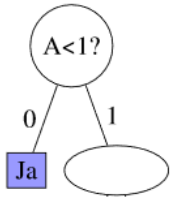
item	A	B	C	label

Vereinfachungen:

- Nur binäre Label und Merkmale
- Zunächst weder Duplikate noch Widersprüche

Induktion von Entscheidungsbäumen

Wie entsteht aus Daten Wissen?



item	A	B	C	label
1	0	0	0	Ja
2	0	0	1	Nein
3	0	1	0	Ja
4	0	1	1	Nein
5	1	0	0	Ja
6	1	0	1	Ja
7	1	1	0	Ja
8	1	1	1	Ja

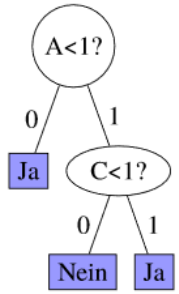
Vereinfachungen:

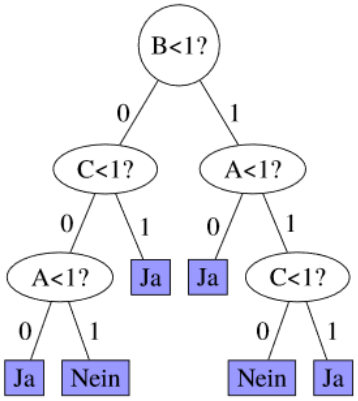
- Nur binäre Label und Merkmale
- Zunächst weder Duplikate noch Widersprüche

Induktion von Entscheidungsbäumen

Wie entsteht aus Daten Wissen?

item	A	B	C	label
1	0	0	0	Ja
2	0	0	1	Nein
3	0	1	0	Ja
4	0	1	1	Nein
5	1	0	0	Ja
6	1	0	1	Ja
7	1	1	0	Ja
8	1	1	1	Ja





Vereinfachungen:

- Nur binäre Label und Merkmale
- Zunächst weder Duplikate noch Widersprüche

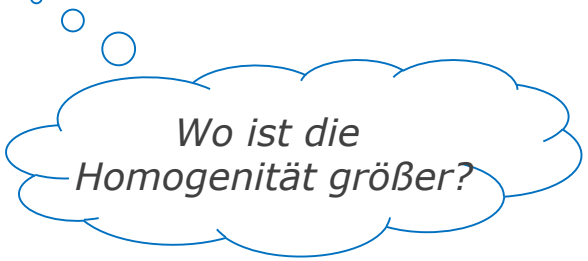
Induktion von Entscheidungsbäumen

Greedy-Regel:
Wähle bei jedem Split dasjenige Attribut, welches möglichst „reine“ („homogene“) Teillisten erzeugt.

item	A	B	C	label
1	0	0	0	Ja
2	0	0	1	Nein
3	0	1	0	Ja
4	0	1	1	Nein
5	1	0	0	Ja
6	1	0	1	Ja
7	1	1	0	Ja
8	1	1	1	Ja

Beispiele:

- Initialer Test nach B: [i1, i2, i5, i6], [i3, i4, i7, i8]
- Initialer Test nach A: [i1, i2, i3, i4], [i5, i6, i7, i8]



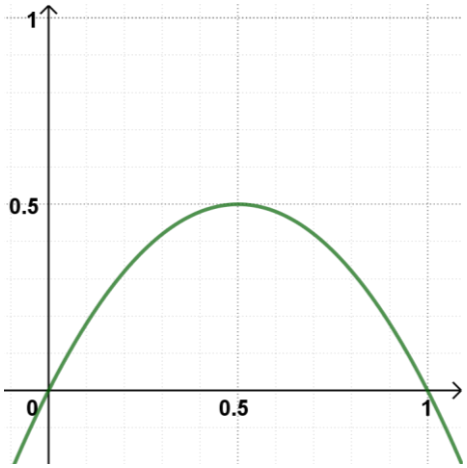
Gini-Index einer Datenliste

- **Berechnungsformel** (für 2 Label):
  Seien p_0 und p_1 die Anteile der Daten mit Label „ja“ bzw. „nein“:

  $$\mathbf{gini(L) := p_0 \cdot (1-p_0) + p_1 \cdot (1-p_1) = p_0 \cdot p_1 + p_1 \cdot p_0 = 2 \cdot p_0 \cdot p_1}$$

- **Inhaltliche Deutung**
  Wahrscheinlichkeit, mit der ein zufällig ausgewähltes Datenelement falsch klassifiziert wird, wenn die Klassifikation zufällig (basierend auf der Verteilung der Label) vorgenommen wird.

- **Interpretation**:
  $gini(L) = 2 \cdot p_0 \cdot p_1$
  $= 2 \cdot p_0 \cdot (1-p_0)$
  $= 2 \cdot p_0 - 2\, p_0^2$



Gini-Index einer Datenliste

- **Beispiel:**

item	A	B	C	label
1	0	0	0	Ja
2	0	0	1	Nein
3	0	1	0	Ja
4	0	1	1	Nein
5	1	0	0	Ja
6	1	0	1	Ja
7	1	1	0	Ja
8	1	1	1	Ja

$gini(L) = 2 \cdot p_0 \cdot p_1$

$p_0 = 2/8 \quad p_1 = 6/8$

$gini(L) = 2 \cdot 2/8 \cdot 6/8 = 3/8 = 0{,}375$

Gini-Index eines Splits

- Bei einem Split nach Merkmal X entstehen zwei Teillisten, X_0 und X_1.
- **Definiere:**
  Gini-Index des Splits nach X als gewichteter Mittelwert der Gini-Indizes von X_0 und X_1.

item	A	B	C	label
1	0	0	0	Ja
2	0	0	1	Nein
3	0	1	0	Ja
4	0	1	1	Nein
5	1	0	0	Ja
6	1	0	1	Ja
7	1	1	0	Ja
8	1	1	1	Ja

Gini-Index eines Splits

- Bei einem Split nach Merkmal X entstehen zwei Teillisten, X_0 und X_1.
- **Definiere:**
  Gini-Index des Splits nach X als gewichteter Mittelwert der Gini-Indizes von X_0 und X_1.

item	A	B	C	label
1	0	0	0	Ja
2	0	0	1	Nein
3	0	1	0	Ja
4	0	1	1	Nein
5	1	0	0	Ja
6	1	0	1	Ja
7	1	1	0	Ja
8	1	1	1	Ja

- **Beispiel**:

$$splitGini(A) = 4/8 \cdot gini([i1, i2, i3, i4]) + 4/8 \cdot gini([i5, i6, i7, i8])$$
$$= 4/8 \cdot 0{,}5 + 4/8 \cdot 0$$
$$= 0{,}25$$

$$splitGini(B) = 4/8 \cdot gini([i1, i2, i5, i6]) + 4/8 \cdot gini([i3, i4, i7, i8])$$
$$= 0{,}375$$

Algorithmus in Pseudocode:

```
erzeugeBaum(Datenliste)
    wenn alle Elemente der Datenliste das gleiche Label haben
        dann
            erzeuge ein Blatt und beschrifte es mit dem Label
        sonst
            Wähle ein Merkmal zum Testen, welches zwei nichtleere
            Kinderknoten mit einem minimalen gewichteten Gini-
            Index erzeugt.
            Erzeuge einen Knoten und beschrifte ihn mit dem Test
            Teile die Datenliste anhand des Testes in zwei
            Teillisten
            linkes Kind := erzeugeBaum(Teilliste1)
            rechtes Kind := erzeugeBaum(Teilliste2)
```