# Big Data Anwendungen

Stream Mining

# Agenda

- Einführung
- Deskriptive Methoden zur Datenexploration
- Datenqualität
- Klassifikation
- Recommender Systems
- Clusteringverfahren

- Stream Mining
  - Einordnung
  - H-Tree
  - CDH-Tree

- Social Network Analysis
- Technische Lösungen
- Datenschutz und gesellschaftliche Aspekte

# Data Streams

- Daten werden kontinuierlich generiert
  - Kaufdaten in Supermärkten
  - Daten von GPS Systemen
  - Änderungen von Preisen
  - Logdaten von Telefonanrufen / Webseiten
  - Postings in sozialen Medien
  - Daten von Sensornetzwerken

- Besonderheiten
  - Datenvolumen ist größer als bei klassischen Ansätzen
  - Daten ändern sich im zeitlichen Verlauf
    (z.B. Temperaturen im Jahresverlauf, Änderung Konsumverhalten)
  - Explizites Modell oft uninteressant...
    (z.B. Vorhersage Anzahl Telefonate, künftige Position, ...)
  - ... dafür aktuell gültiges Modell

- Hier
  - Entscheidungsbaum lernen für Data Streams

# Implikationen

- Eigenschaften des Entscheidungsbaums
  - Beobachtungen
    - Analyse aller Beobachtungen
    - Anpassung des Entscheidungsbaums nach jeder Beobachtung
    - Kein Speichern der Beobachtungen
  - Echtzeit
    - Permanente Anpassung des Entscheidungsbaums
    - Hohe Performanzanforderung
  - Anwendung
    - Entscheidungsbaum wird parallel genutzt und trainiert
    - Bewertung der Güte muss permanent erfolgen

- Zwei Datengenerierende Prozesse
  - Stationäre Daten
    (Zeit hat keinen Einfluss auf Vorhersage, H-Tree)
  - Zeitabhängige Daten
    (Saisonale Einflüsse usw. ändern die Vorhersage, CDH-Tree)

# Agenda

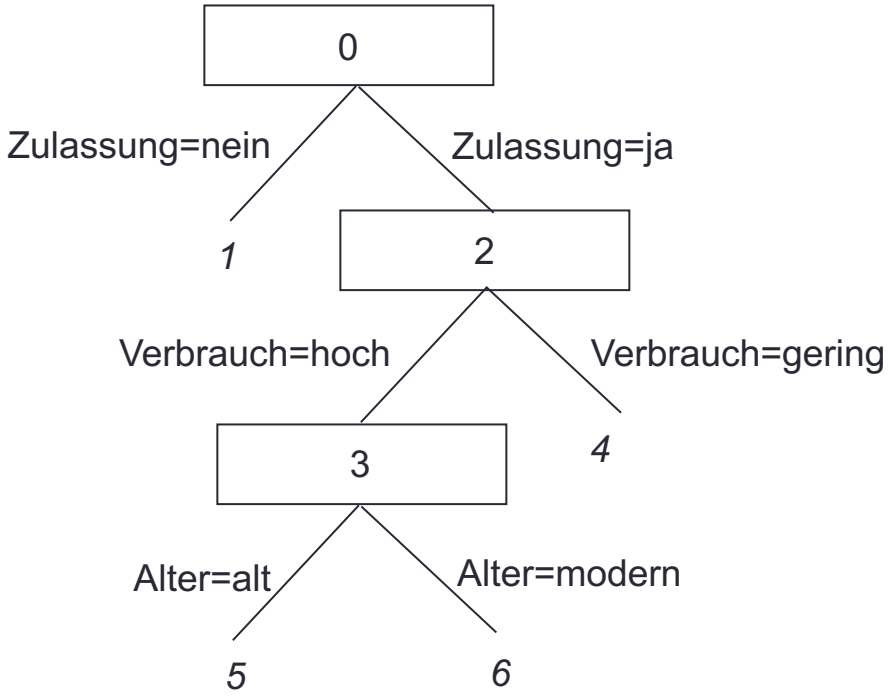
- Einführung
- Deskriptive Methoden zur Datenexploration
- Datenqualität
- Klassifikation
- Recommender Systems
- Clusteringverfahren

- Stream Mining
  - Einordnung
  - H-Tree
  - CDH-Tree

- Social Network Analysis
- Technische Lösungen
- Datenschutz und gesellschaftliche Aspekte

# Grundprinzip

- Aufbau
  - Knoten entstehen nacheinander
    ... und Entstehung wird nummeriert
  - Split immer dann, wenn Splitkriterium erreicht
  - Splits sind n-är
  - Entscheidungen über „Arrays“
- Beispiel
  - Baum ist entstanden durch
    - Split in 0 ⇒ Knoten 1 und 2
    - Split in 2 ⇒ Knoten 3 und 4
    - Split in 3 ⇒ Knoten 5 und 6
  - Nummer des nächsten Knoten: 7
- Unterschiede zum Entscheidungsknoten
  - Knoten und Blätter haben ID
  - Nächster ID wird gemerkt



# Arrays

- Idee
  - Speicherung von Arrays an jedem Knoten
  - Arrays sollen Änderungen am Baum erleichtern (enthalten alle Daten für Split und Anwendung)
- Alle Knoten
  - `VerfügbareAttribute`
    Attribute die nicht in Vorgängern für Split genutzt wurde
  - `AnzahlProKlasse`
    Anzahl der Beobachtungen für jede einzelne Klasse
- Innere Knoten
  - `SplitAttribut`
    Gibt genutztes Splitattribut an
- Blattknoten
  - `AnzahlTreffer`
    Zählt die Anzahl der Beobachtungen seit Erstellung
  - `AnzahlProAttributKlasse`
    Anzahl der Beobachtungen für jede einzelne Klasse, Wert Kombination

# Baumerstellung am Beispiel I

- Wurzelknoten (Initialisierung)
  - `VerfügbareAttribute {Zulassung, Verbrauch, Alter}`
  - `SplitAttribut {}`
  - `AnzahlTreffer 0`

0

  - `AnzahlProAttributKlasse`

| | ja | nein | hoch | gering | modern | alt |
|---|---|---|---|---|---|---|
| Kauf | | | | | | |
| Kein Kauf | | | | | | |

  - `AnzahlProKlasse {Kauf: 0, kein Kauf: 0}`

Data Stream
(Zulassung, Verbrauch, Alter, Kauf)

1 (nein, hoch, modern, nein)
2 (nein, gering, modern, nein)
3 (nein, hoch, alt, nein)
4 (nein, hoch, modern, nein)
5 (ja, gering, alt, ja)
6 (ja, gering, modern, ja)
7 (ja, hoch, alt, nein)
8 (ja, hoch, modern, ja)

# Baumerstellung am Beispiel II

- Wurzelknoten (nach 4. Beobachtung - Split)
  - `VerfügbareAttribute {Zulassung, Verbrauch, Alter}`
  - `SplitAttribut {}`
  - `AnzahlTreffer 4`
  - `AnzahlProAttributKlasse`

AnzahlTreffer als Splitkriterium: 4

0

Data Stream
(Zulassung, Verbrauch, Alter, Kauf)

1 (nein, hoch, modern, nein)
2 (nein, gering, modern, nein)
3 (nein, hoch, alt, nein)
4 (nein, hoch, modern, nein)
5 (ja, gering, alt, ja)
6 (ja, gering, modern, ja)
7 (ja, hoch, alt, nein)
8 (ja, hoch, modern, ja)

| | ja | nein | hoch | gering | modern | alt | Summe |
|---|---|---|---|---|---|---|---|
| Kauf | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Kein Kauf | 0 | 4 | 3 | 1 | 3 | 1 | 4 |
| Summe | 0 | 4 | 3 | 1 | 3 | 1 | |

- `AnzahlProKlasse {Kauf: 0, Kein Kauf: 4}`

Kein Split: Da alle Beobachtungen in einer Klasse

# Baumerstellung am Beispiel III

- Wurzelknoten (nach 8. Beobachtung - Split)
  - `VerfügbareAttribute {Zulassung, Verbrauch, Alter}`
  - `SplitAttribut {}`
  - `AnzahlTreffer 8`
  - `AnzahlProAttributKlasse`

AnzahlTreffer als Splitkriterium – jetzt 2x4

0

Data Stream
(Zulassung, Verbrauch, Alter, Kauf)

1 (nein, hoch, modern, nein)
2 (nein, gering, modern, nein)
3 (nein, hoch, alt, nein)
4 (nein, hoch, modern, nein)
5 (ja, gering, alt, ja)
6 (ja, gering, modern, ja)
7 (ja, hoch, alt, nein)
8 (ja, hoch, modern, ja)

| | ja | nein | hoch | gering | modern | alt | Summe |
|---|---|---|---|---|---|---|---|
| Kauf | 3 | 0 | 1 | 2 | 2 | 1 | 3 |
| Kein Kauf | 1 | 4 | 4 | 1 | 3 | 2 | 5 |
| Summe | 4 | 4 | 5 | 3 | 5 | 3 | |

- `AnzahlProKlasse {Kauf: 3, Kein Kauf: 5}`

Split: Beobachtungen in beiden Klassen

# Einschub – Entropie mit absoluten Häufigkeiten

- Bisher: $\mathrm{E} = -\sum_{i=1}^{|C|}(p_i \cdot \log_2 p_i)$

- Wünschenswert: Zwischenschritt zur Berechnung der $p_i$ überspringen, da …
  … nur Anzahl der Beobachtungen pro Klasse bekannt!

- Mögliche Umformung

$$\begin{aligned}
\mathrm{E} &= -\sum_{i=1}^{|C|}(p_i \cdot \log_2 p_i) = -\sum_{i=1}^{|C|}\left(\frac{n_\mathrm{i}}{\mathrm{n}} \cdot \log_2 \frac{n_i}{n}\right) \\
&= -\sum_{i=1}^{|C|}\left(\frac{n_\mathrm{i}}{\mathrm{n}} \cdot (\log_2 n_i - \log_2 n)\right) \\
&= -\sum_{i=1}^{|C|}\left(\frac{n_\mathrm{i}}{\mathrm{n}} \log_2 n_i\right) + \sum_{i=1}^{|C|}\left(\frac{n_\mathrm{i}}{\mathrm{n}} \log_2 n\right) \\
&= -\frac{1}{n}\sum_{i=1}^{|C|}(n_i \log_2 n_i) + \frac{1}{n}\log_2 n \sum_{i=1}^{|C|} n_i \\
&= \frac{1}{n}\left(-\sum_{i=1}^{|C|}(n_i \log_2 n_i) + n \log_2 n\right)
\end{aligned}$$

- Somit
  Berechnung Entropie direkt aus Tabelle `AnzahlProAttributKlasse`

# Baumerstellung am Beispiel IV

- Entropieberechnung:
  - 1. Tabelle (Zulassung)
    - $(-3 \cdot \log_2 3$
      $-1 \cdot \log_2 1 - 4 \cdot \log_2 4$
      $+4 \cdot \log_2 4 + 4 \cdot \log_2 4)/8$
    - $= 0.4056$
  - 2. Tabelle (Verbrauch)
    - $(-1 \cdot \log_2 1 - 2 \cdot \log_2 2$
      $-4 \cdot \log_2 4 - 1 \cdot \log_2 1$
      $+5 \cdot \log_2 5 + 3 \cdot \log_2 3)/8$
    - $= 0.7956$
  - 3. Tabelle (Alter)
    - $-2 \cdot \log_2 2 \; -1 \cdot \log_2 1$
      $-3 \cdot \log_2 3 - 2 \cdot \log_2 2$
      $+5 \cdot \log_2 5 + 3 \cdot \log_2 3)/8$
    - $= 0.9512$
- Entropie nach Split über Zulassung...
  ... am geringsten: Split über Zulassung

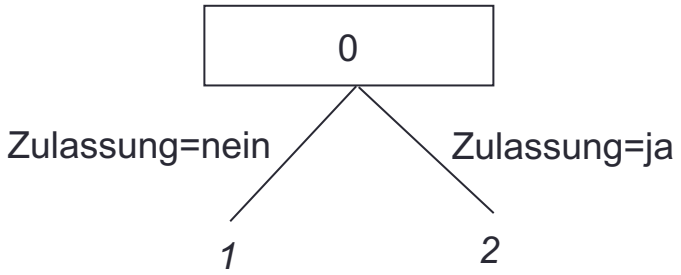
Wird ignoriert

| | ja | nein | Summe |
|---|---|---|---|
| Kauf | 3 | 0 | 3 |
| Kein Kauf | 1 | 4 | 5 |
| Summe | 4 | 4 | |

| | hoch | gering | Summe |
|---|---|---|---|
| Kauf | 1 | 2 | 3 |
| Kein Kauf | 4 | 1 | 5 |
| Summe | 5 | 3 | |

| | modern | alt | Summe |
|---|---|---|---|
| Kauf | 2 | 1 | 3 |
| Kein Kauf | 3 | 2 | 5 |
| Summe | 5 | 3 | |

# Baumerstellung am Beispiel V



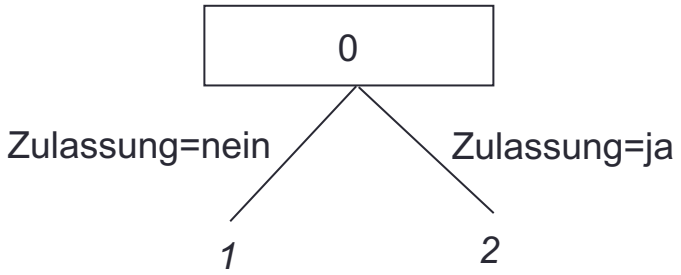

- Wurzelknoten (nach Split)
  - `VerfügbareAttribute {Zulassung, Verbrauch, Alter}`
  - `SplitAttribut {Zulassung}`
  - `AnzahlProKlasse: {Kauf: 3, Kein Kauf: 5}`
- Blattknoten 1 und 2 (nach Split)
  - `VerfügbareAttribute {Verbrauch, Alter}`
  - `AnzahlTreffer: 0`
  - `AnzahlProAttributKlasse`

| | hoch | gering | modern | alt | Summe |
|---|---|---|---|---|---|
| Kauf | 0 | 0 | 0 | 0 | 0 |
| Kein Kauf | 0 | 0 | 0 | 0 | 0 |
| Summe | 0 | 0 | 0 | 0 | |

  - `AnzahlProKlasse: {Kauf: 0, Kein Kauf: 0}`

# Baumerstellung am Beispiel VI

- Stellen wir uns vor: Baum wurde weiter gelernt...
  ... und eine neue Beobachtung...
  ... soll klassifiziert werden mit (Zulassung=ja)




- Finaler Baum
  - Wurzelknoten
    `AnzahlProKlasse: {Kauf: 3, Kein Kauf: 5}`
  - Blattknoten 2
    `AnzahlProKlasse: {Kauf: 1, Kein Kauf: 0}`
- Vorhersage in Blattknoten 2
  - Naiver Ansatz
    Wahl „Kauf“ (in Blattknoten Mehrheit für diese Klasse)
  - Fortgeschrittener Ansatz (in der Regel so realisiert)
    Wahl „Kein Kauf“ (Anzahl von „AnzahlProKlasse“ traversierter Knoten!)
    `[{Kauf: 4 = 3 + 1, Kein Kauf: 5 = 5 + 0}]`
- Evaluierung des Baums durch...
  ... Zurückhalten einiger Beobachtungen bei Training

# Hoeffding Bound

- Bisher: Split immer dann, wenn...
  - ... vorgegebene Anzahl neuer Beobachtungen eingetroffen und...
  - ... Anzahl der Beobachtungen im Knoten nicht eindeutig in einer Klasse

- Problem
  - H Tree soll nicht unendlich lange wachsen
  - H Tree soll durch weiteren Split signifikant besser werden

- Lösung: Hoeffding Bound
  - Split nur dann, wenn Informationsgewinn für Split „deutlich besser“...
    ... als Informationsgewinn aller anderen Splits
  - Dabei soll gelten: Informationsgewinn muss umso „deutlich besser“ sein...
    - ... je größer der Wertebereich des Splitattributs
    - ... je weniger Beobachtungen Grundlage für Split bilden
  - Für Split muss also gelten

$$E(\text{beste Alternative}) - E(\text{potentieller Split}) > E \cdot \sqrt{\ln\left(\frac{1}{\delta}\right)/2n}$$

mit $E$: Entropie ohne Split, $n$: Anzahl Beobachtungen, $\delta$: Signifikanzniveau

# Agenda

- Einführung
- Deskriptive Methoden zur Datenexploration
- Datenqualität
- Klassifikation
- Recommender Systems
- Clusteringverfahren

- Stream Mining
  - Einordnung
  - H-Tree
  - CDH-Tree

- Social Network Analysis
- Technische Lösungen
- Datenschutz und gesellschaftliche Aspekte

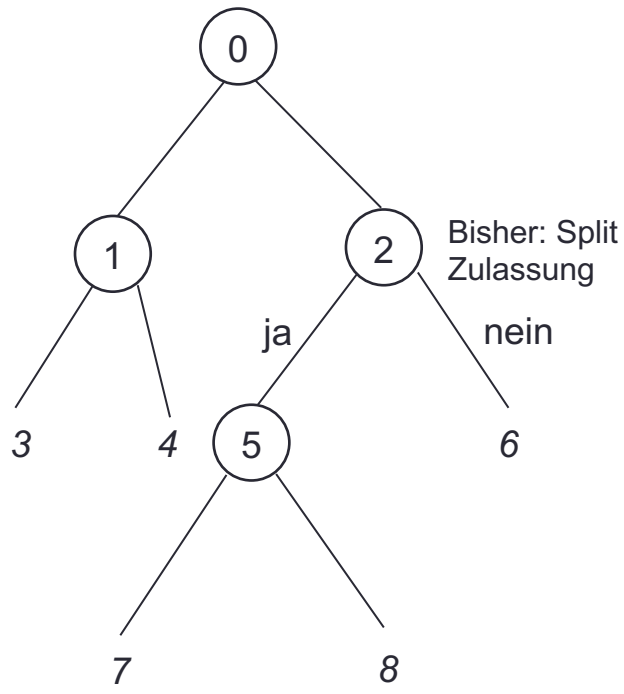
# CDH Tree

- Ziel
  Erweiterung des H Tree, so dass sich das Modell über die Zeit ändern kann

- Inhaltliche Änderungen
  - Alle Arrays werden auch an inneren Knoten aktualisiert
  - Speicherung der Daten über „Sliding Windows“ (FIFO-Methode)
    - Beobachtungen werden in betroffenen Knoten gesammelt
    - Übersteigt die Anzahl der Beobachtungen im Knoten einen Wert $n$...
      ... Löschen der zuerst eingefügten Beobachtungen
  - Für innere Knoten regelmäßige Prüfung, ob Splits noch analog...
    ... Gegebenenfalls: Umstrukturierung des entsprechenden Teilbaums
    - Phase 1: Aufbau
      - Dafür Identifikation „verdächtiger“ Knoten
        (Alternativer Split gemäß Hoeffding Bound besser als aktueller Split)
      - Für jeden „verdächtigen“ Knoten Aufbau eines parallelen Teilbaums
    - Phase 2: Auswahl
      - Berechnung der Vorhersagegüte aller Teilbäume...
        ... Ersetzen des Ursprungsbaums durch besten Teilbaum

# Baumerstellung am Beispiel I

- Identifikation eines neuen “verdächtigen Knotens“ am Beispiel von Knoten 2



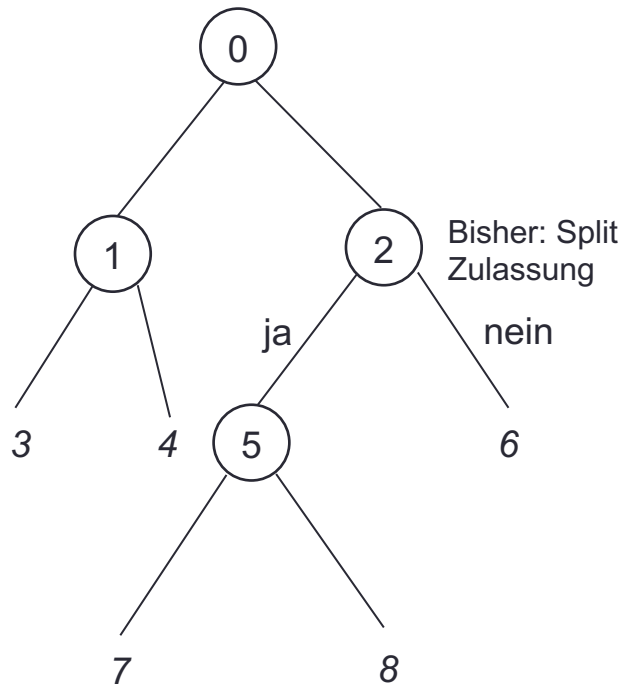

`AnzahlProAttributKlasse`

| | Zulassung | | Alter | | | ∑ |
|---|---|---|---|---|---|---|
| | ja | nein | jung | mittel | alt | |
| Kauf | 12 | 18 | 25 | 3 | 2 | 30 |
| Kein Kauf | 22 | 5 | 1 | 13 | 13 | 27 |
| Summe | 34 | 23 | 26 | 16 | 16 | 57 |

- Entropie vor Split in Knoten 2
  - $E = -\frac{30}{57} \cdot \log_2 \frac{30}{57} - \frac{27}{57} \cdot \log_2 \frac{27}{57} = 0.9980$
- Entropie nach Split Zulassung
  - $E_1 = (-12 \cdot \log_2 12 - 18 \cdot \log_2 18 - 22 \cdot \log_2 22 - 5 \cdot \log_2 5 + 34 \cdot \log_2 34 + 23 \cdot \log_2 23)/57 = 0.8635$
- Entropie nach Split Alter
  - $E_2 = -25 \cdot \log_2 25 - 3 \cdot \log_2 3 - 2 \cdot \log_2 2 - 1 \cdot \log_2 1 - 13 \cdot \log_2 13 - 13 \cdot \log_2 13 + 26 \cdot \log_2 26 + 16 \cdot \log_2 16 + 16 \cdot \log_2 16)/57 = 0.5465$

# Baumerstellung am Beispiel II

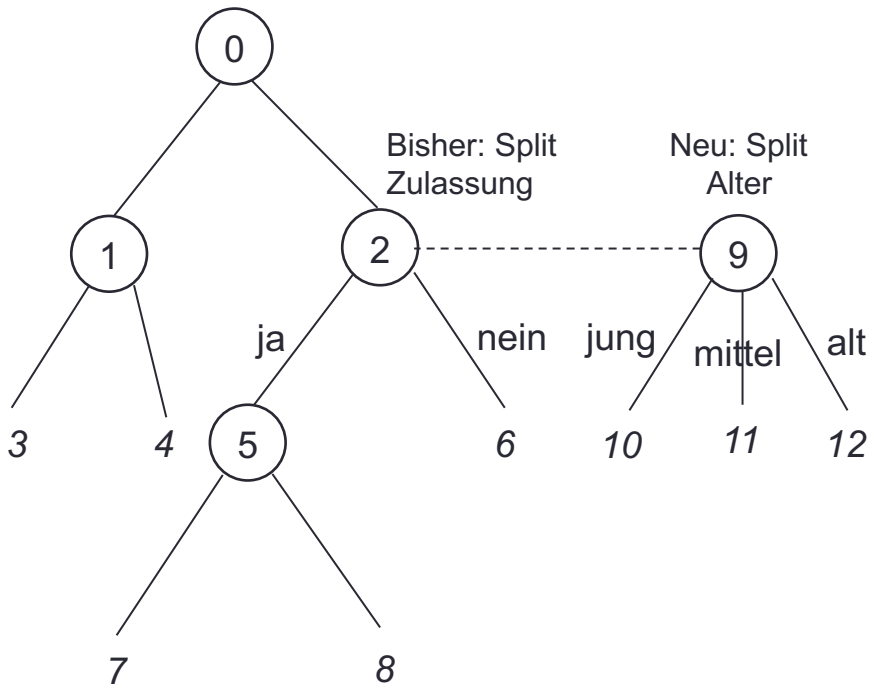
- Identifikation eines neuen “verdächtigen Knotens“ am Beispiel von Knoten 2




- Entropie vor Split
  - $E = 0.9980$
- Entropie nach Split
  - Split bisher (Zulassung): $E_1 = 0.8635$
  - Split neu (Alter): $E_2 = 0.5465$
- Prüfung auf Überschreitung der Hoeffding Bound
  - $E(Split_{bisher}) - E(Split_{neu}) > E \cdot \sqrt{\ln\left(\frac{1}{\delta}\right)/2n}$
  - $0.8635 - 0.5465 > 0.9980 \cdot \sqrt{\frac{\ln\left(\frac{1}{0.001}\right)}{2 \cdot 57}}$
  - $0.3170 > 0.2457$
- ⇒ Knoten 2 ist verdächtig

# Baumerstellung am Beispiel III

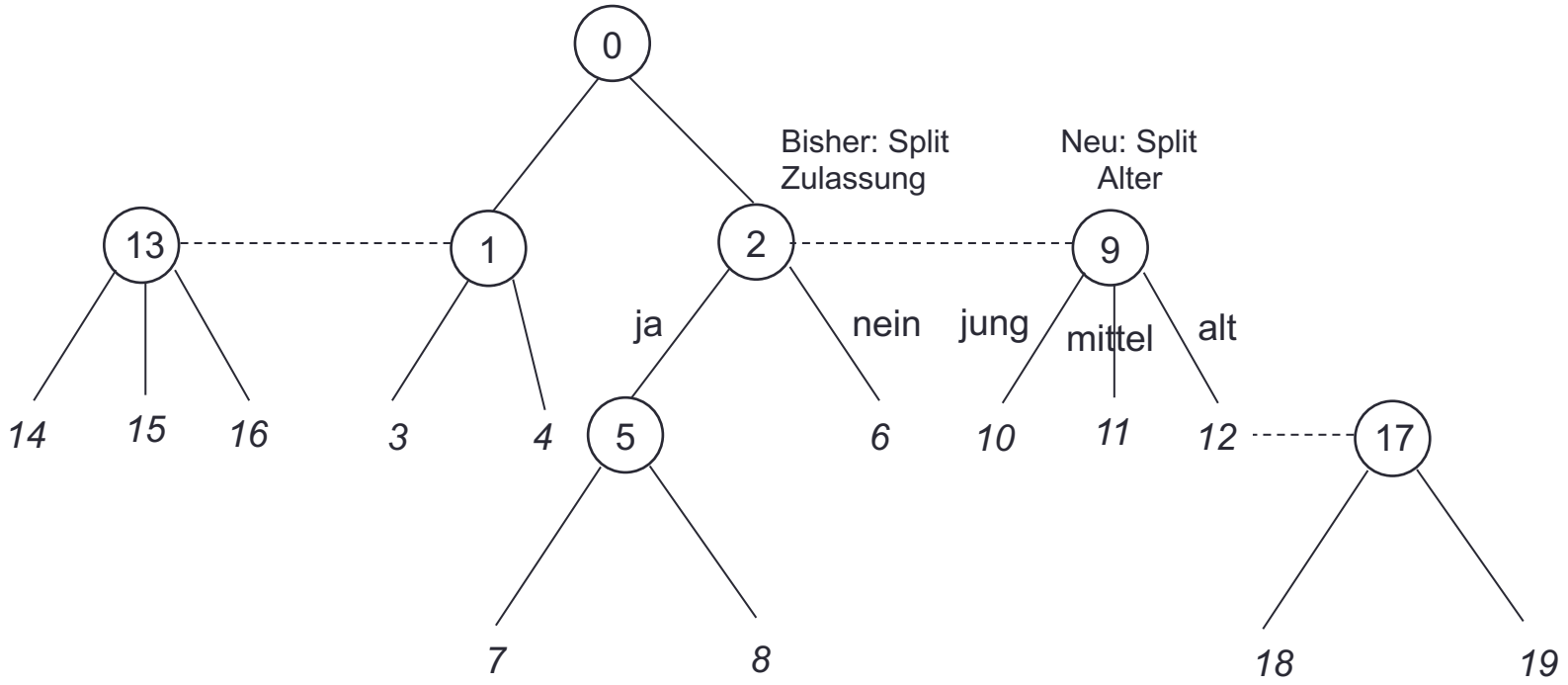
- Baum nach Einfügen eines alternativen Teilbaums




- Alle Daten für Knoten 9 werden...
  ... von Knoten 2 übernommen
- Neue Beobachtungen werden in...
  ... beiden Knoten (also 2 und 9)...
  ... Parallel verarbeitet
- Einfügen neuer Beobachtungen...
  ... für feste Anzahl von Beobachtungen
- Am Ende Baum mit mehreren..
  ... Alternativen Teilbäumen

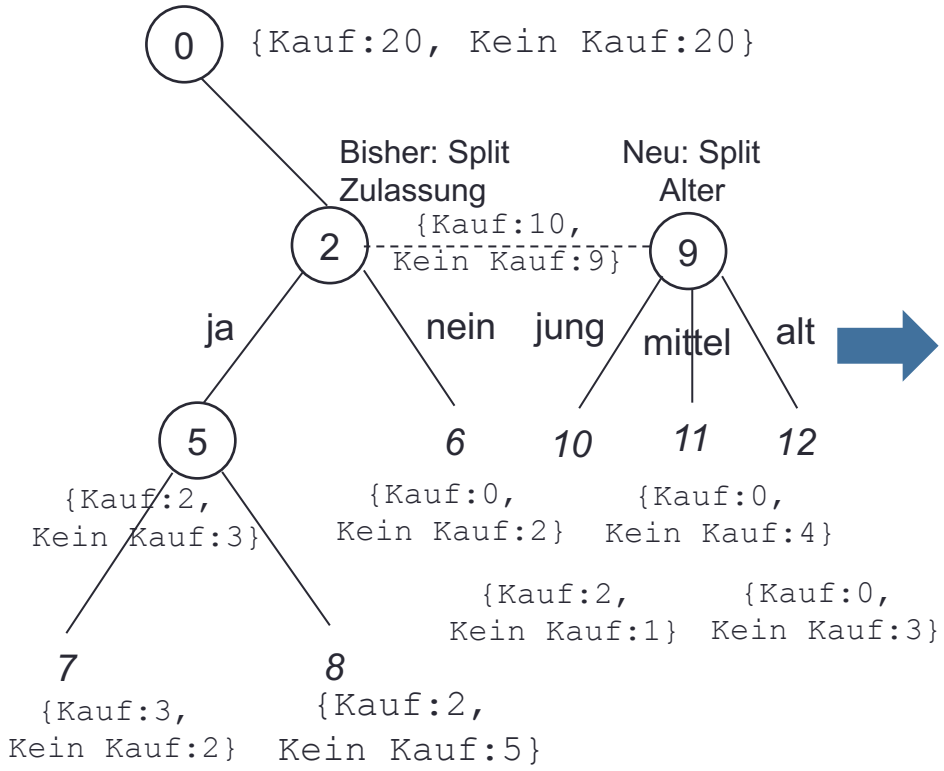
# Baumerstellung am Beispiel IV

- Ergebnis der Phase „Baumerstellung“



# Baumerstellung am Beispiel V

- Vorbereitung „Phase Auswahl“

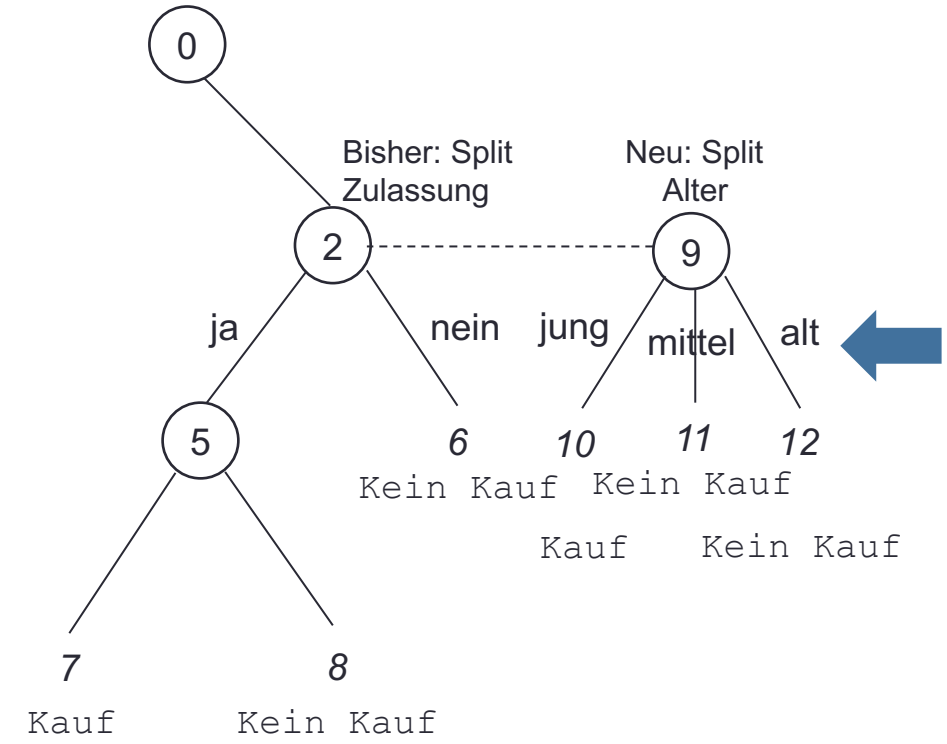



- Für jeden Blattknoten im Baum
  - Ableiten von
    `AnzahlProKlasse:`
    `{Kauf:1, Kein Kauf:0}`
  - ... mit Hilfe der inneren Knoten
  - d.h. Aufaddieren aller inneren Knoten

| Knoten | AnzahlproKlasse |
|---|---|
| 6 | {Kauf:30 (=0+10+20), Kein Kauf:31 (=2+9+20)} |
| 7 | {Kauf:35, Kein Kauf:34} |
| 8 | {Kauf:34, Kein Kauf:37} |
| 10 | {Kauf:32, Kein Kauf:30} |
| 11 | {Kauf:30, Kein Kauf:33} |
| 12 | {Kauf:30, Kein Kauf:32} |

# Baumerstellung am Beispiel VI

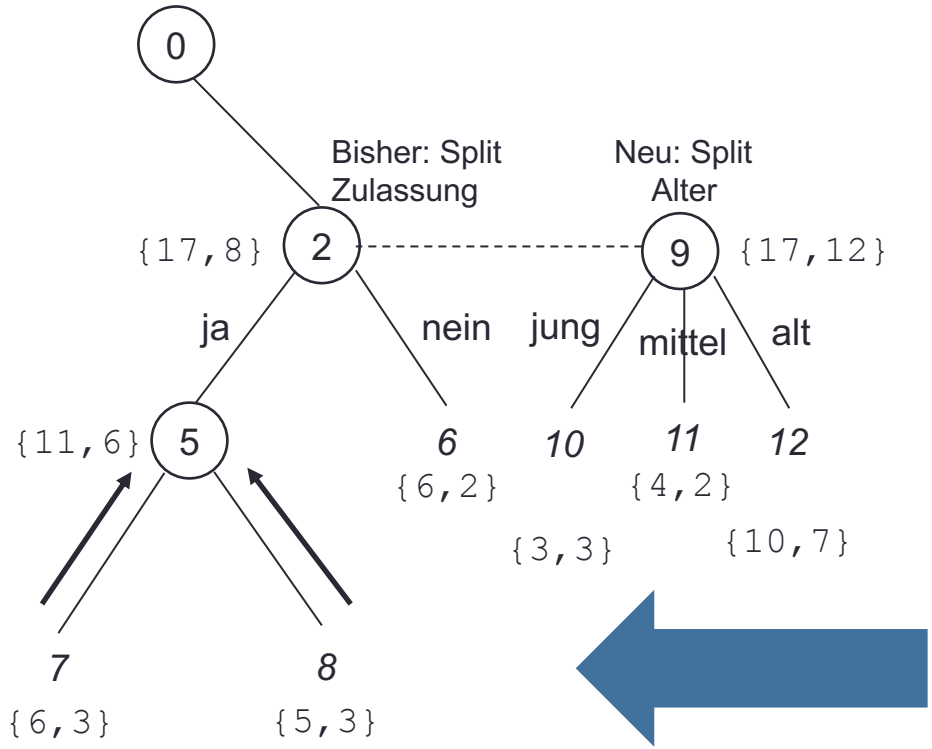
- Ableiten der Vorhersage pro Blattknoten aus Tabelle




| Knoten | AnzahlproKlasse |
|---|---|
| 6 | {Kauf:30, Kein Kauf:31} |
| 7 | {Kauf:35, Kein Kauf:34} |
| 8 | {Kauf:34, Kein Kauf:37} |
| 10 | {Kauf:32, Kein Kauf:30} |
| 11 | {Kauf:30, Kein Kauf:33} |
| 12 | {Kauf:30, Kein Kauf:32} |

# Baumerstellung am Beispiel VII

- Klassifikation von Testdaten und Ableiten einer weiteren Tabelle




Vorhersage 7: Kauf
d.h 6 Testdaten,
davon 3 Kauf

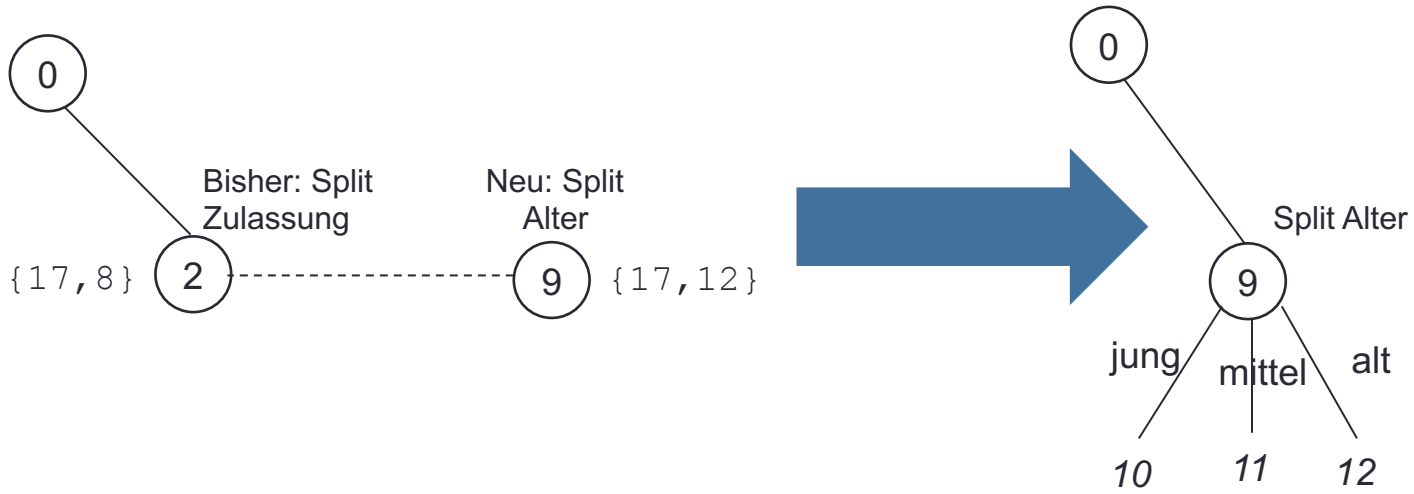
Vorhersage 8: Kein Kauf
d.h. 5 Testdaten,
davon 3 Kein Kauf

- Ableiten der Tabelle (s.u.)
- Induktion entsprechender Werte für innere Knoten des Entscheidungsbaums

| Knoten | Anzahl Test | Anzahl korrekte Prognose |
|---|---|---|
| 6 | 6 | 2 |
| 7 | 6 | 3 |
| 8 | 5 | 3 |
| 10 | 3 | 3 |
| 11 | 4 | 2 |
| 12 | 10 | 7 |

# Baumerstellung am Beispiel VIII

- Knoten 9: 12 richtige Vorhersagen; Knoten 2: 8 richtige Vorhersagen
- Knoten 9 ist besser ⇒ Knoten 2 wird entfernt
- Vorgehen wird bis zur Wurzel für alle Knoten durchgeführt
- Sobald Umstrukturierung abgeschlossen...
  ... Beginn neuer Aufbauphase



# H Tree und CDH Tree

- Vorteile der Verfahren gegenüber Entscheidungsbäumen
  - Erlauben kontinuierliches Trainieren und Testen des Modells
  - Anwendung des Modells schon während des Trainierens möglich
  - Verfahren sind performant, da...
    - ... relevante Zwischenergebnisse vorgehalten werden
    - ... Ermittlung abhängig von absoluten Zahlen
  - Kontinuierliche Anpassung (CDH Tree)...
    ... kann auch wandelnde Welt abbilden

- Nachteile
  - Evaluation im „klassischen Sinn“ schwer / kaum möglich
  - „Ausprobieren“ verschiedener Ansätze und Vergleich kaum möglich
  - Management muss Einsatz vertrauen
  - Wie wahrscheinlich ist ein „Use Case“ ohne historische Daten?
    ... bzw. ist Sammeln von Daten und damit höhere Flexibilität nicht besser?