Machine Learning

Prof. Dr. Fabian Brunner

<fa.brunner@oth-aw.de>

Amberg, 24. Oktober 2023

Übersicht

Ostbayerische Technische Hochschule
Amberg-Weiden

Thema heute: Entscheidungsbaum-Lernen

- Funktionsweise baumbasierter Ansätze zur Klassifikation
- Entropie und Gini-Index als Maße für Unreinheit
- Vorzüge und Nachteile baumbasierter Ansätze
- Einordnung von Entscheidungsbaum-Lernen
- DecisionTrees in Scikit-learn (Übung)
- Rekursive Erzeugung von Entscheidungbäumen (Übung)

Literatur für dieses Thema:

T. Mitchell: Machine Learning. McGraw-Hill. 1997

Einführendes Beispiel

Aufgabenstellung: prognostiziere, ob John Tennis spielen wird.

Trainingsdaten:

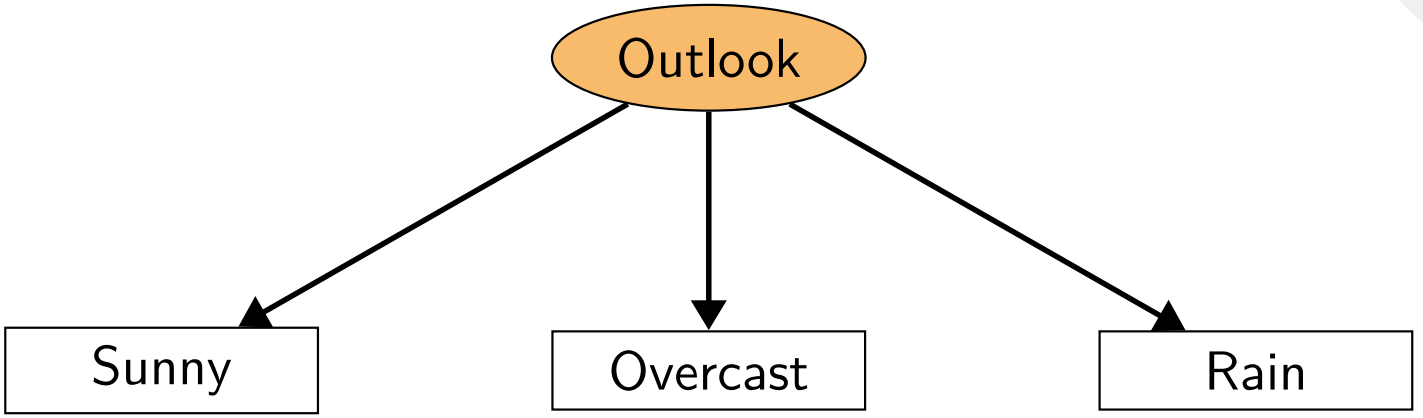
Day	Outlook	Humidity	Wind	Play
1	Sunny	High	Weak	No
2	Sunny	High	Strong	No
3	Overcast	High	Weak	Yes
4	Rain	High	Weak	Yes
5	Rain	Normal	Weak	Yes
6	Rain	Normal	Strong	No
7	Overcast	Normal	Strong	Yes
8	Sunny	High	Weak	No
9	Sunny	Normal	Weak	Yes
10	Rain	Normal	Weak	Yes
11	Sunny	Normal	Strong	Yes
12	Overcast	High	Strong	Yes
13	Overcast	Normal	Weak	Yes
14	Rain	High	Strong	No

$p = 3, m = 14, n = 2$

Neuer Datensatz (Query Point):

15	Sunny	High	Weak	?

Split der Daten



Day	Outlook	Humidity	Wind
1	Sunny	High	Weak
2	Sunny	High	Strong
8	Sunny	High	Weak
9	Sunny	Normal	Weak
11	Sunny	Normal	Strong

2 yes/3 no
→ **weiterer Split**

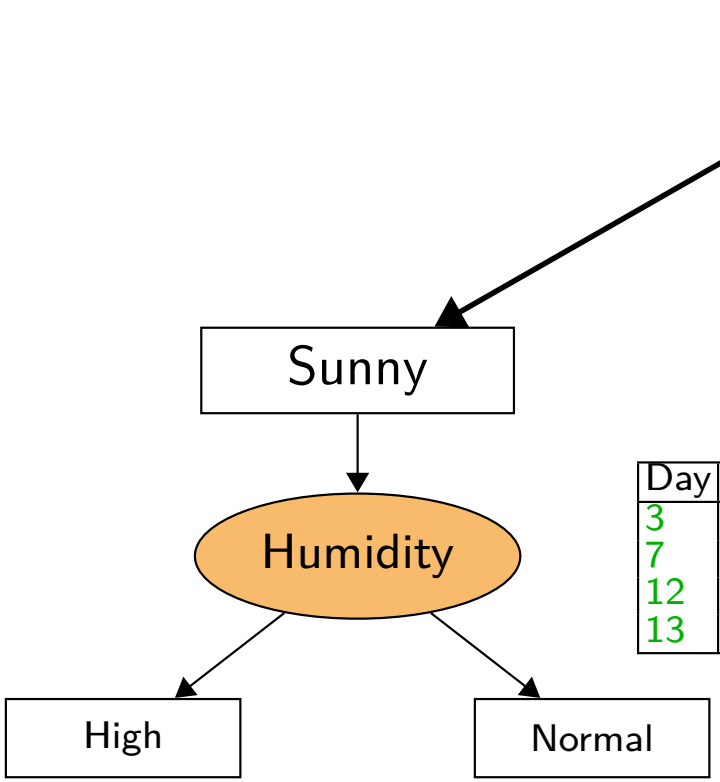
Day	Outlook	Humidity	Wind
3	Overcast	High	Weak
7	Overcast	Normal	Strong
12	Overcast	High	Strong
13	Overcast	Normal	Weak

4 yes/0 no
reines Sample

Day	Outlook	Humidity	Wind
4	Rain	High	Weak
5	Rain	Normal	Weak
6	Rain	Normal	Strong
10	Rain	Normal	Weak
14	Rain	Normal	Strong

3 yes/2 no
→ **weiterer Split**

Split der Daten



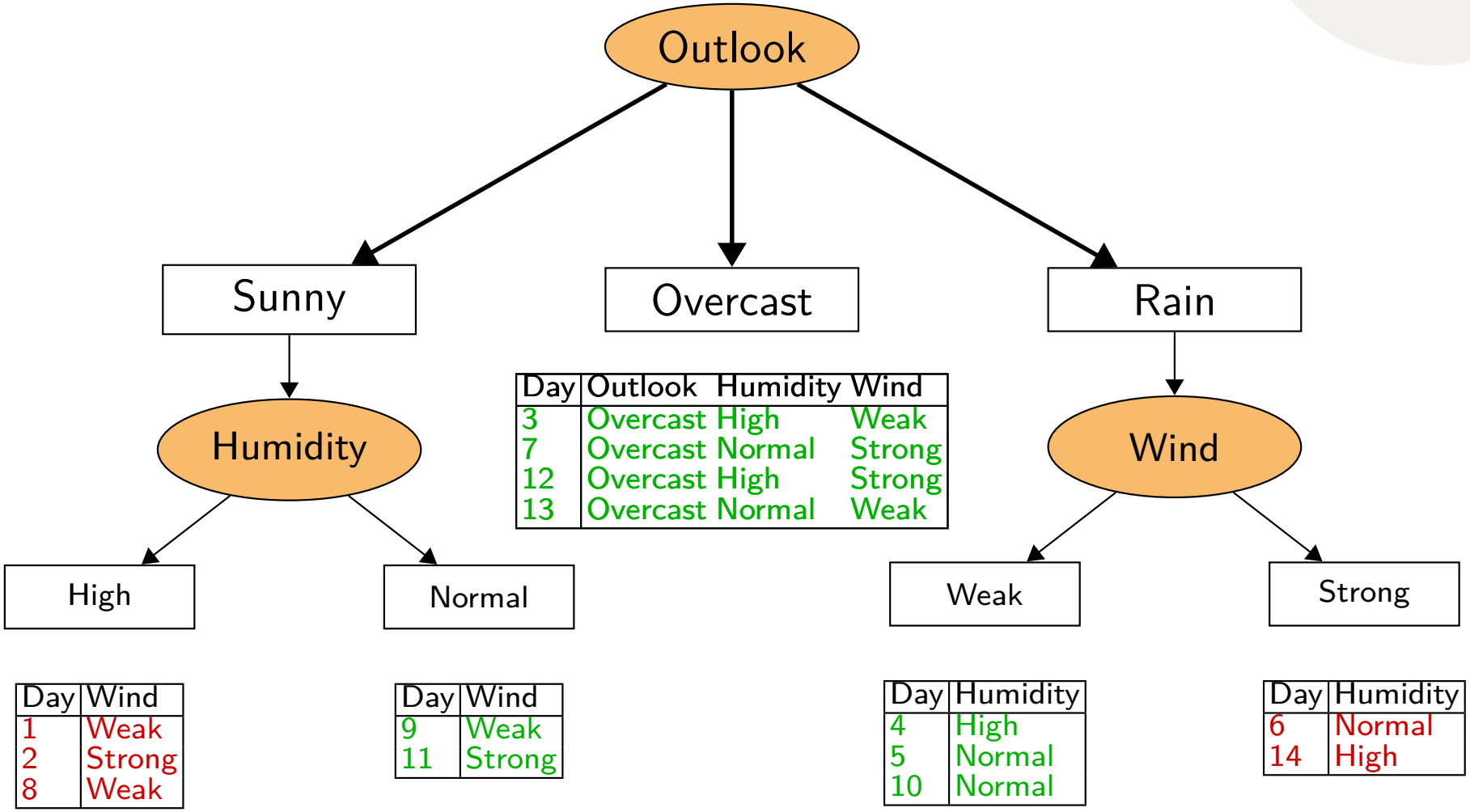
Day	Outlook	Humidity	Wind
3	Overcast	High	Weak
7	Overcast	Normal	Strong
12	Overcast	High	Strong
13	Overcast	Normal	Weak

Day	Outlook	Humidity	Wind
4	Rain	High	Weak
5	Rain	Normal	Weak
6	Rain	Normal	Strong
10	Rain	Normal	Weak
14	Rain	Normal	Strong

Day	Wind
1	Weak
2	Strong
8	Weak

Day	Wind
9	Weak
11	Strong

Split der Daten



Day	Outlook	Humidity	Wind
3	Overcast	High	Weak
7	Overcast	Normal	Strong
12	Overcast	High	Strong
13	Overcast	Normal	Weak

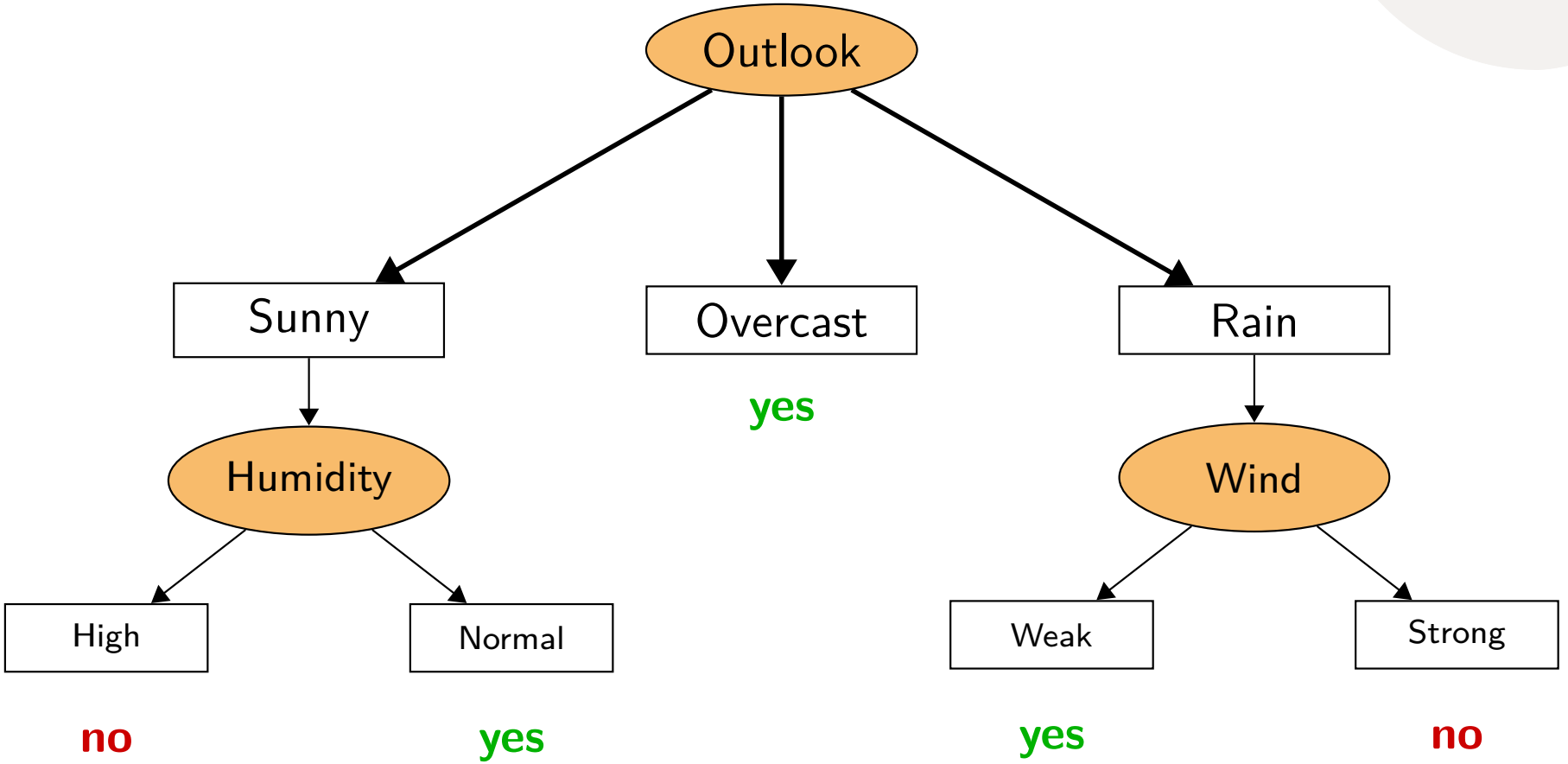
Day	Wind
1	Weak
2	Strong
8	Weak

Day	Wind
9	Weak
11	Strong

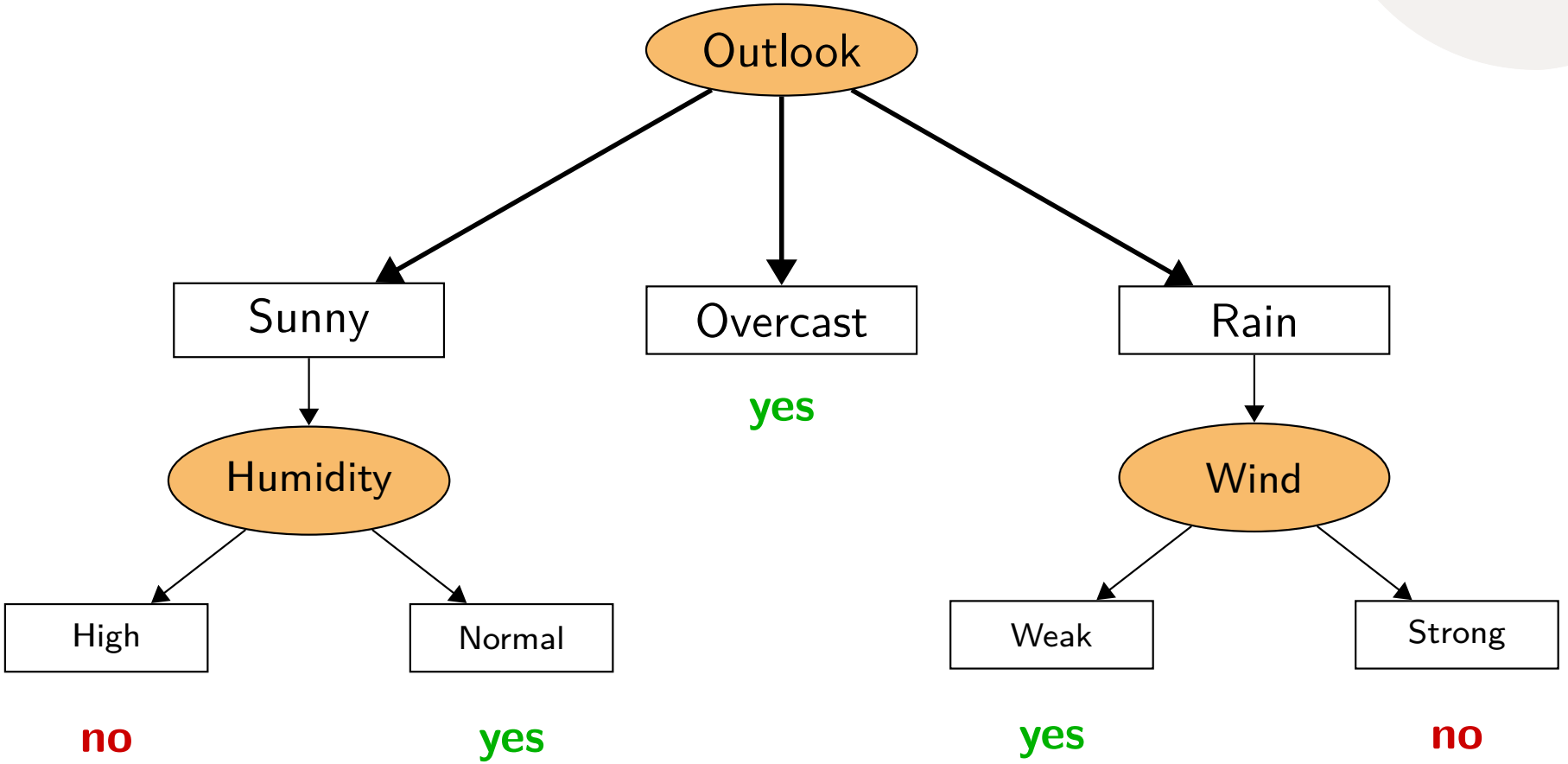
Day	Humidity
4	High
5	Normal
10	Normal

Day	Humidity
6	Normal
14	High

Split der Daten



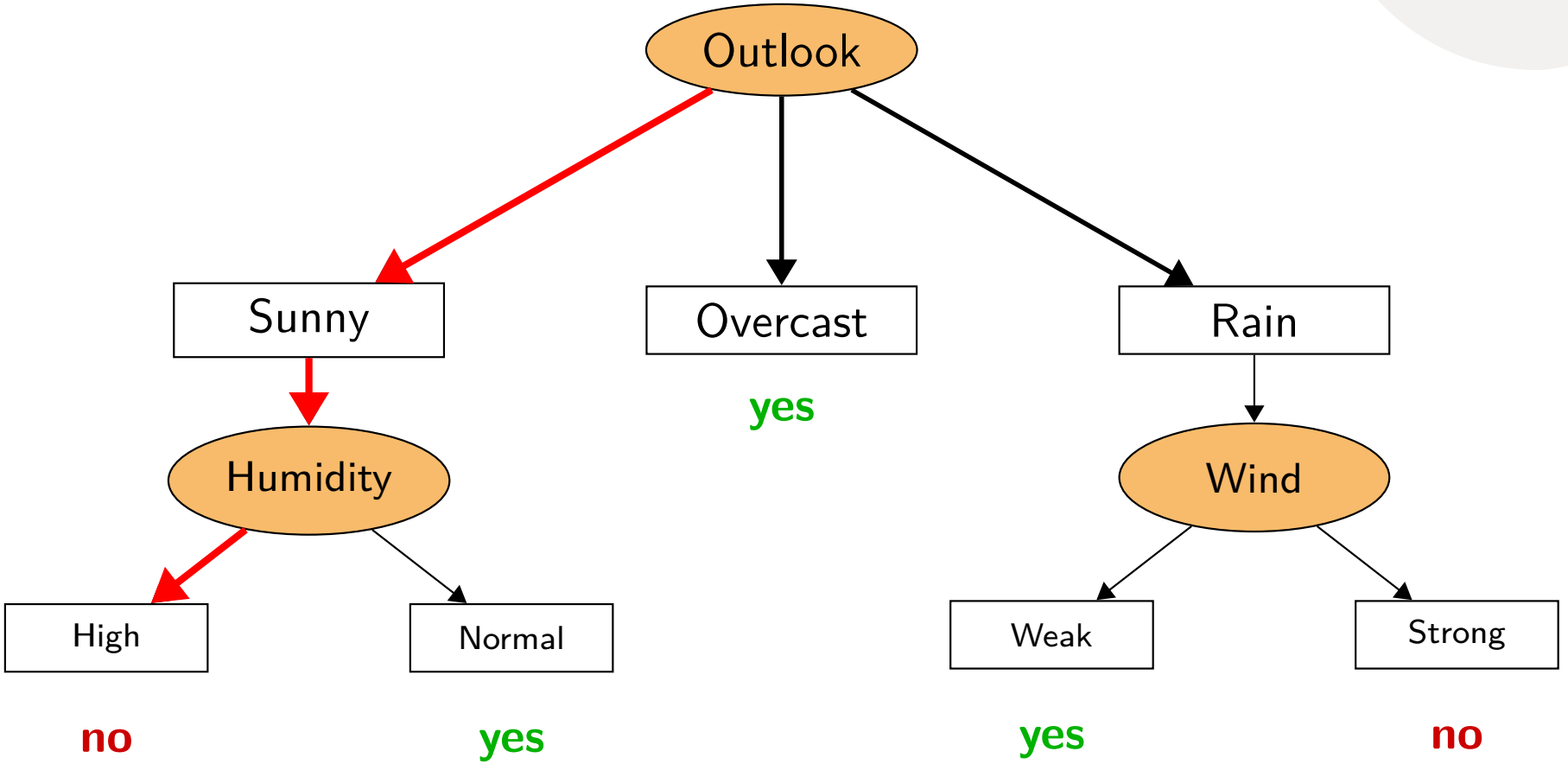
Split der Daten



Neue Daten:

Day	Outlook	Humid	Wind
15	Sunny	High	Weak

Split der Daten



Neue Daten:

Day	Outlook	Humid	Wind	
15	Sunny	High	Weak	→ **no**

Repräsentation eines Entscheidungsbaums

Bestandteile des Baums

- Jeder (orange) Knoten repräsentiert ein Attribut
- Jede Kante (weiße Rechtecke) repräsentiert eine Ausprägung.
- Jedes Blatt repräsentiert eine Entscheidung/einen Prognosewert.

Repräsentierte Regeln

- Durch den Entscheidungsbaum wird eine Disjunktion von Konjunktionen von Bedingungen an die Attributwerte als Entscheidungskriterium dargestellt.
- Für obiges Beispiel: Entscheidung für „yes“ falls

$$
\begin{aligned}
&((\mathit{Outlook} = \mathit{Sunny}) \wedge (\mathit{Humidity} = \mathit{Normal})) \\
&\vee (\mathit{Outlook} = \mathit{Overcast}) \\
&\vee ((\mathit{Outlook} = \mathit{Rain}) \wedge (\mathit{Wind} = \mathit{Weak}))
\end{aligned}
$$

oder (∨) und (∧)

Kriterien für Split der Daten

Grundidee beim Entscheidungsbaum-Lernen

- Erstelle einen Baum, bei dem in jedem Knoten die Trainingsdaten anhand eines Merkmals gesplittet werden.
- Setze den Prozess so lange fort, bis alle Blätter rein bezüglich der Zielvariablen sind.
- Ordne jedem Blatt den Zielfunktionswert der enthaltenen Samples zu.
- Klassifikation eines neuen Objekts: Durchlaufe den Baum und ordne dem Objekt die Klasse des Blatts zu, in das es fällt.

Kriterien für Split der Daten

Grundidee beim Entscheidungsbaum-Lernen

- Erstelle einen Baum, bei dem in jedem Knoten die Trainingsdaten anhand eines Merkmals gesplittet werden.
- Setze den Prozess so lange fort, bis alle Blätter rein bezüglich der Zielvariablen sind.
- Ordne jedem Blatt den Zielfunktionswert der enthaltenen Samples zu.
- Klassifikation eines neuen Objekts: Durchlaufe den Baum und ordne dem Objekt die Klasse des Blatts zu, in das es fällt.

Nach welchen Kriterien sollte gesplittet werden?

- Reinheit/Homogenität der Samples bezüglich der Zielvariable
- Möglichst große Samples

Kriterien für Split der Daten

Ostbayerische Technische Hochschule Amberg-Weiden

Grundidee beim Entscheidungsbaum-Lernen

- Erstelle einen Baum, bei dem in jedem Knoten die Trainingsdaten anhand eines Merkmals gesplittet werden.
- Setze den Prozess so lange fort, bis alle Blätter rein bezüglich der Zielvariablen sind.
- Ordne jedem Blatt den Zielfunktionswert der enthaltenen Samples zu.
- Klassifikation eines neuen Objekts: Durchlaufe den Baum und ordne dem Objekt die Klasse des Blatts zu, in das es fällt.

Nach welchen Kriterien sollte gesplittet werden?

- Reinheit/Homogenität der Samples bezüglich der Zielvariable
- Möglichst große Samples

Frage:

- Wie misst man „Reinheit“ mathematisch?

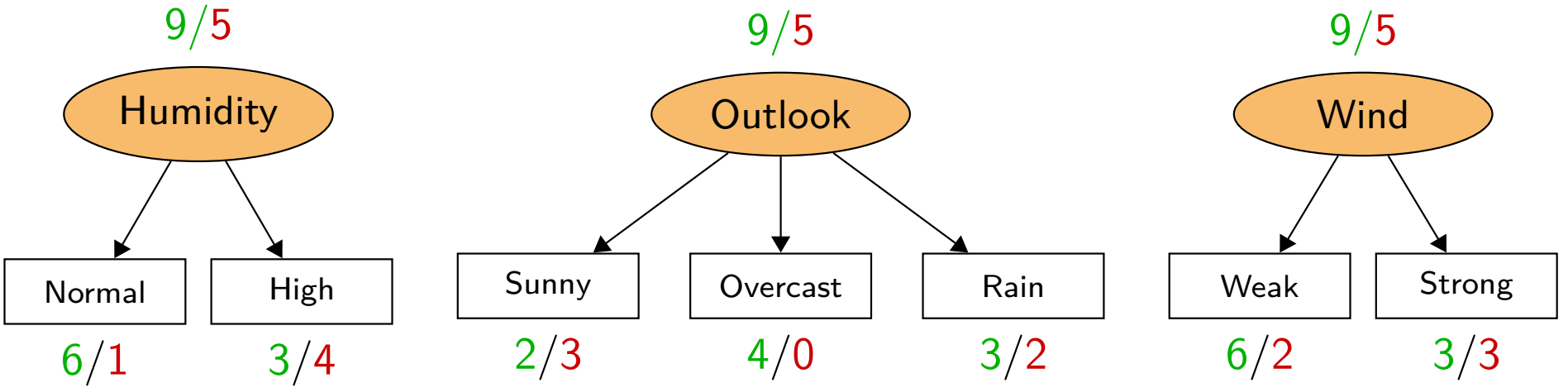
Rekursive Baumerzeugung - Pseudocode für den ID3-Algorithmus

Ostbayerische Technische Hochschule Amberg-Weiden

Der Entscheidungsbaum wird rekursiv erstellt:

function SPLIT(node, X)
    determine the best attribute A to split the data X
  for each value *val* of A **do**
    create a child node
    X_{child} = subset of X where $A = val$
    if X_{child} is pure **then** ~~**return**~~ continue
    else
      SPLIT(child node, X_{child})
    end if
  end for
end function

Auswahl des Attributs für einen Split



- Wähle das Attribut, für das der Split zu möglichst „reinen“ Subsamples führt.
- Bester Fall: reines Subsample (z.B. 4/0 für Outlook=„Overcast“)
- Schlechtester Fall: Gleichverteilung der Zielvariablen im Subsample (z.B. 3/3 für Wind=„Strong“)
- Benötigt: Maß für die Reinheit von Subsamples bezüglich der Zielvariablen

Purity measures

Ziel:

- Kennzahl für die Reinheit eines Samples
- Vorschlag: Anteil der Subsamples mit Zielfunktionswert „yes“.

Purity measures

Ziel:

- Kennzahl für die Reinheit eines Samples
- Vorschlag: Anteil der Subsamples mit Zielfunktionswert „yes“.
- Problem: $4/0$ und $0/4$ sind beide rein! $\rightarrow$ Symmetrie erforderlich.

Purity measures

Ziel:

- Kennzahl für die Reinheit eines Samples
- Vorschlag: Anteil der Subsamples mit Zielfunktionswert „yes“.
- Problem: 4/0 und 0/4 sind beide rein! $\rightarrow$ Symmetrie erforderlich.

Entropiefunktion

Man betrachte ein Klassifikationsproblem mit den n Klassen $C_1, \ldots, C_n$. Sei p_i der Anteil der Samples mit Zielfunktionswert C_i in einem Sample S der Trainingsdatenmenge. Dann ist die **Entropie von** S definiert durch

$$H(S) = -\sum_{i=1}^{n} p_i \log_2(p_i) \ .$$

Interpretation:

- Die Entropie $H(S)$ gibt an, wie viele Bits notwendig sind, um zu entscheiden, welcher Klasse ein Datensatz angehört.
- Frage: welches Maximum und Minimum nimmt H an?

Entropie-Funktion

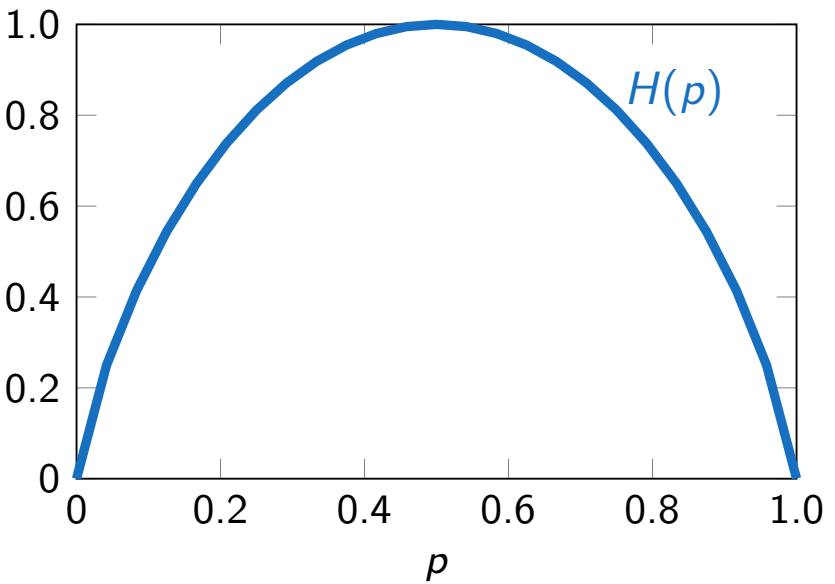
Beispiel für unser binäres Klassifikationsproblem ($n = 2$):

- Reines Sample (4/0):

$$H(S) = -\frac{4}{4}\log_2\left(\frac{4}{4}\right) - \frac{0}{4}\log_2\left(\frac{0}{4}\right) = 0 \text{ bits}$$

- Perfekt durchmischtes Sample (3/3):

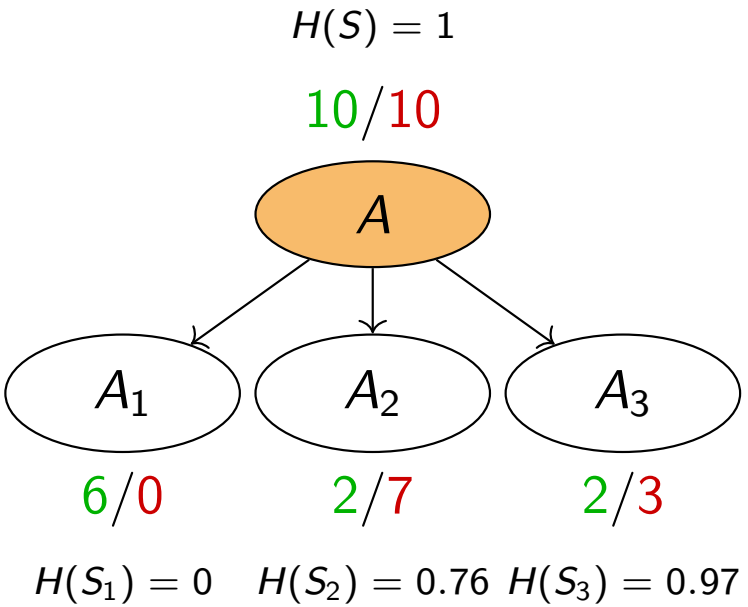
$$H(S) = -\frac{3}{6}\log_2\left(\frac{3}{6}\right) - \frac{3}{6}\log_2\left(\frac{3}{6}\right) = 1 \text{ bit} .$$

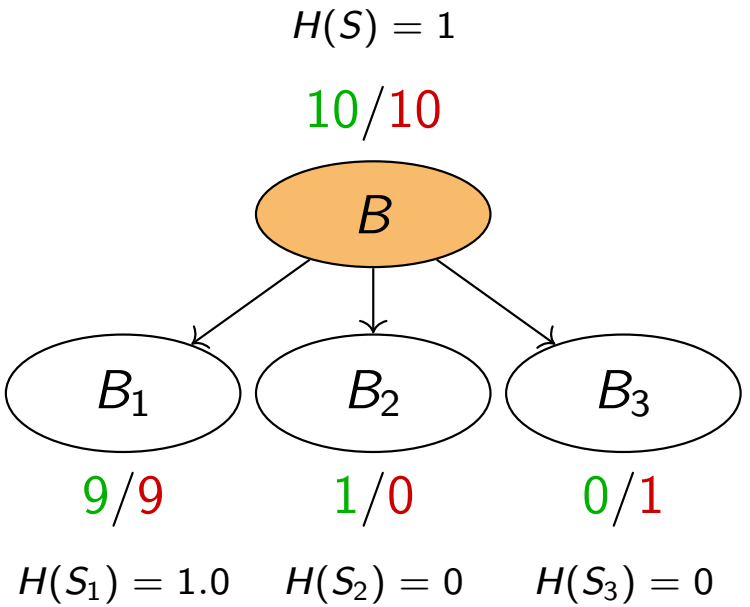


Auswahl des Split-Attributs

Angenommen, es stehen zwei Attribute A und B für einen Split wie folgt zur Auswahl:

Große reine samples mehr aussagekräfti

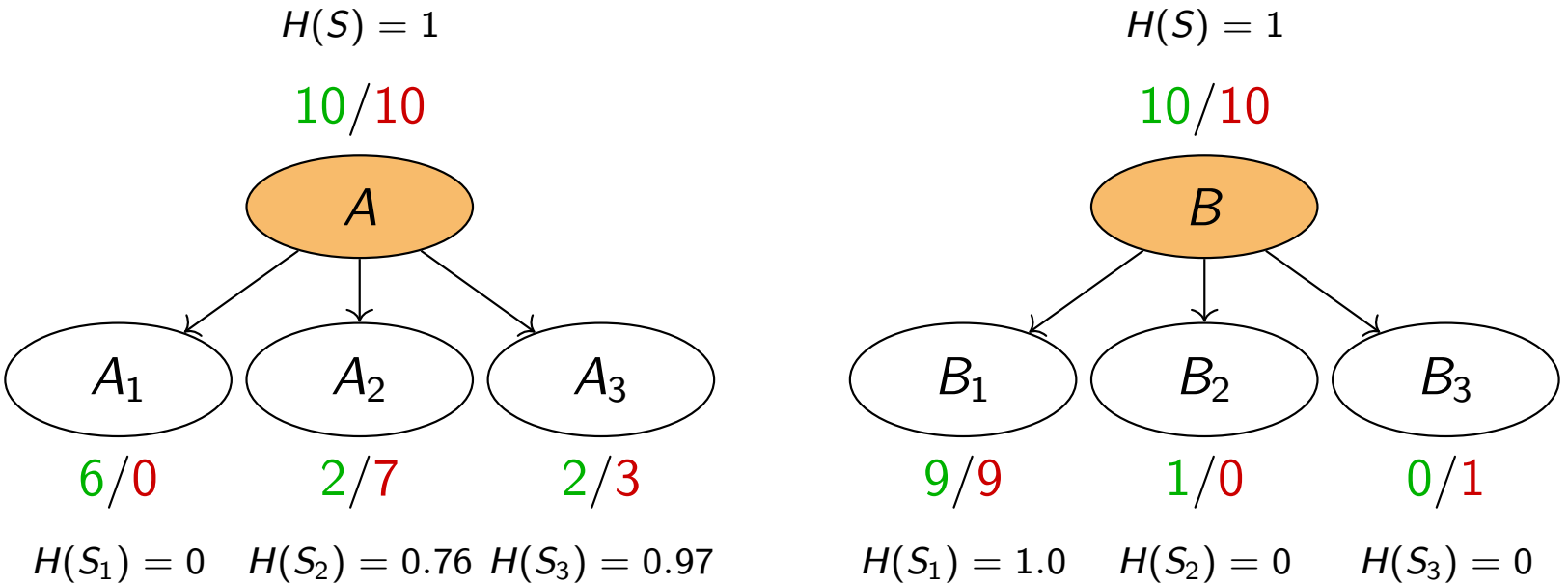




Welches Merkmal würden Sie für einen Split auswählen?

Auswahl des Split-Attributs

Angenommen, es stehen zwei Attribute A und B für einen Split wie folgt zur Auswahl:



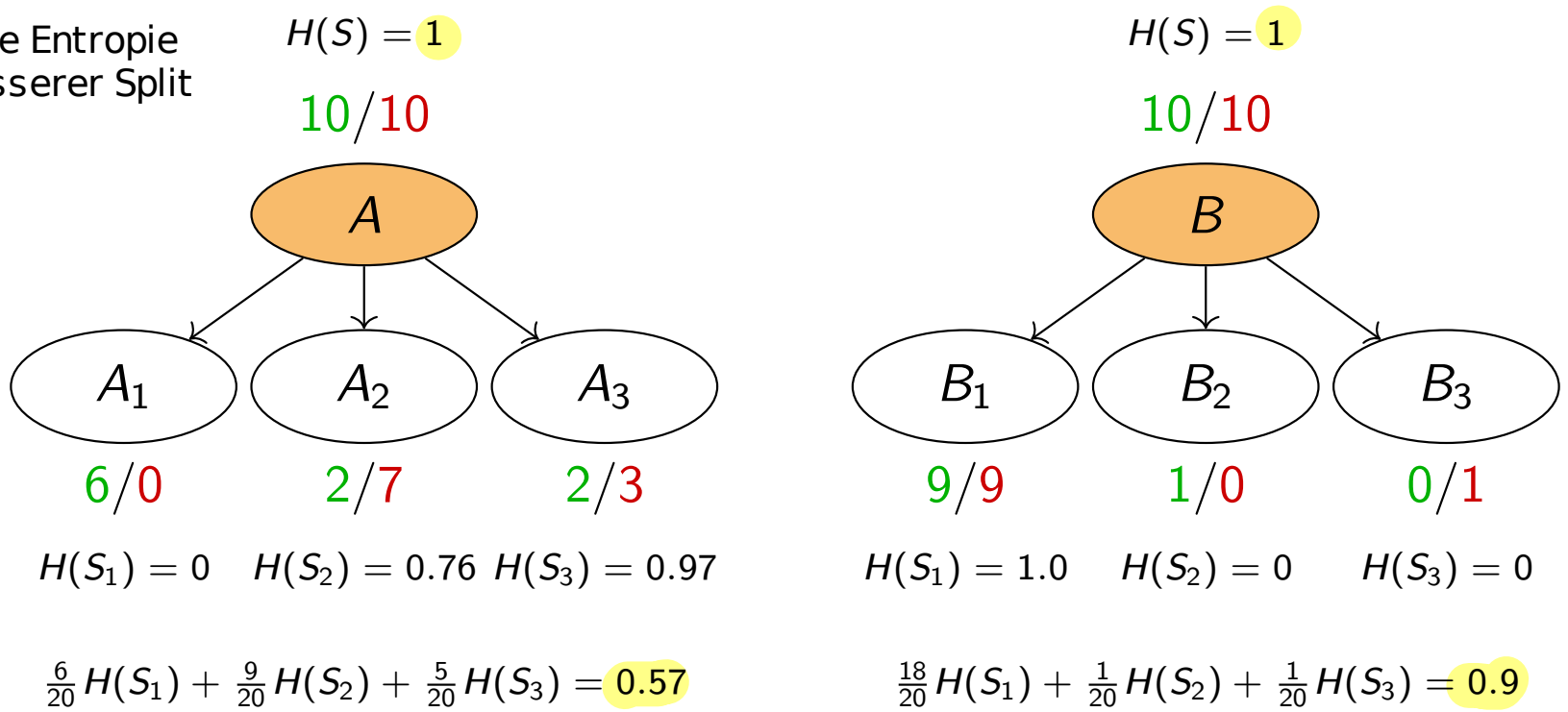
Welches Merkmal würden Sie für einen Split auswählen?

Idee: Berechne gewichteten Mittelwert der Entropien der entstehenden Subsamples, um auch die Größe der entstehenden Samples zu berücksichtigen.

Auswahl des Split-Attributs

Angenommen, es stehen zwei Attribute A und B für einen Split wie folgt zur Auswahl:

kleinere Entropie
-> besserer Split



Welches Merkmal würden Sie für einen Split auswählen?

Idee: Berechne gewichteten Mittelwert der Entropien der entstehenden Subsamples, um auch die Größe der entstehenden Samples zu berücksichtigen.

Information Gain

Auswahl eines Attributs für Split:

- Wünschenswert: reine Subsamples sollen möglichst groß sein.
- Idee: berücksichtige die durch die Elementanzahl gewichtete mittlere Entropie nach einem Split und vergleiche mit der Entropie vor dem Split.
- Der „Information Gain“ (s. unten) gibt die (gewichtete) mittlere Reduktion der Entropie durch den Split an einem bestimmten Attribut an.
- Vorgehen beim Entscheidungsbaum-Lernen: wähle diejenige Variable für einen Split aus, für die der Information Gain maximiert wird.

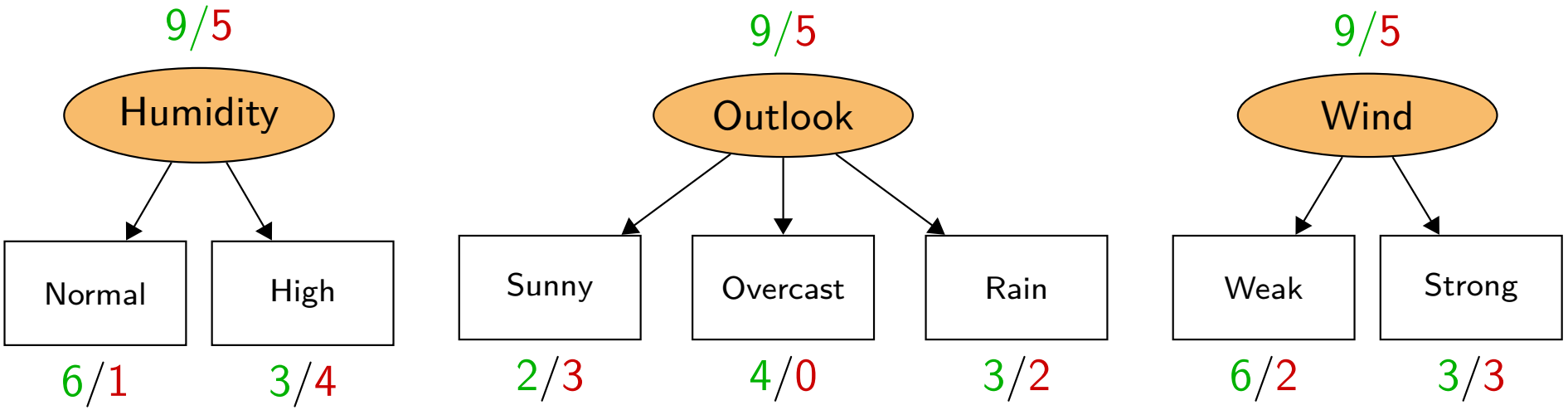
Information Gain beim Split an Attribut A

$$Gain(S, A) = H(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} H(S_v)$$

Bemerkung: statt der Entropie H können auch andere sog. „impurity measures“ verwendet werden, um den Information Gain zu definieren. Dazu muss nur H in der obigen Formel ersetzt werden.

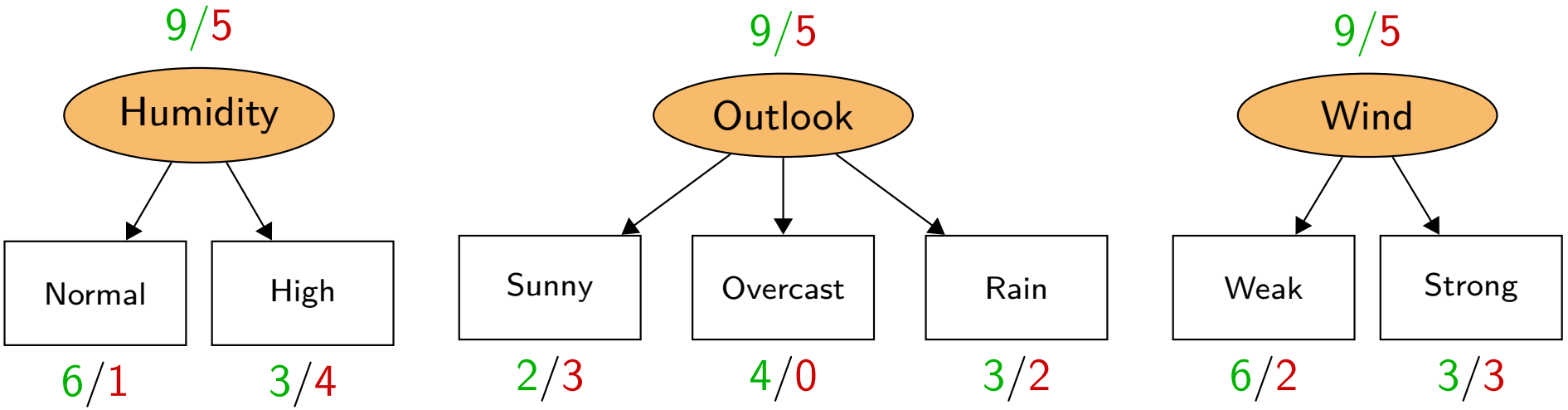
Beispiel

Berechnen Sie den Information Gain für die folgenden Splits:



Beispiel

Berechnen Sie den Information Gain für die folgenden Splits:



Ergebnis:

$$Gain(S, Humidity) = 0.151$$
$$Gain(S, Outlook) = 0.246$$
$$Gain(S, Wind) = 0.048$$

bester Infomationsgewinn
-> erster split

Greedy algorithmus. findet immer nur den nächstbesten baum
nicht ungedigt den allerbesten

Gini Impurity

Ein weiteres Maß für die (Un-)Reinheit eines Samples („impurity measure“) ist die sog. **Gini impurity** (bzw. **Gini-Index**).

Gini Impurity für ein Sample S

Man betrachte ein Klassifikationsproblem mit den n Klassen $C_1, \ldots, C_n$. Sei p_i der Anteil der Samples mit Zielfunktionswert C_i in einem Sample S der Trainingsdatenmenge. Dann ist die **Gini impurity von** S definiert durch

$$Gini(S) = 1 - \sum_{i=1}^{n} p_i^2 \ .$$

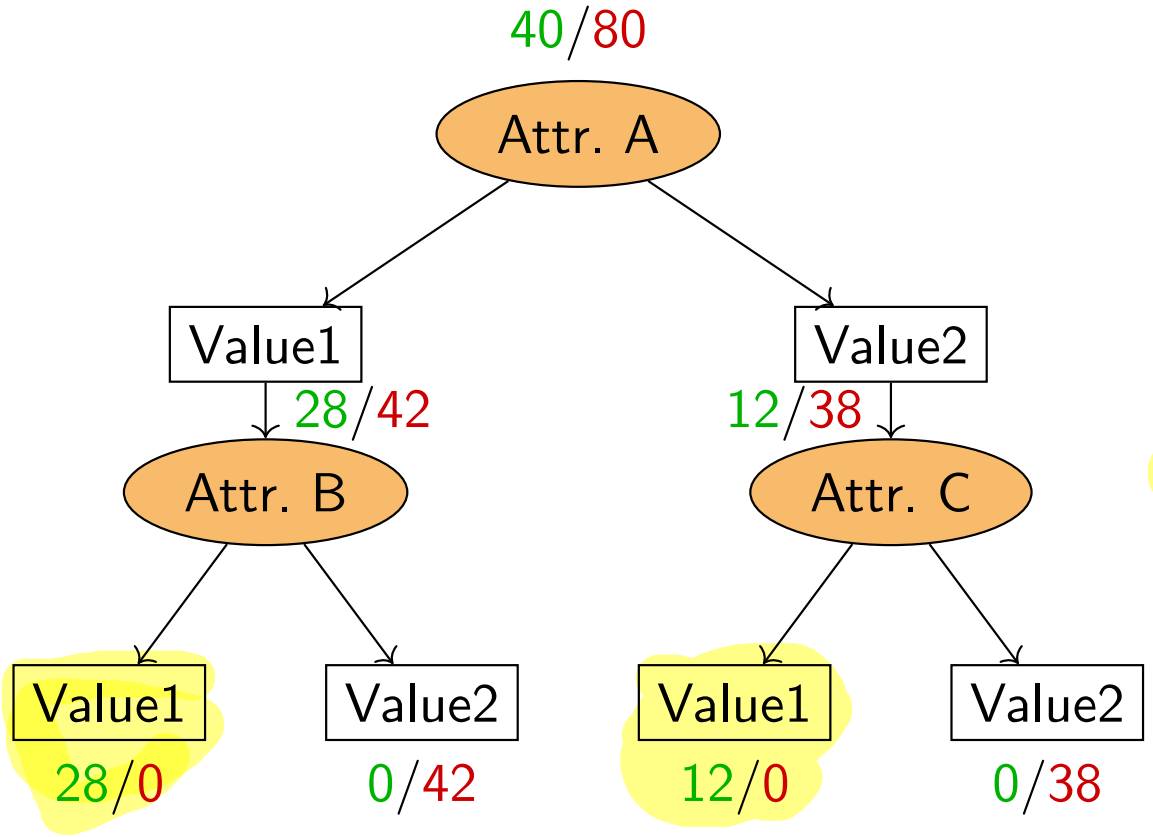
- Der Information Gain unter Verwendung der Gini impurity beim Split an einem Attribut A lautet:

$$Gain(S, A) = Gini(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Gini(S_v) \ .$$

- Wähle Attribut A für den Split aus, für das $Gain(S, A)$ maximiert wird, d.h. für den die gewichtete Gini impurity minimiert wird.

Misclassification Error als Splitting-Kriterium

Bei Verwendung des Misclassification Errors als „impurity measure“ hätte man im folgenden Beispiel einen Information Gain von 0 beim Split an Attr. A:



Misclassification Error:

$$\frac{40}{120} = \frac{1}{3}$$

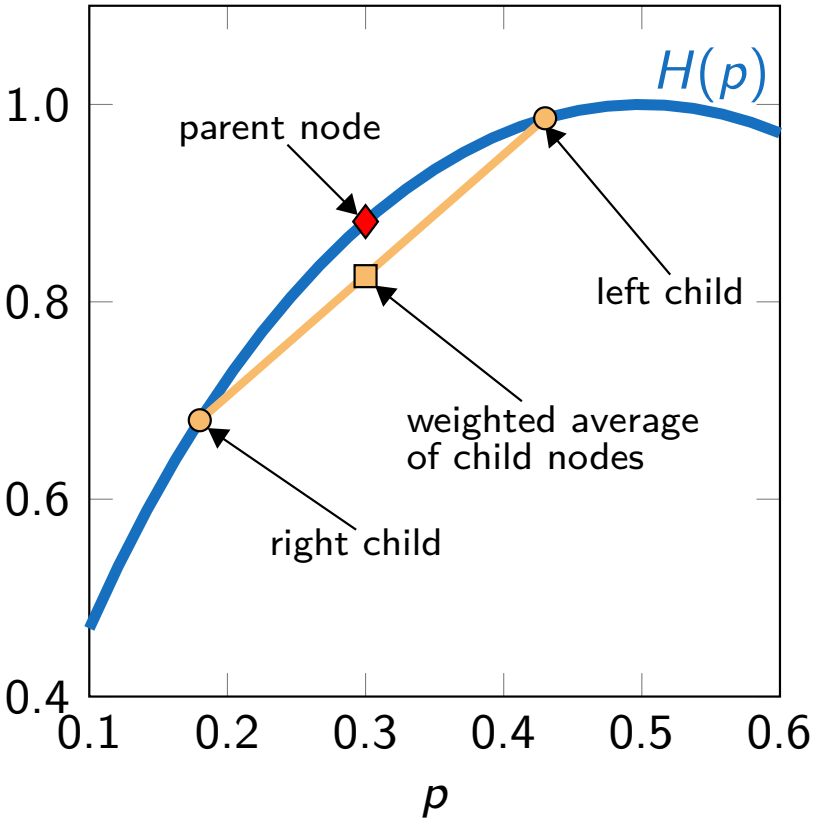
Information Gain (basierend auf Misclassification Error):

$$\frac{40}{120} - \frac{70}{120} \cdot \frac{28}{70} - \frac{50}{120} \cdot \frac{12}{50} = 0$$

Misclassification Error vs. Entropie

Eigenschaften der Entropiefunktion

- Die Entropiefunktion $H(p)$ ist strikt konkav..
- Der Graph verläuft stets über der Verbindungsgerade zwischen zwei Punkten.
- Die Entropie eines Knotens, der kein reines Sample repräsentiert, ist stets größer als die gewichtete Entropie der Kindknoten.
- Falls ein Sample S nicht rein ist, gilt also stets $Gain(S, A) > 0$.
- Gleiches gilt für die Gini Impurity, aber nicht für den Misclassification Error



Varianten des Entscheidungsbaum-Lernens

Ostbayerische Technische Hochschule Amberg-Weiden

ID3 (Iterative Dichotomizer 3)

- Einer der ersten Entscheidungsbaum-Algorithmen
- Veröffentlicht in Quinlan, J. R. (1986). Induction of decision trees. Machine learning, 1 (1), 81-106.
- Behandlung kategorischer Features
- keine kontinuierlichen Features
- kein Pruning (beschneidung des Baums)
- Maximierung des Information Gain bezüglich der Entropie

C4.5

- Erweiterung von ID3
- Veröffentlicht in Quinlan, J. R. (1993). C4.5: Programs for Machine Learning. Machine Learning, 1 (16), pp. 235-240.
- Berücksichtigung kontinuierlicher und kategorischer Features
- Berücksichtigung von fehlenden Werten
- Durchführung von Post-Pruning

Varianten des Entscheidungsbaum-Lernens

CART (Classification and regression tree)

- Breiman, L. et. al. (1984). Classification and regression trees. Belmont, California: Wadsworth International Group.
- Kontinuierliche und kategorische Features
- Ausschließlich binäre Splits: für ein kontinuierliches Feature A kann jede Ausprägung a_i einen möglichen Split definieren, indem die Subsamples

$$S_1 = \{Samples \ : \ A \geq a_i\} \text{ und } S_2 = \{Samples \ : \ A < a_i\}$$

  gebildet werden. Für ein kategorisches Feature A definiert jede Ausprägung a_i einen binären Split, bei welchem die Subsamples

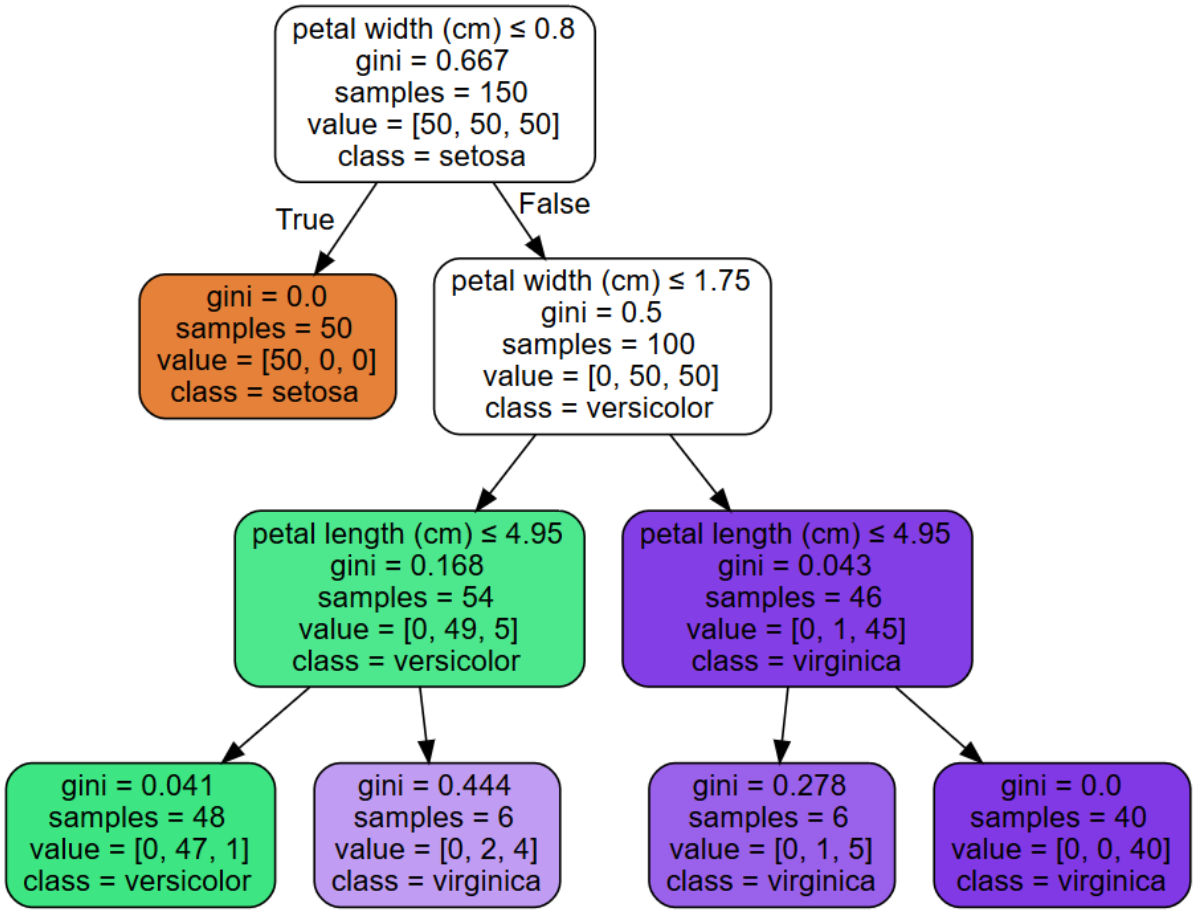
$$S_1 = \{Samples \ : \ A = a_i\} \text{ und } S_2 = \{Samples \ : \ A \neq a_i\}$$

  gebildet werden.
- Verwendung der Gini-impurity zur Merkmalsauswahl
- Tendenz zu tiefen und schmalen Bäumen
- Durchführung von Pre-Pruning

CART-Tree für den Iris-Datensatz

Entscheidungsbaum für den Iris-Datensatz:

- Pre-Pruning mit den Parametern `max_depth=3`, `min_samples_leaf=5`
- Ergebnis:



Decision Tree Learning als Greedy -Algorithmus

Ostbayerische Technische Hochschule Amberg-Weiden

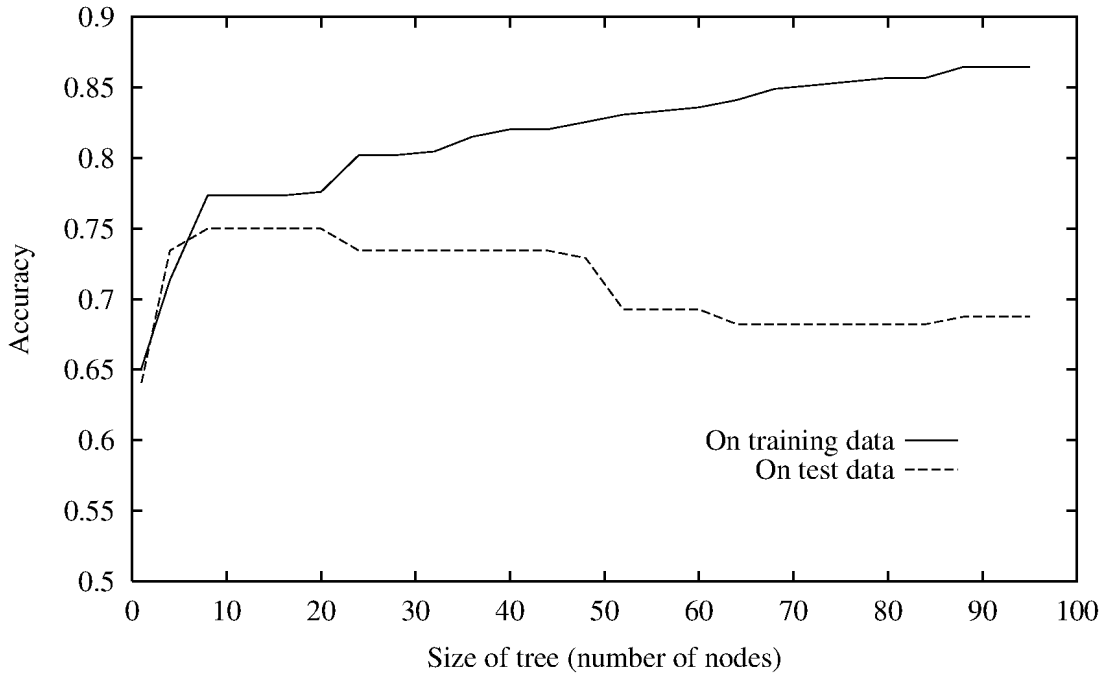
Entscheidungsbaum-Lernen als Greedy-Algorithmus

- Entscheidungsbaum-Lernen zeichnet sich durch „**greedy search**“ (gierige Suche) aus.
- Greedy-Algorithmen zeichnen sich dadurch aus, dass Sie in jedem Schritt den Folgezustand auswählen, der zum Zeitpunkt der Wahl den größten Gewinn bzw. das beste Ergebnis verspricht. Sie lösen Probleme häufig performant, aber nicht optimal.
- Der Hypothesenraum wird beim Entscheidungsbaum-Lernen unvollständig durchsucht (kein Backtracking), in jedem Schritt wird nur nach dem „nächstbesten“ Split durch Maximierung des Information Gain gesucht.
- „Ockhams[1] Rasiermesser“ (Prinzip der Sparsamkeit): bei der Modellerstellung werden kürzere Bäume gegenüber längeren Bäumen bevorzugt (preference bias).
- Wie könnte man das rechtfertigen?
  längere Bäume tendieren zu overfitting

[1] Wilhelm von Ockham (1288–1347)

Overfitting

- Falls keine widersprüchlichen Samples enthalten sind (gleiche Attribute, verschiedene Zielfunktionswerte; z.B. durch Rauschen, Unsicherheit), wird ein nach obigen Regeln erstellter Entscheidungsbaum alle Samples des Trainingsdatensatzes korrekt klassifizieren.
- Problem: typischerweise schlechte Verallgemeinerung auf neue Daten (Overfitting).



Quelle: T. Mitchell. Machine Learning. McGrawHill, 1997 (S. 67)

Maßnahmen gegen Overfitting

Maßnahmen gegen Overfitting:

- **Statistische Tests:** Vermeidung von Splits, die nicht statistisch signifikant sind.
- **Pre-Pruning:** Beschränkung der Größe des Baums, sodass er sich nicht perfekt an die Trainingsdaten anpasst, z.B. durch
  - Begrenzung der Tiefe des Baums ($\rightarrow$ Parameter `max_depth` in sklearn).
  - Festlegung einer Mindestanzahl an Datensätzen pro Blatt ($\rightarrow$ Parameter `min_samples_leaf` in sklearn).

  $\rightarrow$ zusätzliche Hyperparameter (wie festlegen?)
- **Post-Pruning:** Aufbau des vollständigen Baums und nachträgliche Entfernung von Knoten
  - Auswertung auf zusätzlichem Validierungs-Datensatz
  - Entfernung von Knoten, solange die Klassifikationsgüte auf dem Validierungs-Datensatz noch hinreichend ist

Weitere Aspekte

Umgang mit fehlenden Werten

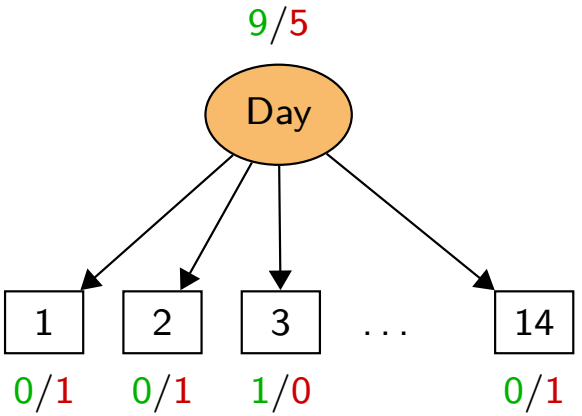
- Viele Machine Learning-Verfahren erfordern das Vorhandensein aller Werte für einen Datensatz. Bei fehlenden Werten (missing values) müssen Datensätze entweder verworfen werden oder es müssen Werte im Rahmen der Datenvorbereitung eingesetzt werden („Imputation“).
- Dadurch kann der Datensatz verfälscht werden.
- Entscheidungsbäume können mit fehlenden Werten umgehen, indem diese bei der Berechnung des Information Gain einfach nicht berücksichtigt werden.
- Entscheidungsbaum-Lernen erfordert in dieser Hinsicht weniger Datenvorverarbeitungsschritte als andere Methoden.

Umgang mit kategorischen und kontinuierlichen Features

- Entscheidungsbäume können sowohl mit kategorischen als auch mit kontinuierlichen/numerischen Features umgehen.
- Eine Transformation von kategorischen in numerische Features ist nicht zwingend erforderlich.

Weitere Aspekte

Kategorische Features mit vielen Ausprägungen



$$H(S) = -\frac{9}{14}\log_2\left(\frac{9}{14}\right) - \frac{5}{14}\log_2\left(\frac{5}{14}\right) \approx 0.94$$

$$Gain(S, Day) = 0.94 - 0 - 0 - \ldots - 0 = 0.94$$

$\rightarrow$ Split an Merkmal „Day“ und Abbruch (alle Subsamples rein)

- Beim ID3-Verfahren unter Verwendung des Information Gain werden Attribute mit vielen Ausprägungen bevorzugt.
- Besser: Verwendung des **Gain Ratio**:

$$SplitEntropy(S, A) = -\sum_{v \in Values(A)} \frac{|S_v|}{|S|}\log_2\left(\frac{|S_v|}{|S|}\right) ,$$

$$GainRatio(S, A) = \frac{Gain(S, A)}{SplitEntropy(S, A)} .$$

Regression Trees

- Entscheidungsbaum-Lernen kann auch zur Regression, d.h. zur Vorhersage kontinuierlicher Werte, verwendet werden.
- Ein Blatt liefert den Wert als Prognose zurück, der sich durch Mittelwertbildung aller Zielfunktionswerte der Trainings-Samples in diesem Blatt ergibt.
- Zum Aufbau des Baums wird statt des Information Gain die **Standardabweichung** verwendet:

$$I(S) = \frac{1}{N_S} \sum_{i \in S} \left(y^{(i)} - \hat{y}_S \right)^2,$$

wobei N_S die Anzahl der Datensätze und $\hat{y}_S = \frac{1}{N_S} \sum_{i \in S} y^{(i)}$ den Mittelwert der Zielfunktionswerte im Sample S des Trainingsdatensatzes bezeichnet.

Einordnung von Entscheidungsbaum-Lernen

Ostbayerische Technische Hochschule Amberg-Weiden

Batch- vs. Online Learning

- Machine Learning-Verfahren werden hinsichtlich der Verarbeitung der Daten beim Modelltraining unterschieden.
- Beim **Batch Learning** wird das Modell anhand des gesamten Trainingsdatensatzes erstellt.
- Beim **Online Learning** hingegen kann das Modell anhand einzelner Trainingsdatensätze trainiert oder modifiziert/aktualisiert werden.
- Decision Trees sind ein Beispiel für Batch Learning.

Einordnung von Entscheidungsbaum-Lernen:

Einordnung von Entscheidungsbaum-Lernen

Batch- vs. Online Learning

- Machine Learning-Verfahren werden hinsichtlich der Verarbeitung der Daten beim Modelltraining unterschieden.
- Beim **Batch Learning** wird das Modell anhand des gesamten Trainingsdatensatzes erstellt.
- Beim **Online Learning** hingegen kann das Modell anhand einzelner Trainingsdatensätze trainiert oder modifiziert/aktualisiert werden.
- Decision Trees sind ein Beispiel für Batch Learning.

Einordnung von Entscheidungsbaum-Lernen:

- Supervised Learning (zur Klassifikation und Regression)
- Eager Learning
- Nicht-parametrisiertes Verfahren
- Batch-Learning
- Greedy-Algorithmus

Vor- und Nachteile von baumbasierten Ansätzen

Vorteile:

- Geeignet für numerische und kategorische Daten
- Geeignet für Klassifikation und Regression
- Geringer Aufwand für Datenvorbereitung
- Hohe Nachvollziehbarkeit („White Box - Modell“)
- Einfach zu implementieren (rekursiv!)
- Basisverfahren für Ensemble-Methoden (z.B. Random Forest)

Nachteile:

- Geringe Stabilität (kleine Änderungen in den Daten können zu großen Unterschieden im Ergebnis führen)
- Anfällig für Overfitting (Übertraining)
- Greedy-Algorithmus

Übung

Ziele der Übung

- Training von Entscheidungsbäumen in Scikit-learn
- Vorbereitung der Daten mit pandas
- Visualisierung eines Entscheidungsbaums mit graphviz
- Implementierung eines CART-Trees „from scratch“