Applied Machine Learning in Engineering

Lecture 06, May 23, 2023

Prof. Merten Stender

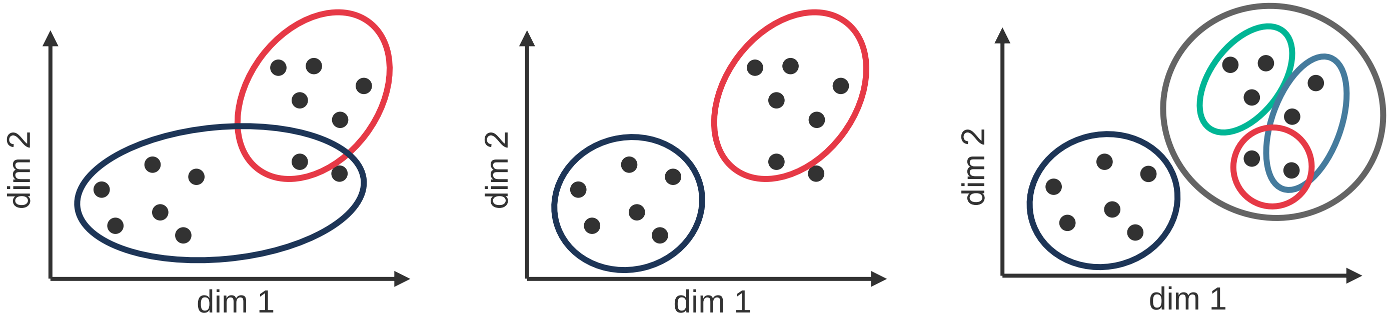
Cyber-Physical Systems in Mechanical Engineering, Technische Universität Berlin

www.tu.berlin/cpsme
merten.stender@tu-berlin.de

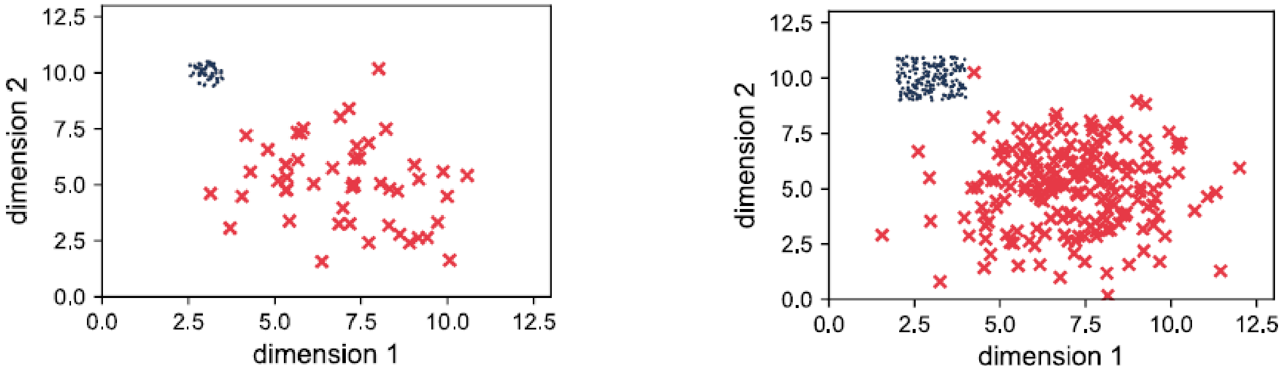
Recap: Lecture 05

- **Unsupervised learning**: finding data groups of similar characteristics

- **Types of clusterings**
  1. Nesting
  2. Exclusiveness
  3. Completeness



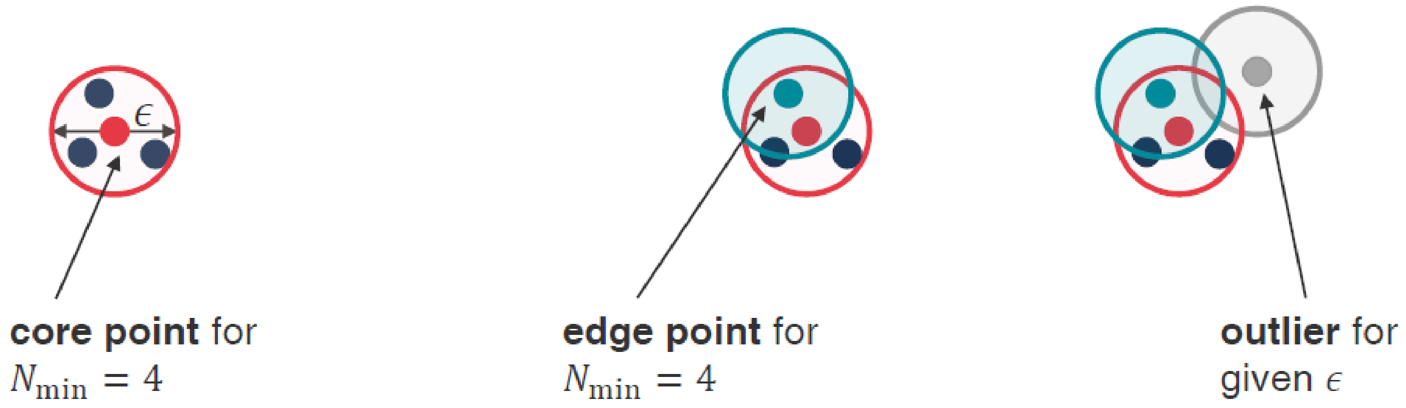
- **Types of clusters**
  1. Distribution
  2. Density
  3. Size or variance



Recap: Lecture 05

- **Unsupervised learning**: finding data groups of similar characteristics

- **DBSCAN clustering**
  - Algorithm and 3 types of points: core points, edge points, outliers
  - Importance of data normalization
  - Limitations (clusters of very different density)



Recap: Exercise 05

- From scratch implementation of Z-scoring
- initialization
- .fit method: calculate mean and std. deviation of the given data set
- .transform: z-scoring operation
- .inverse_transform: reverse z-scoring operation

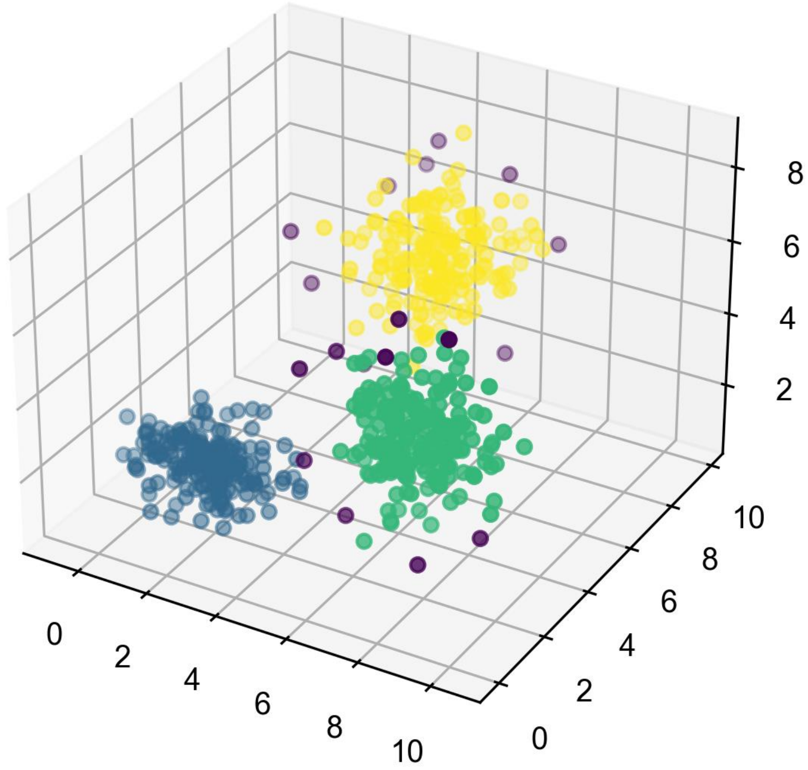
```
class Zscorer():

    def __init__(self):
        self.mean: float
        self.sigma: float

    def fit(self, data):
        self.data = data
        num_cols = data.shape[1]
        mean = sum(self.data)/len(self.data)
        self.mean = np.ones((len(self.data), num_cols))*mean
        self.sigma = np.sqrt(sum((self.data-self.mean)**2)/(len(self.data)-1))

    def transform(self):
        self.data = (self.data - self.mean)/self.sigma
        return (self.data)

    def inverse_transform(self):
        self.data = self.sigma*self.data + self.mean
        return (self.data)
```

Recap: Exercise 05

- Clustering a 3-dim. data set

- Suggestion for method / hyperparameters:
  - Data normalization using z-scoring
  - DBSCAN with $N_{\text{min}} = 3$ and ϵ =1

- Cluster validity:
  - Silhouette coefficient $S = 0.632$



Today

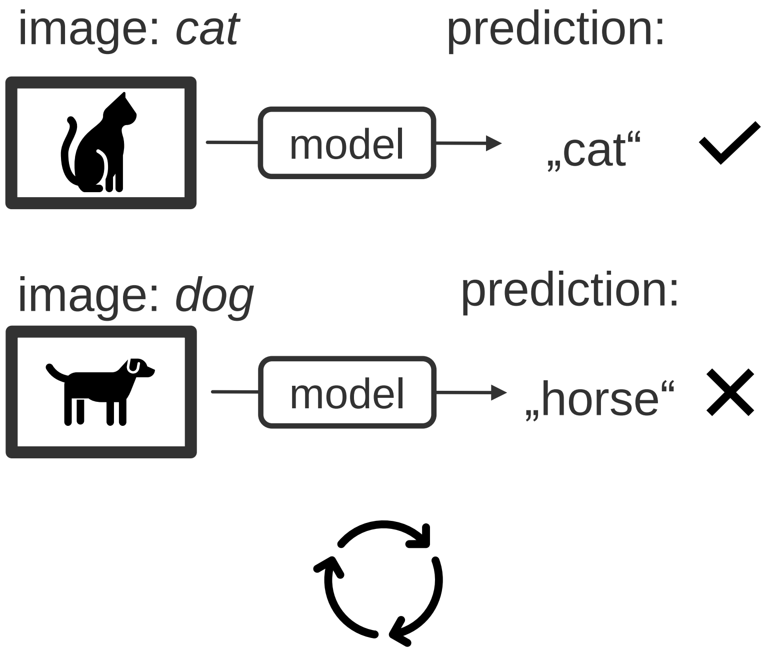
- Recognize supervised learning tasks
- Understand tree-based decision models
- Understand measures of class purity

Agenda

Cyber-Physical Systems
in Mechanical Engineering TU Berlin

Machine Learning:

- Supervised learning
- Introduction to decision trees
- Information entropy as generic concept



Python:

X-Student Research Groups

Cyber-Physical Systems
in Mechanical Engineering TU Berlin

- Research teams of 15 students (BUA) and young researchers
- Seminar (6 ECTS, free choice modules) for one semester (winter 23)

more info

Proposal for Research Group: **Physical Reservoir Computing**

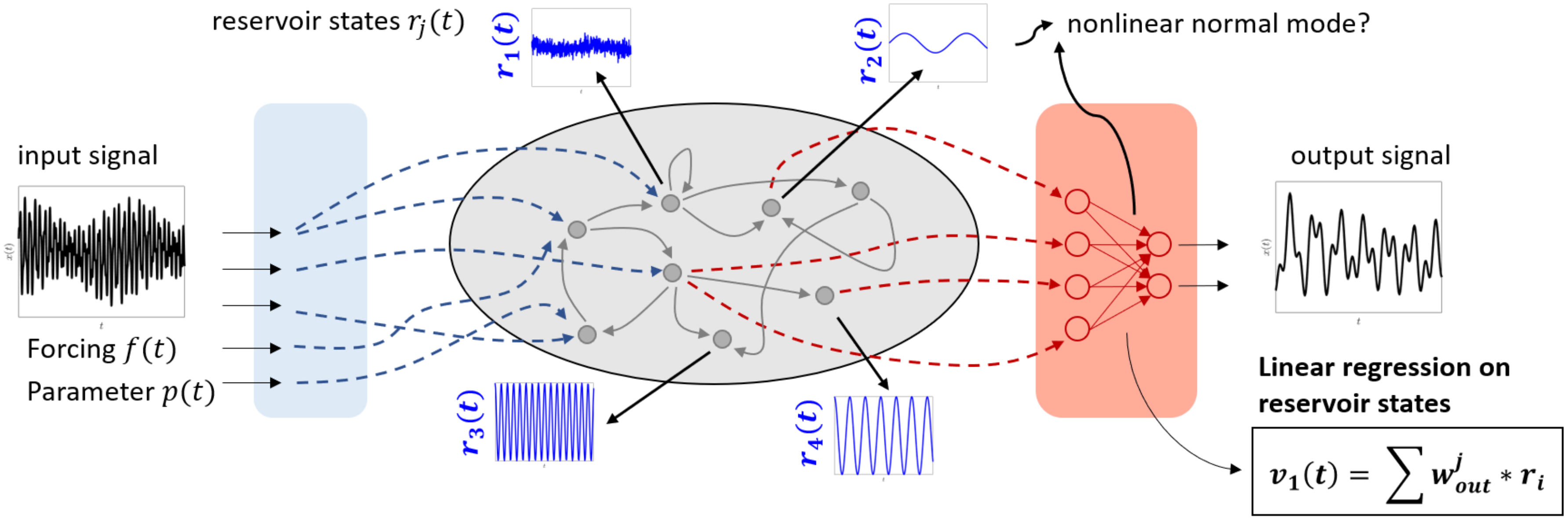
Use a bucket of water for building a machine learning computer

- **Build a demonstrator** (electronics, micro-controllers, computer vision, coding, ML)
- Show a **proof of concept** for time series prediction or natural language processing
- **Present results** at scientific conference or publish scientific paper

X-Student Research Groups

Cyber-Physical Systems
in Mechanical Engineering TU Berlin

- Coordinator in CPSME group: Dr. Manish Yadav



- **Spread the word!**
- Interested? Write a short mail to merten.stender@tu-berlin.de

Supervised Learning

Supervised Learning

Supervised learning = fitting prediction models to data for which ground truth targets exist

$$\mathcal{M}_{\boldsymbol{\theta}}\colon\ \mathbf{X} \mapsto \mathbf{y}\ ,\ \mathbf{X} \in \mathbb{R}^{N\times n},\ \mathbf{y} \in \mathbb{R}^{N\times m}$$

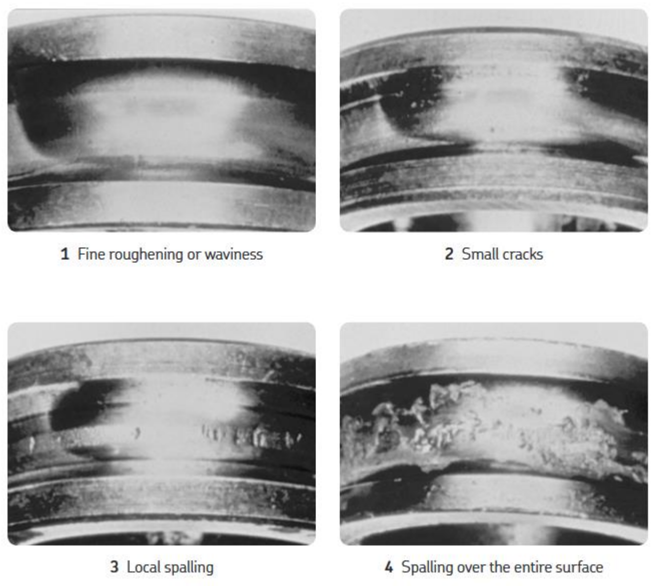
- **Ground truth** data (‘*labels*’)
  - Desired target quantities y_i
  - → Allows comparing y_i against model predictions $\hat{y}_i$, prediction error $E = \|\boldsymbol{y} - \hat{\boldsymbol{y}}\|$
  - → Quantitative statements about model prediction quality

- **Model fitting**:
  - Reduction of error on training data set $\min_{\boldsymbol{\theta}} E(D_{\text{train}},\ \boldsymbol{\theta})$ through optimization of $\boldsymbol{\theta}$
  - Model validation on hold-out validation data set D_{val}
  - Under- and overfitting as potential issues

- **Classification** and **regression tasks**

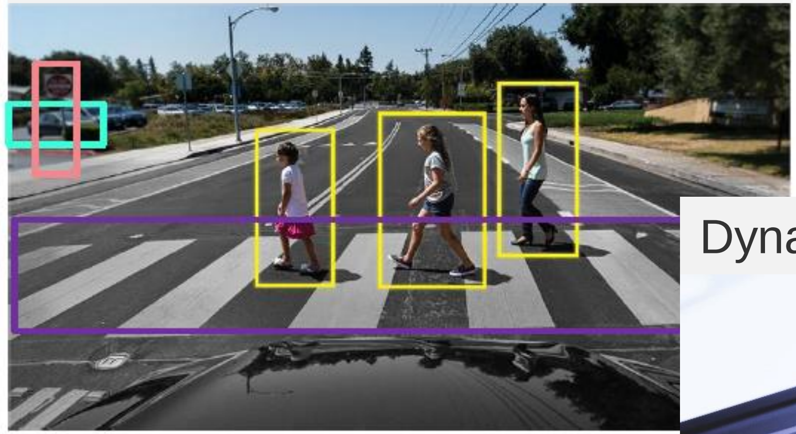
Application Cases in Engineering

Structural Health Monitoring
Predictive Maintenance
Remaining life time prediction



SKF® Bearing damage and failure analysis

Computer vision



Dynamical behavior prediction

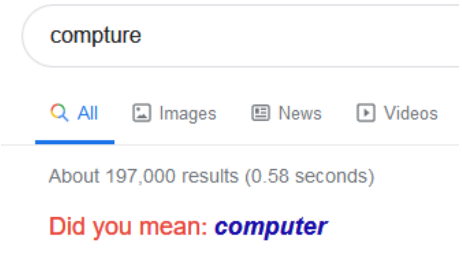
… and many more

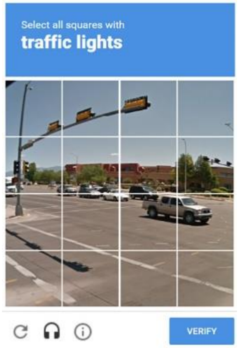
Generation of Targets

Cyber-Physical Systems
in Mechanical Engineering TU Berlin

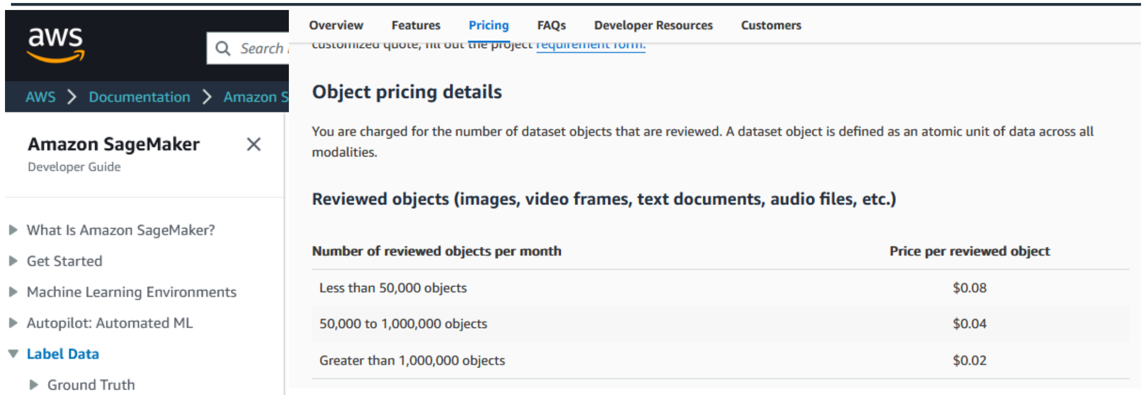
- **Extremely important** (‚*trash in – trash out*'), yet tedious and expensive
  - Correct and high-quality ground truth targets are of crucial importance
  - Less but high-quality data should always be preferred over large and less-quality data

- Creative ways to generate labels:
  - Did you mean … ? → grammar / language models
  - reCAPTCHA - *are you a robot?* → computer vision





- Professional data labeling services



Amazon, 06 April 2023

Decision Trees

Decision Trees

input features $\boldsymbol{x} = [x_1, x_2, x_3]$

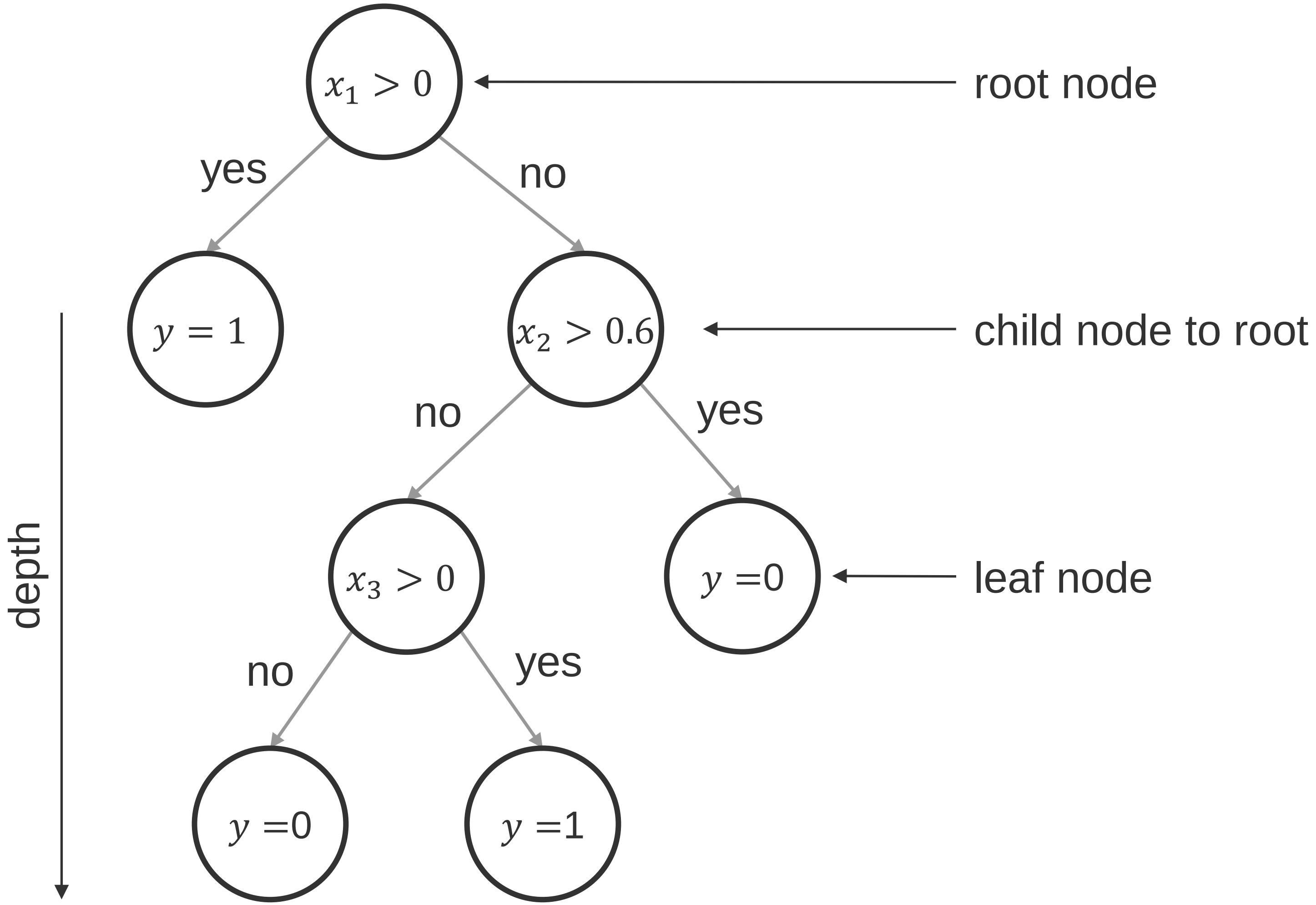
x_1: sun is shining: {0, 1}

x_2: probability of rain: [0,1.0]

x_3: ambient temperature: [-20, 40] °C

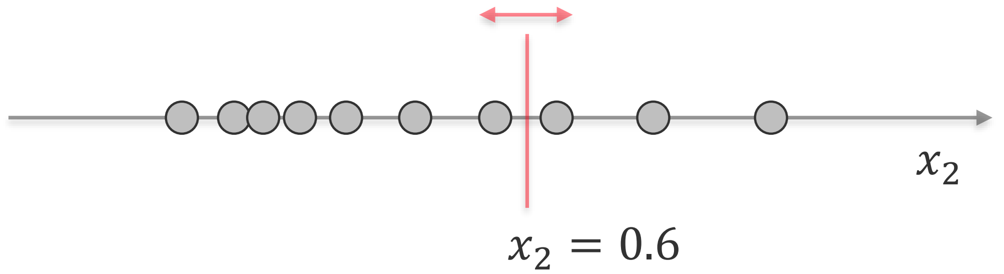
target y

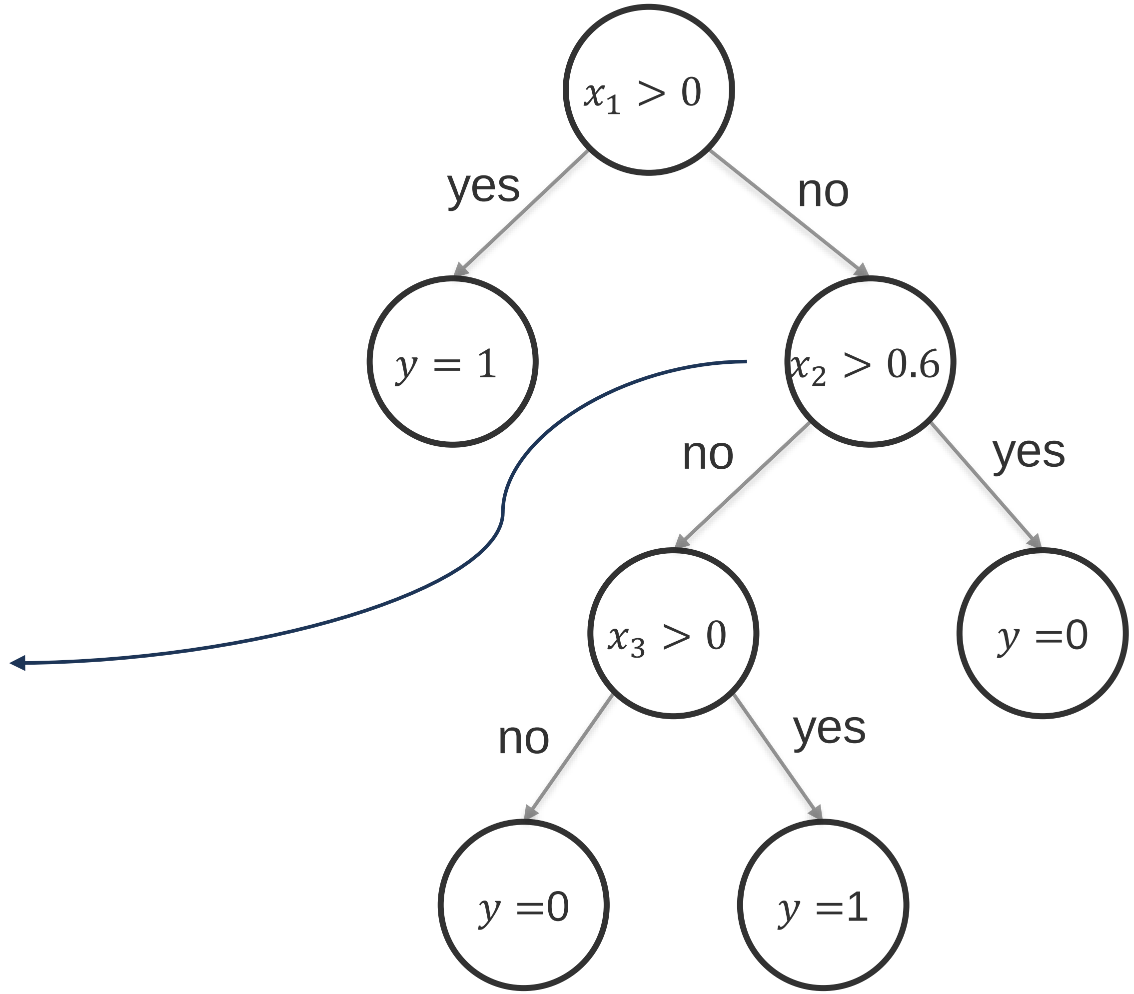
y: ride bike to work? {0, 1}



Decision Trees

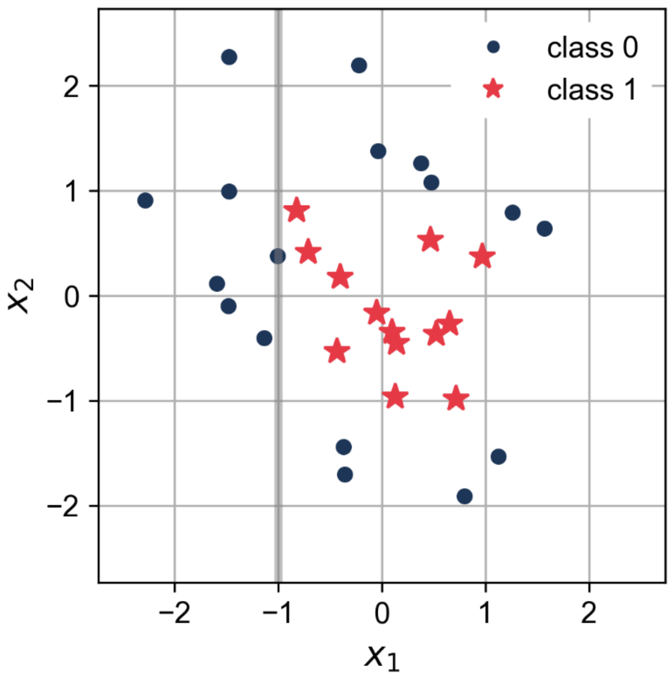
- Sequential decision rules
- Greedy algorithm
  - Previous splits not affected by current split
  - Algorithm does not ‘look ahead‘
- Recursive binary **feature space segmentation**





Decision Trees: Example

- Aim: classify 2-dimensional data set with two classes (binary classification task)



Which split to do?

- Feature dimension (x_1, x_2)?
- Feature value (x^*) ?

- Split condition 1: $x_1 > -1.0$

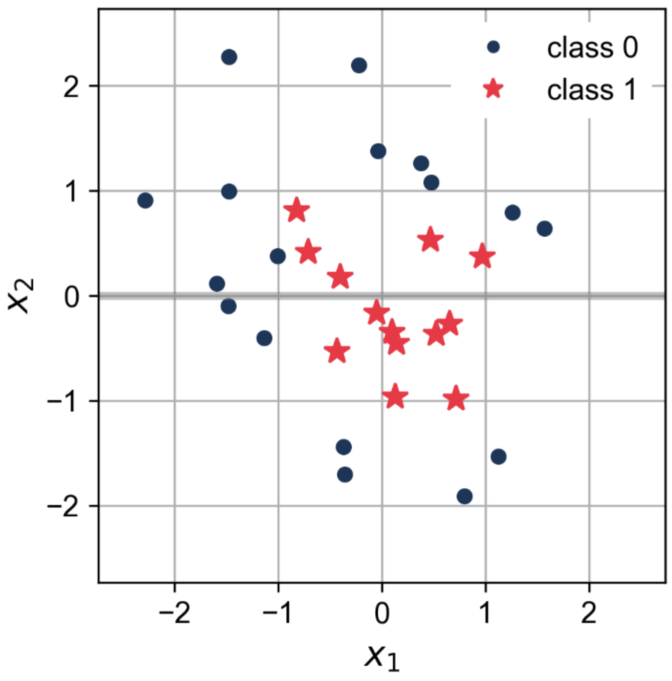
child 1 (condition false)
$7 \times$ ●
$0 \times$ ★

child 2 (condition true)
$10 \times$ ●
$13 \times$ ★

Decision Trees: Example

Cyber-Physical Systems
in Mechanical Engineering TU Berlin

- Aim: classify 2-dimensional data set with two classes (binary classification task)



Which split to do?

- Feature dimension (x_1, x_2)?
- Feature value (x^*) ?

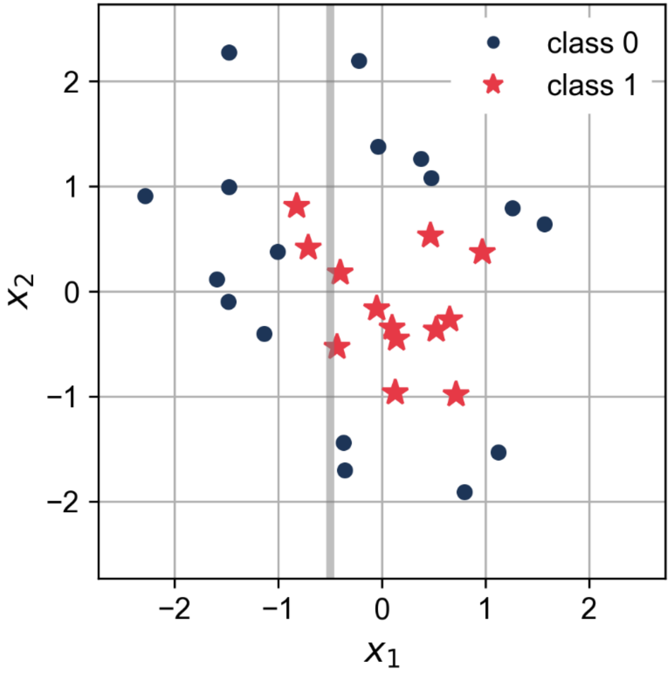
- Split condition 2: $x_2 > 0$

child 1 (condition false)
$6\times$ ●
$8\times$ ★

child 2 (condition true)
$11\times$ ●
$5\times$ ★

Decision Trees: Example

- Aim: classify 2-dimensional data set with two classes (binary classification task)



Which split to do?

- Feature dimension (x_1, x_2)?
- Feature value (x^*) ?

- Split condition 3: $x_1 > -0.5$

child 1 (condition false)

$7\times$ ●

$2\times$ ★

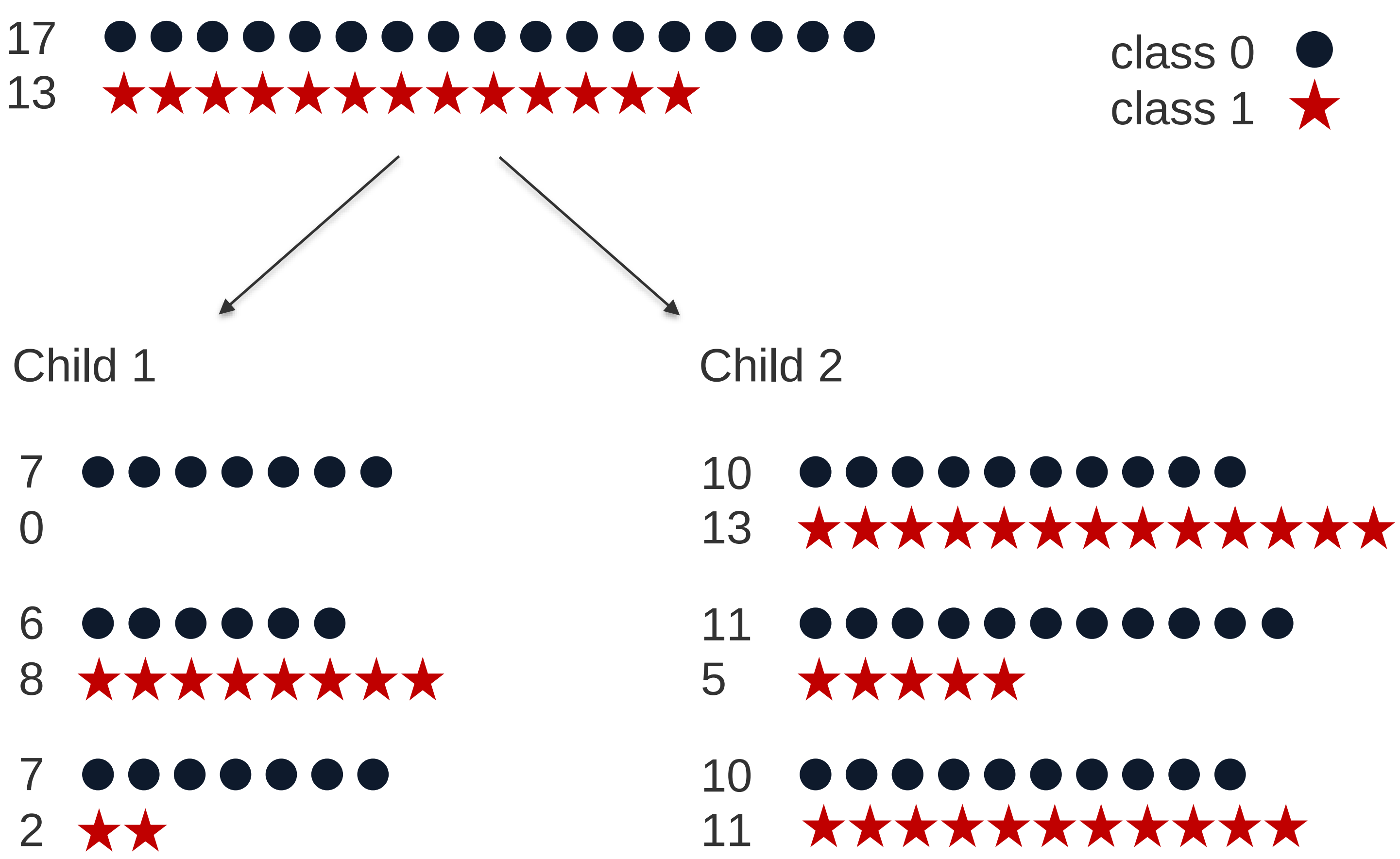
child 2 (condition true)

$10\times$ ●

$11\times$ ★

Selecting the Best Split

- Full data set
- Data split
  - Which to select?

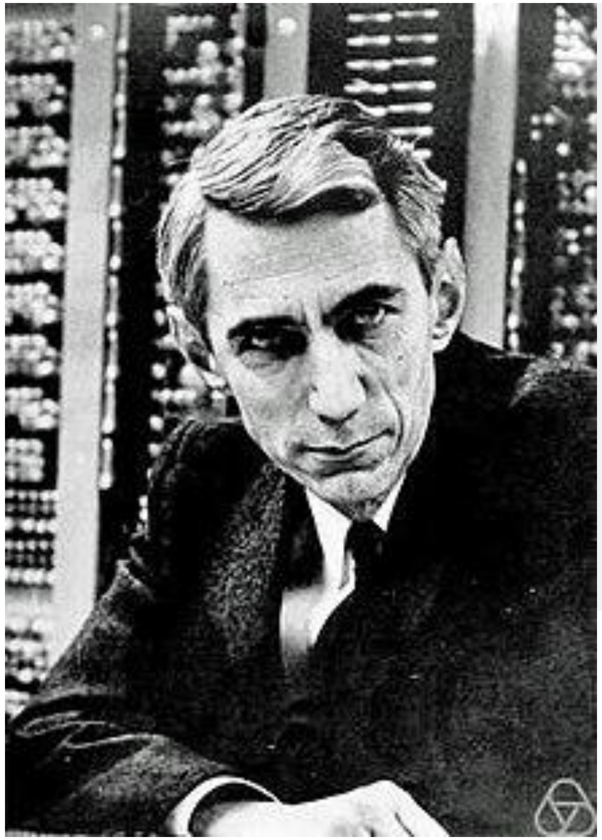


$$H(x) = -\sum_{i \in C} P(x_i) \log P(x_i)$$

Claude E. Shannon (1916-2001)

Founder of information theory

1948: A Mathematical Theory of Communication



Shannon Entropy: Definition

Cyber-Physical Systems
in Mechanical Engineering TU Berlin

- Also denoted *information entropy* or *entropy index*

$$H(x) = -\sum_{i \in C} P(x_i) \log_a(P(x_i)) = \sum_{i \in C} P(x_i) \log_a\left(\frac{1}{P(x_i)}\right) \in [0,]$$

- $C = \{c_1, c_2, c_3\}$ set of distinct classes

- $P(x_i)$ probability of a single event *i:*
  - fraction of population composed of a single species i

- $H(x)$ amount of information gained by observing an event of probability $P(x_i)$

a base of the logarithm

The unit of entropy depends on the base of the logarithm

- Computer science: $a = 2$
  → unit *bits*

- Euler‘s number: $a = e$
  → unit *nats*

Shannon Entropy: Intuition

Cyber-Physical Systems
in Mechanical Engineering TU Berlin

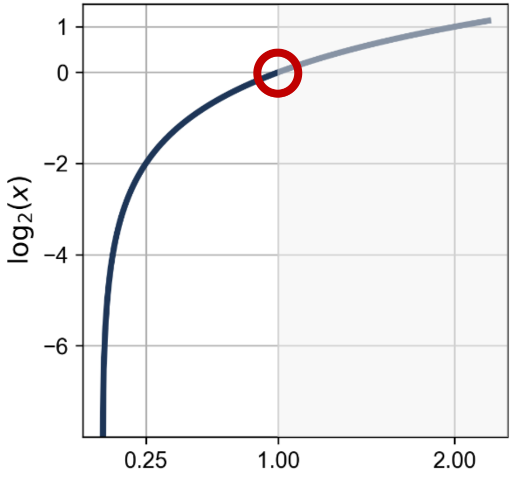
- Intuition and edge cases

$$H(x) = -\sum_{i \in C} P(x_i) \cdot \log_2(P(x_i))$$

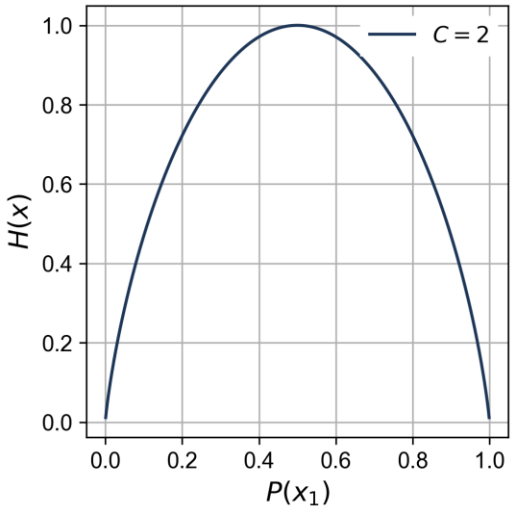
$C = 2$ (binary classification): $P(x_1) + P(x_2) = 1$

$$H(x) = -P(x_1) \cdot \log_2\big(P(x_1)\big) - P(x_2) \cdot \log_2\big(P(x_2)\big)$$
$$H(x) = -P(x_1) \cdot \log_2\big(P(x_1)\big) - (1 - P(x_1)) \cdot \log_2\big(1 - P(x_1)\big)$$

Log-2



$x = P(x_i)$



- $P(x_i) = 1 \rightarrow$ entropy$= 0$
- $\max\big(H(x)\big)_{C=2} = 1.0$ for $P(x_1) = P(x_2)$

Arbitrary C:

- $\max\big(H(x)\big)_C = -C \cdot \left(\frac{1}{C}\log\frac{1}{C}\right) = -\log\frac{1}{C}$

Shannon Entropy: Example

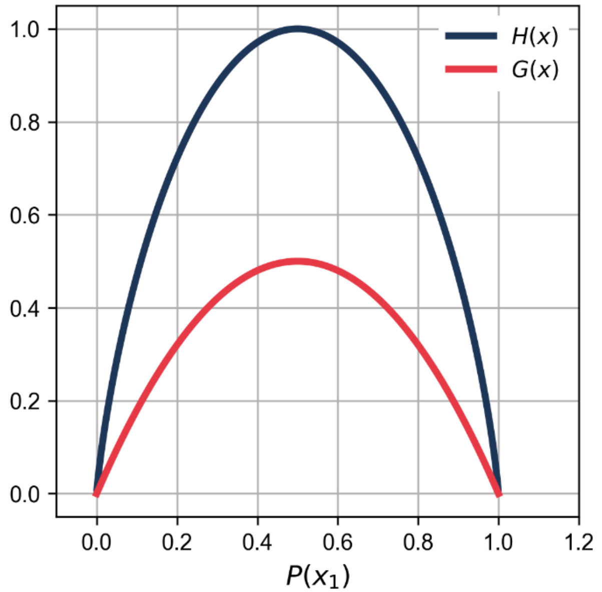
$$H(x) = -\sum_{i \in C} P(x_i) \log_2\big(P(x_i)\big)$$

- Example: **calculate** the uncertainty coming with a certain character appearing next
  - sequence of numbers $[2\ 3\ 0\ 2\ 7\ 1\ ]$
  - probabilities: $P(0) = 1/6,\ P(1) = 1/6,\ P(2) = 2/6,\ P(3) = 1/6,\ P(7) = 1/6$
  - entropy $H(x) = 2.2516$

- Edge case 1: Outcome is certain: vanishing entropy $H(x) = 0$, e.g. $[2\ 2\ 2\ 2\ 2\ 2\ ]$
- Edge case 2: The more proportial the frequencies of occurance are, the harder it gets to make a prediction, hence the larger the entropy $H(x) \gg 0$, e.g. $[1\ 2\ 3\ 4\ 5\ 6\ ]$

Measuring purity of a population

- Information entropy $H(X) = -\sum_{i=1}^{K} P_i \log_2(P_i)$ P_i : probability of class i out of K classes
- Gini index $G(X) = 1 - \sum_{i \in C} P(x_i)^2$



- Cases:
  - All members of the data set belong to a single class

    10 ×● , 0 ×★ $H(X) = -\frac{10}{10} \cdot \log\frac{10}{10} - \frac{0}{10} \cdot \log\frac{0}{10} = 0 - 0 = 0$

  - Even distribution of members per class

    5 ×● , 5 ×★ $H(X) = -\frac{5}{10} \cdot \log\frac{5}{10} - \frac{5}{10}\log\frac{5}{10} = 0.5 + 0.5 = 1$

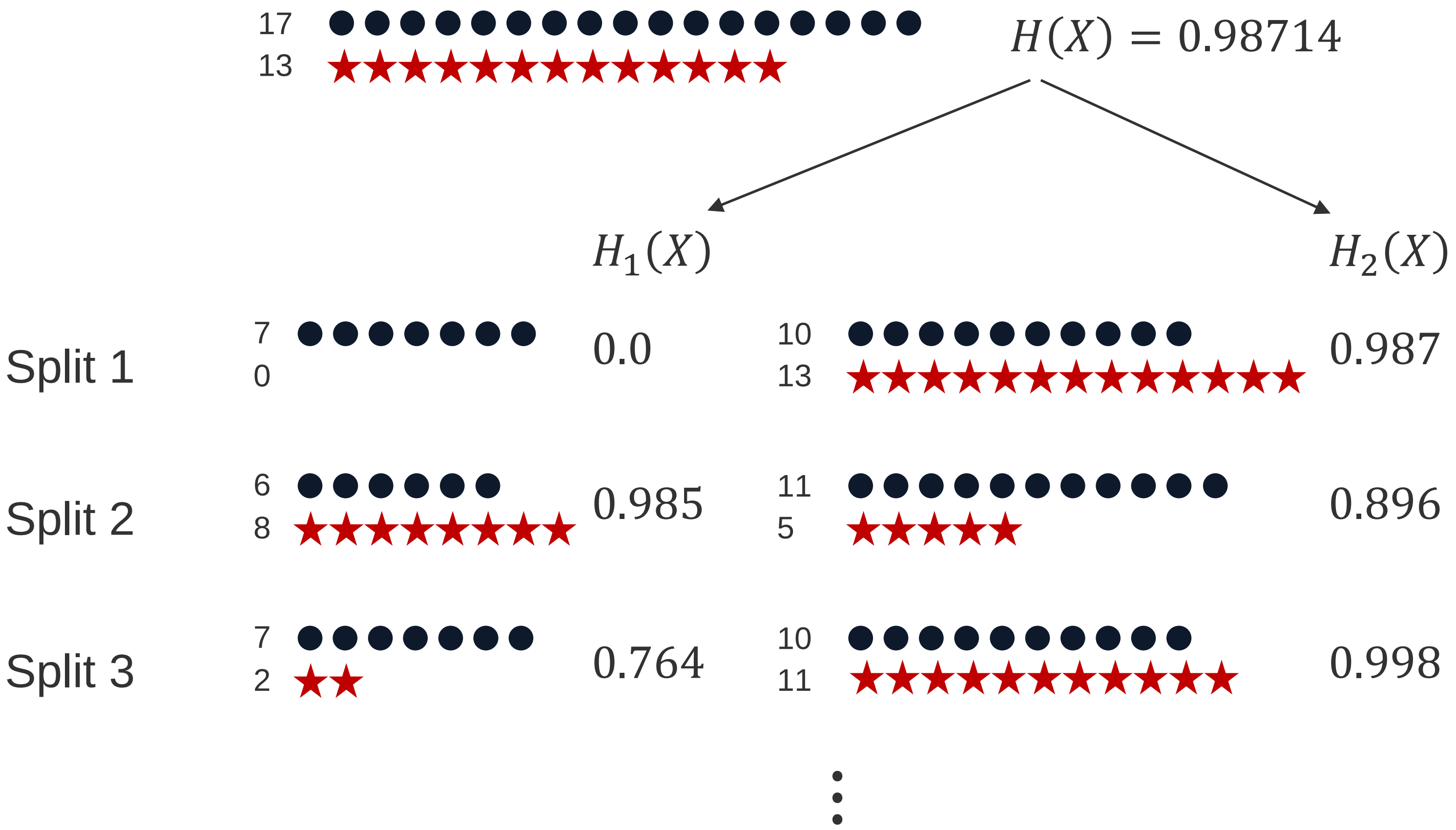
- Our data set at root:

17 ●●●●●●●●●●●●●●●●●
13 ★★★★★★★★★★★★★

$$H(X) = -\frac{17}{30}\log\frac{17}{30} - \frac{13}{30}\log\frac{13}{30} = 0.98714$$

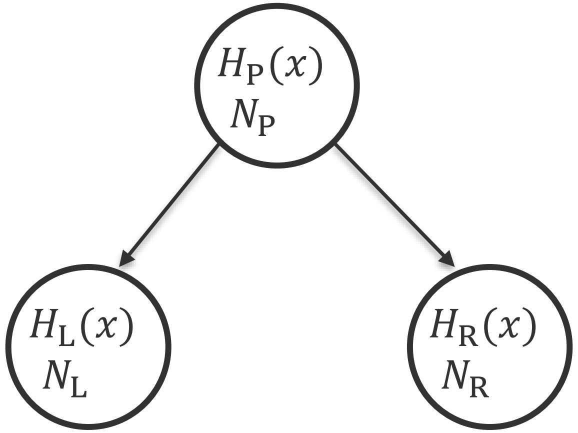
Selecting the best split

- Root (parent) data set
- **Compute** the entropy of every possible data split
- What now? Which split to select?
  → **Information gain**



Information Gain

- DT: segmentation of the feature space
- Aim: maximal class purity of the sub-data set
- Purity metric: entropy
  - Entropy at parent node: $H_{\mathrm{p}}(x)$
  - Entropy at children: $H_{\mathrm{L,\,R}}(x)$
  - Number of samples: $N_{\mathrm{P,\,L,\,R}}$



- Information gain $I(x) = H_{\mathrm{P}}(x) - \left(\frac{N_{\mathrm{L}}}{N_{\mathrm{P}}} \cdot H_{\mathrm{L}}(x) + \frac{N_{\mathrm{R}}}{N_{\mathrm{P}}} \cdot H_{\mathrm{R}}(x)\right)$

- Split for maximum information gain $x^* = \max(I(x))$

Best Split: Information Gain

17 ●●●●●●●●●●●●●●●●●
13 ★★★★★★★★★★★★★

$H(X) = 0.98714$

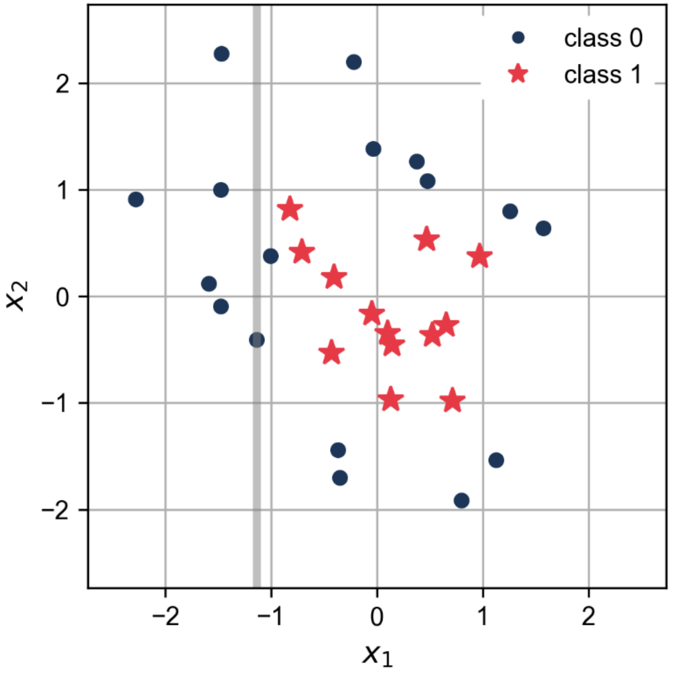
Information gain

$$I(x) = H_P(x) - \frac{1}{2}(H_1(x) + H_2(x))$$

		$H_1(X)$		$H_2(X)$	
Split 1	7 ●●●●●●● 0	0.0	10 ●●●●●●●●●● 13 ★★★★★★★★★★★★★	0.987	$I(x)$ =0.230
Split 2	6 ●●●●●● 8 ★★★★★★★★	0.985	11 ●●●●●●●●●●● 5 ★★★★★	0.896	$I(x)$ =0.049
Split 3	7 ●●●●●●● 2 ★★	0.764	10 ●●●●●●●●●● 11 ★★★★★★★★★★★	0.998	$I(x) = 0.059$

Decision Trees: Example

- Previous slides: 3 examplary data splits
- Actual: 60 splits possible (30 for feature dimension 1, 30 for feature dimension 2)

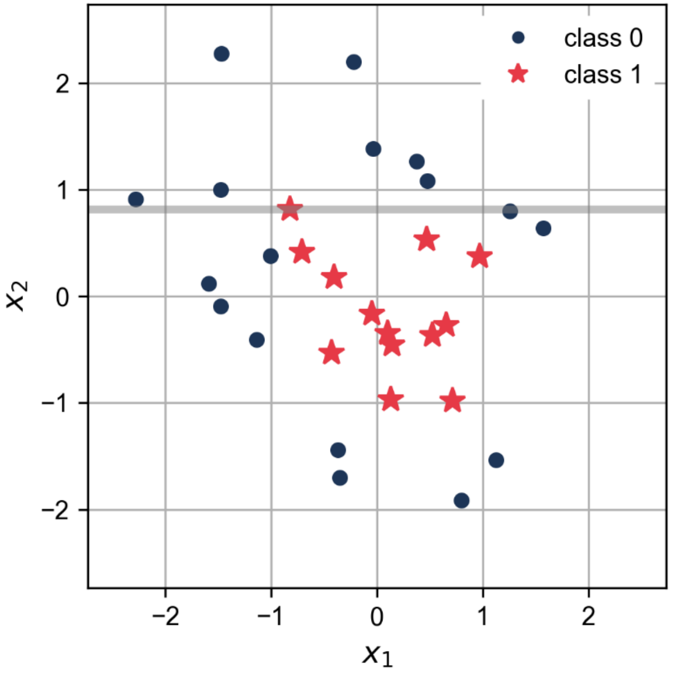


Greedy approach:

- Computation of all information gains
- Top 3 splits:
  - (3) condition: $x_1 \leq -1.137$ → information gain $I(x) = 0.191$

Decision Trees: Example

- Previous slides: 3 examplary data splits
- Actual: 60 splits possible (30 for feature dimension 1, 30 for feature dimension 2)

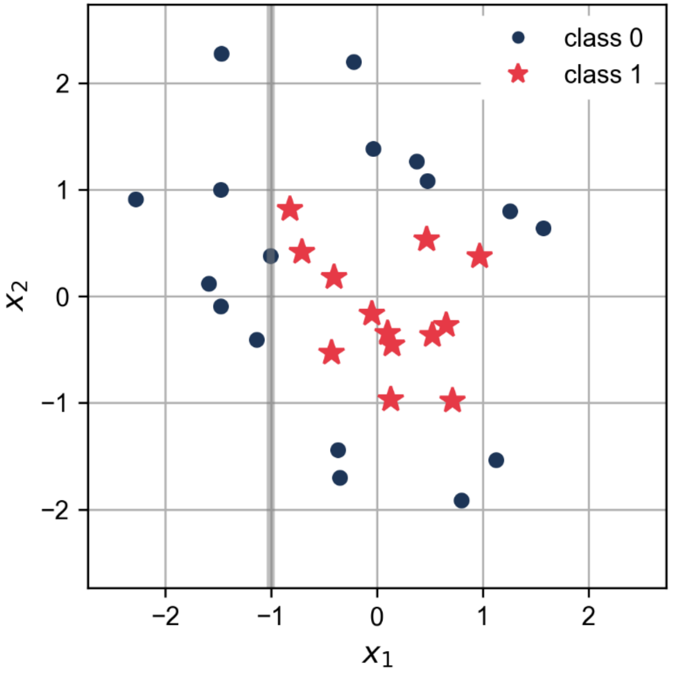


Greedy approach:

- Computation of all information gains
- Top 3 splits:
  - (3) condition: $x_1 \leq -1.137$ → information gain $I(x) = 0.191$
  - (2) condition: $x_2 \leq 0.813$ → information gain $I(x) = 0.229$

Decision Trees: Example

- Previous slides: 3 examplary data splits
- Actual: 60 splits possible (30 for feature dimension 1, 30 for feature dimension 2)



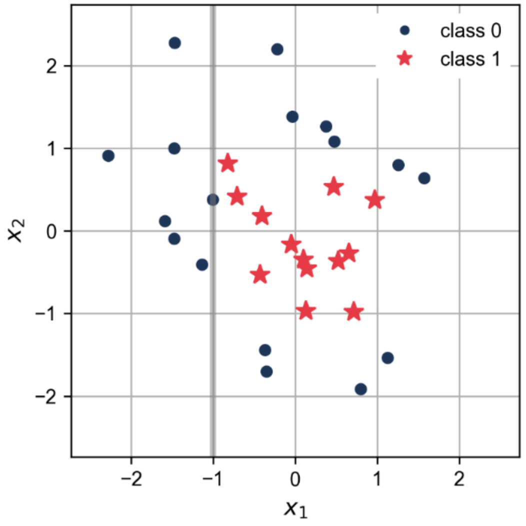
Greedy approach:

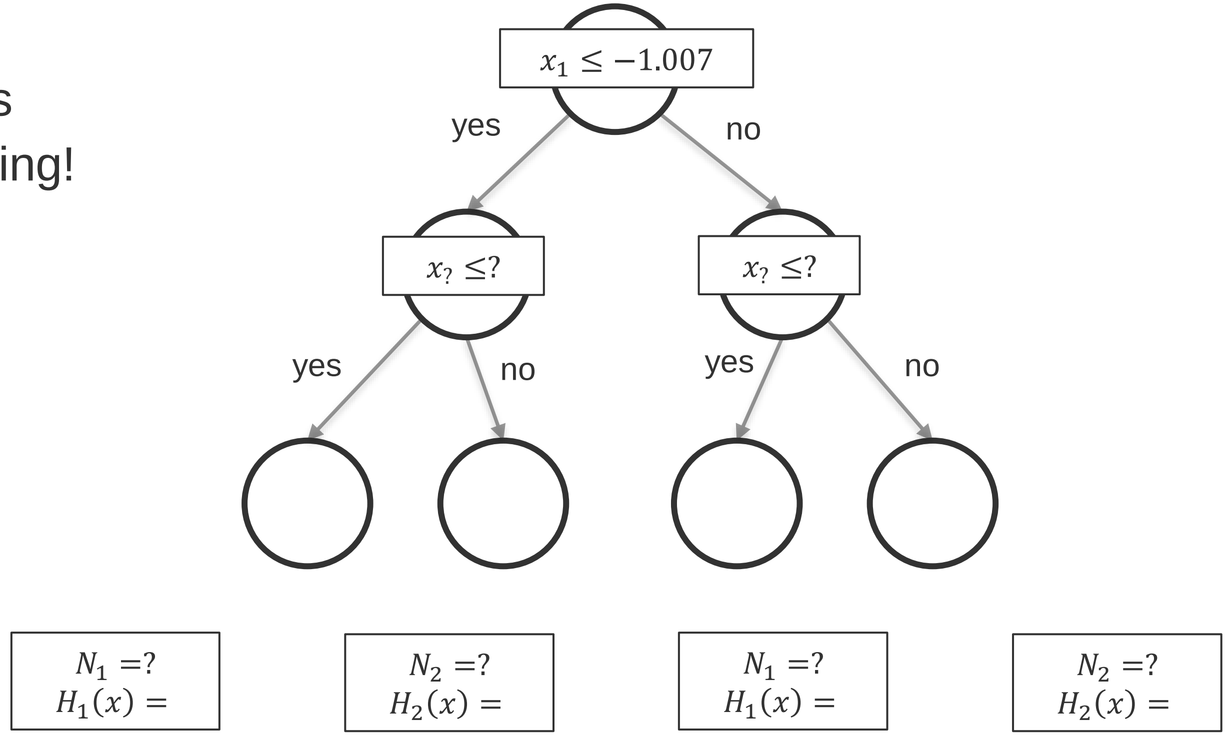
- Computation of all information gains
- Top 3 splits:
  - (3) condition: $x_1 \leq -1.137$ → information gain $I(x) = 0.191$
  - (2) condition: $x_2 \leq 0.813$ → information gain $I(x) = 0.229$
  - (1) condition: $x_1 \leq -1.007$ → information gain $I(x) = 0.229$

Decision Trees: Example

- **First split** of the data set:
  - Left child node: $x_1 \leq -1.007$ — 7 members of class 0
  - Right child node: $x_1 > -1.007$ — 10 members of class 0, 13 members of class 1

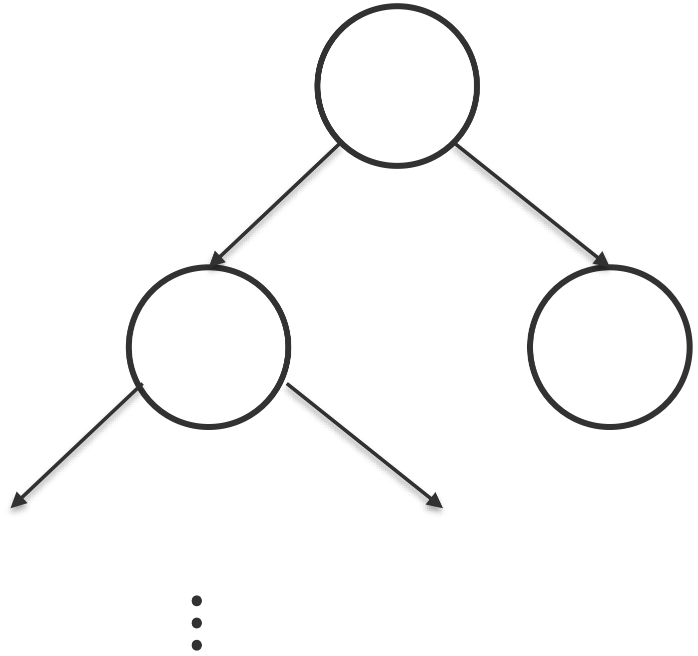
- **Second split**:
  - Optimal (information gain) split of child nodes
  - Here: left child node is pure → no more splitting!





Stopping Criteria

- Aim: maximize purity in leaf nodes
- Without any constrains: there is a solution with $H(x) = 0$ in each leaf node
  - Worst case: $N = 1$ samples per leaf
  - Overfitting the data
- Exessive splitting leads to **overfitting**:
  - Very strong performance on training data set
  - Weak performance on new (unseen) data
- **Constraints** to tree growing
  - Minimum number of samples per node N_{min}
  - Maximum depth of tree D_{max}
- **Impure leafs**: return the most-common class label



scikit-learn: DecisionTreeClassifier

Install User Guide API Examples Community More Go

Prev Up Next

scikit-learn 1.2.2
Other versions

Please **cite us** if you use the software.

sklearn.tree.DecisionTreeClassifier
DecisionTreeClassifier
Examples using sklearn.tree.DecisionTreeClass

sklearn.tree.DecisionTreeClassifier

class `sklearn.tree.`**`DecisionTreeClassifier`**(**, criterion='gini', splitter='best', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features=None, random_state=None, max_leaf_nodes=None, min_impurity_decrease=0.0, class_weight=None, ccp_alpha=0.0*) [source]

A decision tree classifier.

Read more in the User Guide.

Parameters:

criterion : *{"gini", "entropy", "log_loss"}, default="gini"*
The function to measure the quality of a split. Supported criteria are "gini" for the Gini impurity and "log_loss" and "entropy" both for the Shannon information gain, see Mathematical formulation.

splitter : *{"best", "random"}, default="best"*
The strategy used to choose the split at each node. Supported strategies are "best" to choose the best split and "random" to choose the best random split.

max_depth : *int, default=None*
The maximum depth of the tree. If None, then nodes are expanded until all leaves are pure or until all leaves contain less than min_samples_split samples.

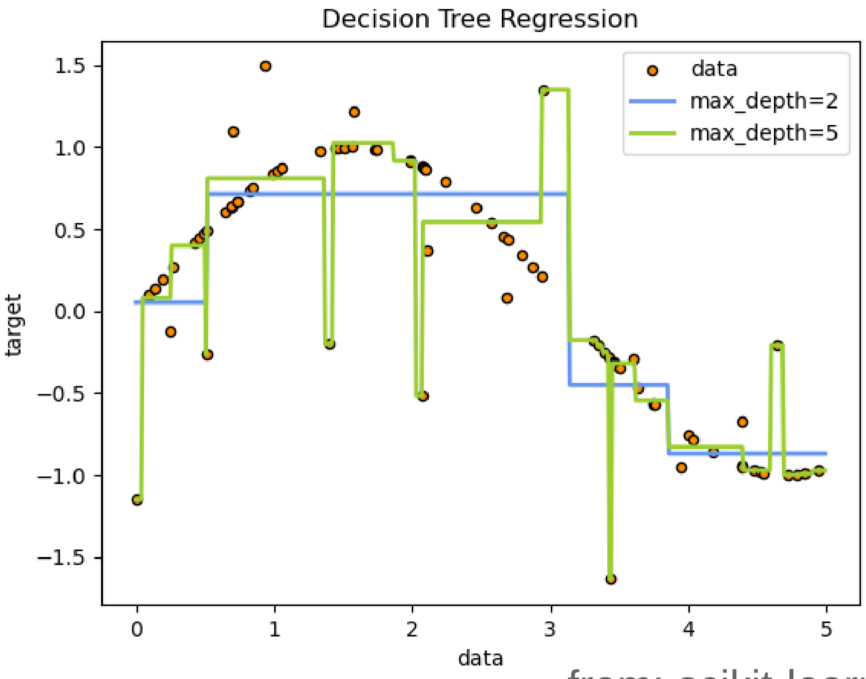
min_samples_split : *int or float, default=2*
The minimum number of samples required to split an internal node:

- If int, then consider `min_samples_split` as the minimum number.
- If float, then `min_samples_split` is a fraction and `ceil(min_samples_split * n_samples)` are the minimum number of samples for each split.

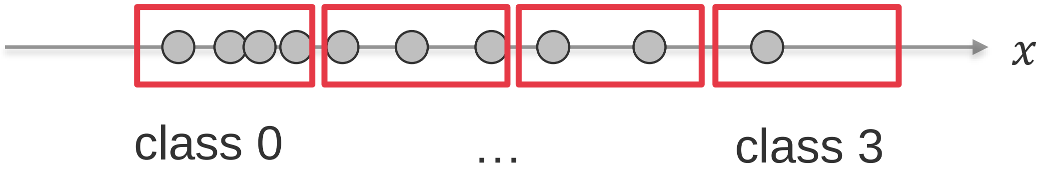
Regression Trees

Cyber-Physical Systems
in Mechanical Engineering TU Berlin

- So far: decision trees for classification problems
- Central idea: binning of continuous values into categories



from: scikit-learn

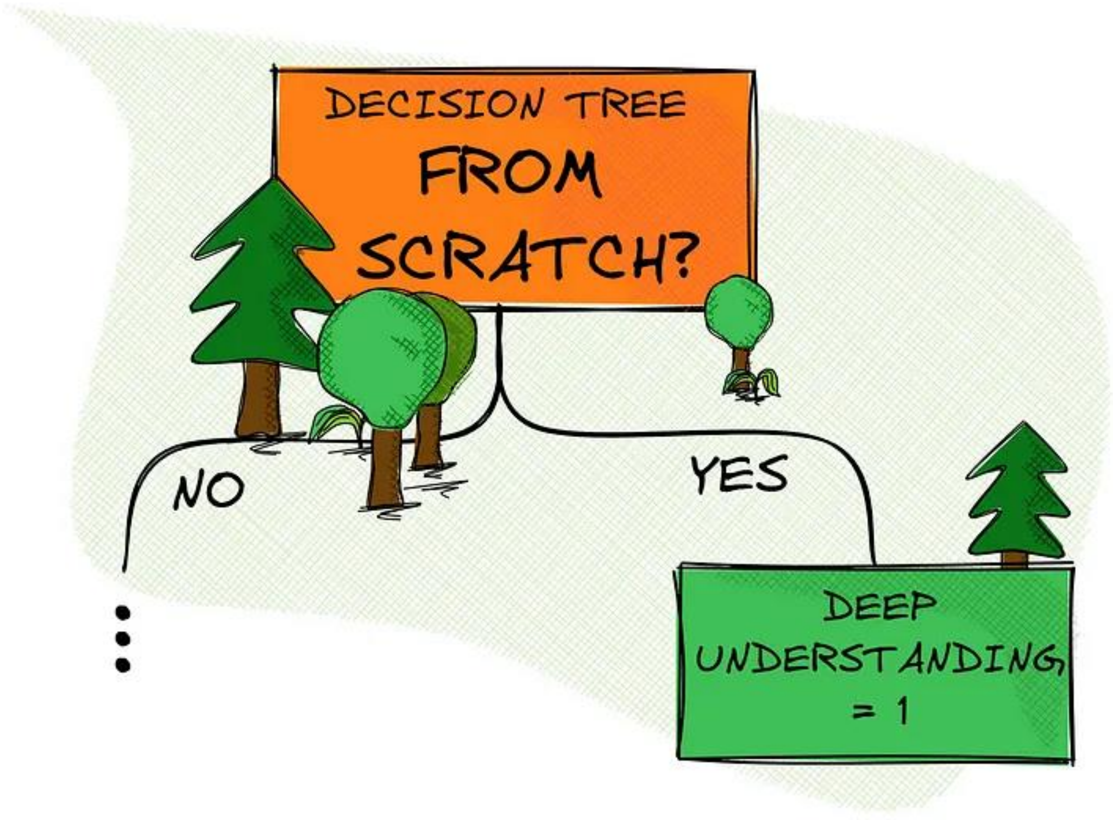


Exercise 06

May 24, 2023

Exercise 06

- Implementation of a decision tree from scratch



© Marvin Lanhenke, https://towardsdatascience.com/implementing-a-decision-tree-from-scratch-f5358ff9c4bb

Questions?