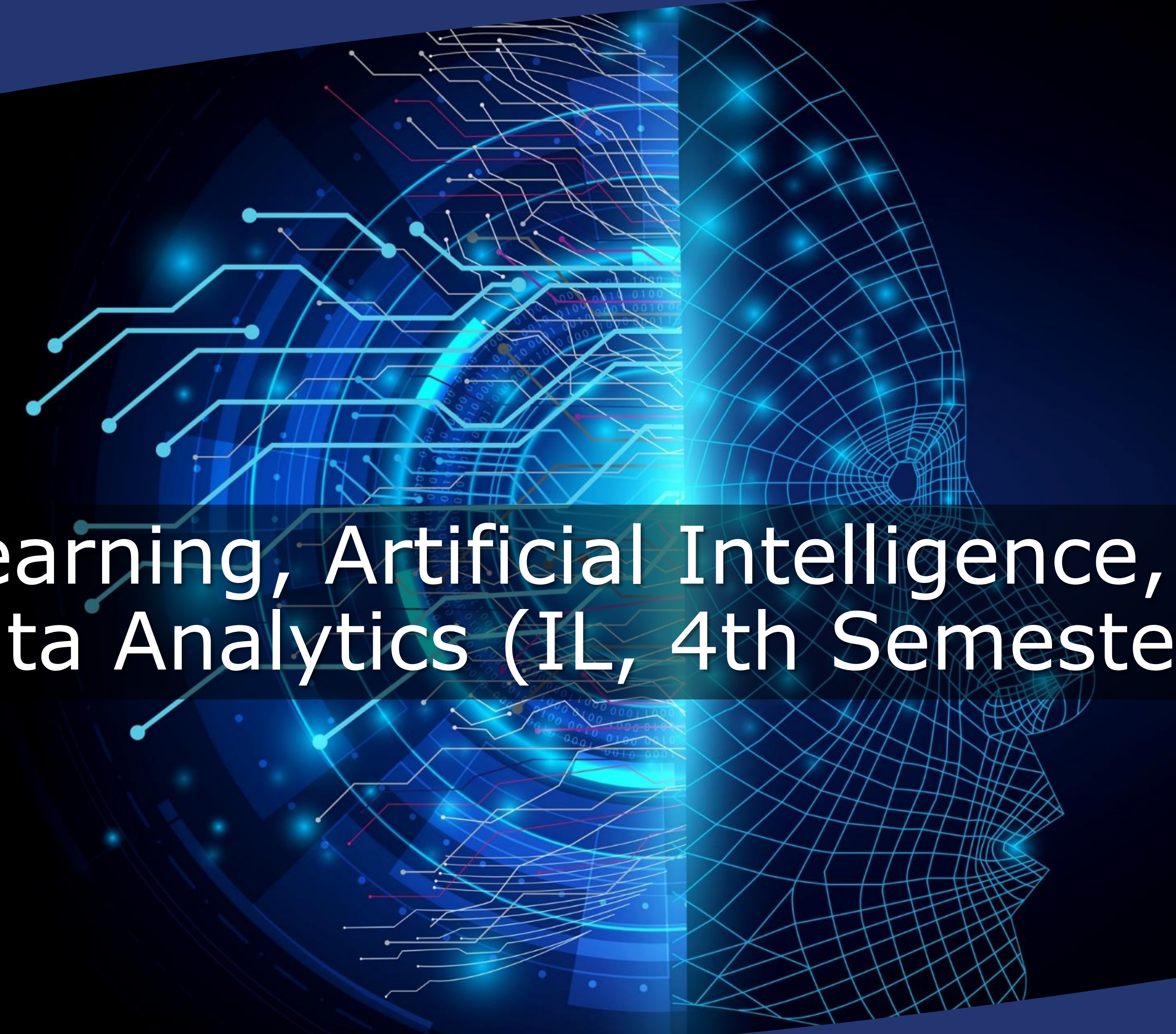
FH KREMS
UNIVERSITY OF APPLIED
SCIENCES / AUSTRIA

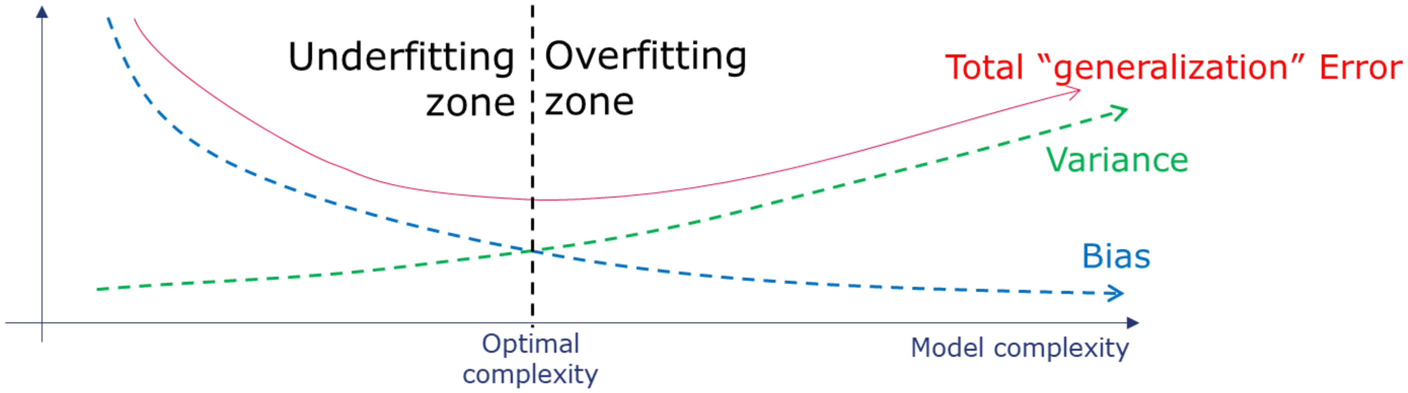


Machine Learning, Artificial Intelligence, and Big Data Analytics (IL, 4th Semester)

Lecture 4

Recap of previous lecture (1/4)

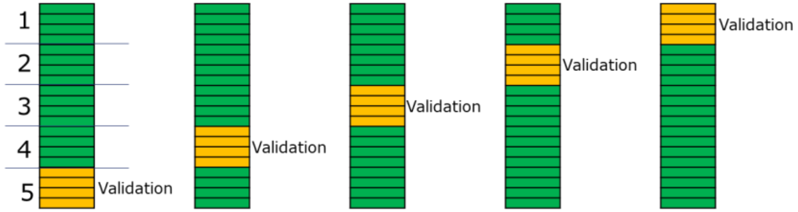
- *Generalization error* is a common issue to deal with in ML
- It can be decomposed into three terms: *Bias*, *Variance*, and *immutable error*
- High bias → Cannot capture the structure of the data → underfit
- High variance → Too sensitive to the training data → overfit



Recap of previous lecture (2/4)

- Techniques to keep the underfit/overfit under control:
- Split Training-Test

- Cross validation

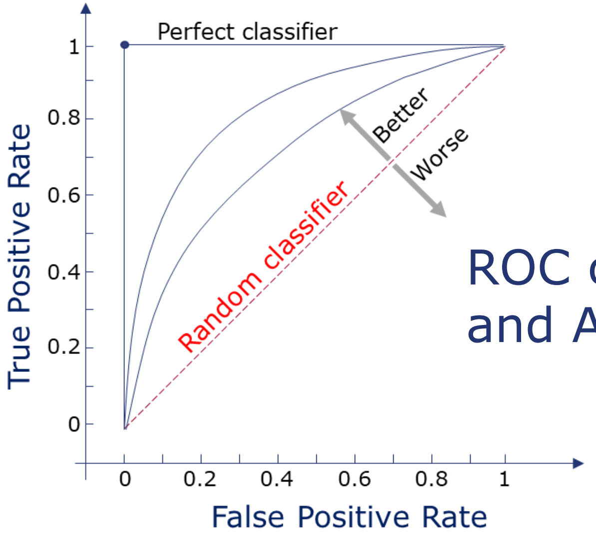


- Bootstrapping: Resampling with replacement
- Bagging: Bootstrapping multiple times and ensemble of multiple models

Recap of previous lecture (3/4)

- **Regression:** RMSE, MAE, MAPE, Rsquare
- **Classification:** Confusion Matrix → accuracy, precision, recall, FP Rate, FN Rate, F1-score, ROC, AUC…

Metric	Formula	Meaning	Visual look	range
Accuracy	(#TP+#TN)/#Total	What fraction does it get right	**TP** FN FP **TN** / **TP** **FN** **FP** **TN**	0-1
Precision	#TP/(#TP+#FP)	When it says 1 how often is it right	**TP** FN FP TN / **TP** FN **FP** TN	0-1
Recall/ Sensitivity	#TP/(#TP+#FN)	What fraction of 1s does it get right (True Positive Rate – TPR)	**TP** FN FP TN / **TP** **FN** FP TN	0-1
Specificity	#TN/(#TN+#FP)	What fraction of 0s does it get right (True Negative Rate – TNR)	TP FN FP **TN** / TP FN **FP** **TN**	0-1
FP Rate	#FP/(#FP+#TN)	What fraction of 0s are called 1s	TP FN **FP** TN / TP FN **FP** **TN**	0-1
FN Rate	#FN/(#TP+#FN)	What fraction of 1s are called 0s	TP **FN** FP TN / **TP** **FN** FP TN	0-1
F1-score	$2 * \frac{precision * recall}{precision + recall}$	How "good" are precision and recall		0-1

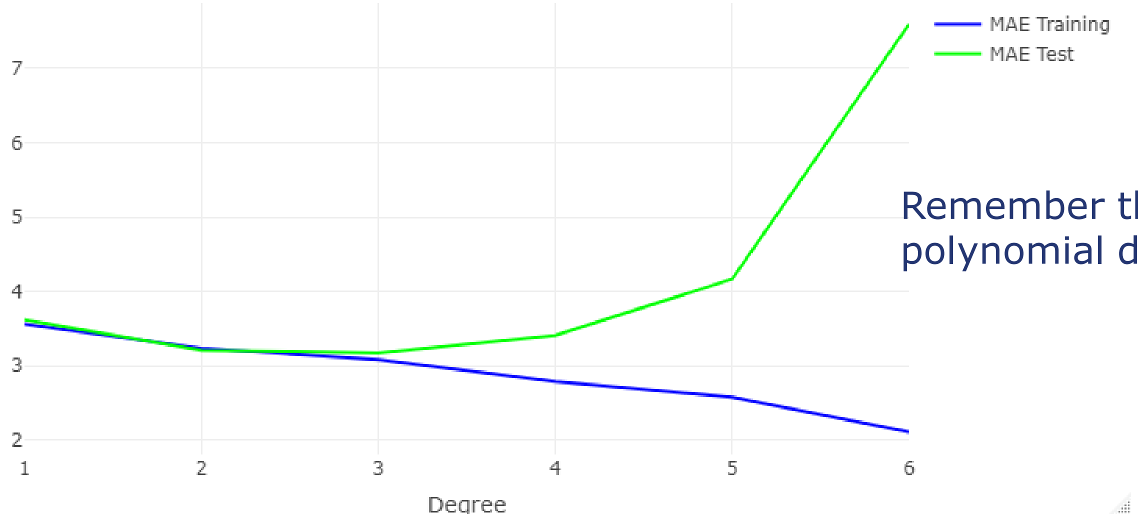


ROC curve and AUC

Recap of previous lecture (4/4)

What we learned in the coding session

- Stratification helps keeping the class balance in train/test dataset.
- You can make overly complex models and fit perfectly to the training data, but it is almost never a good idea!
- Most of the ML techniques rely on watching the error in the validation/test set and choosing the simplest model with acceptable performance



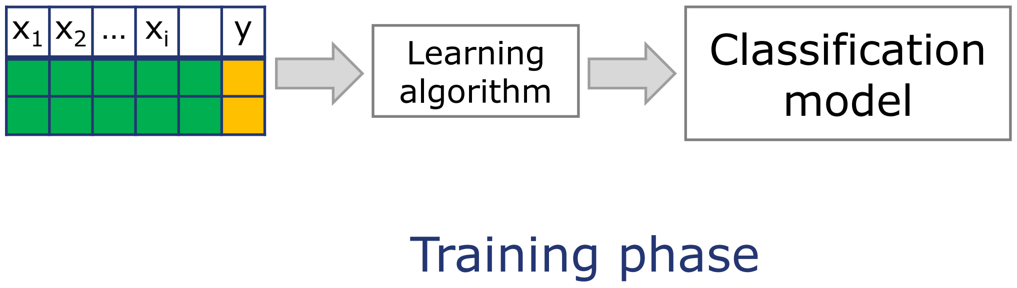
Remember the polynomial degree?

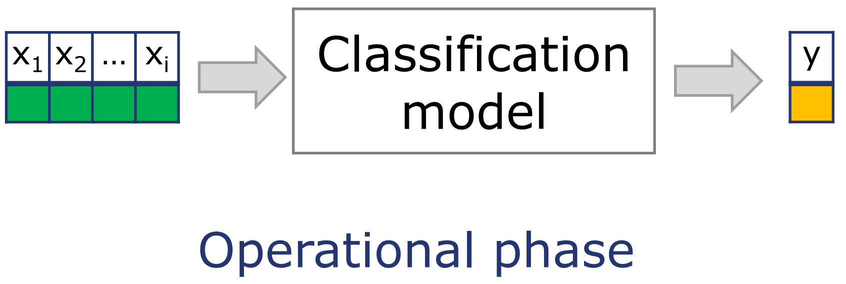
Agenda

- Decision trees
  - Characterizing a tree
  - Building a tree
  - Pruning a tree
- Coding part
- Exercise

Recall the goal of a classification task

- Classification: Predict category from one or more input variables
- A classification model is trained by learning the relation between input and output from a labeled dataset. It learns how to predict the output variable.





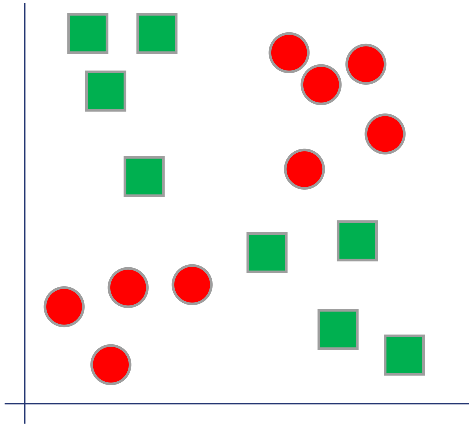
Common classification algorithms

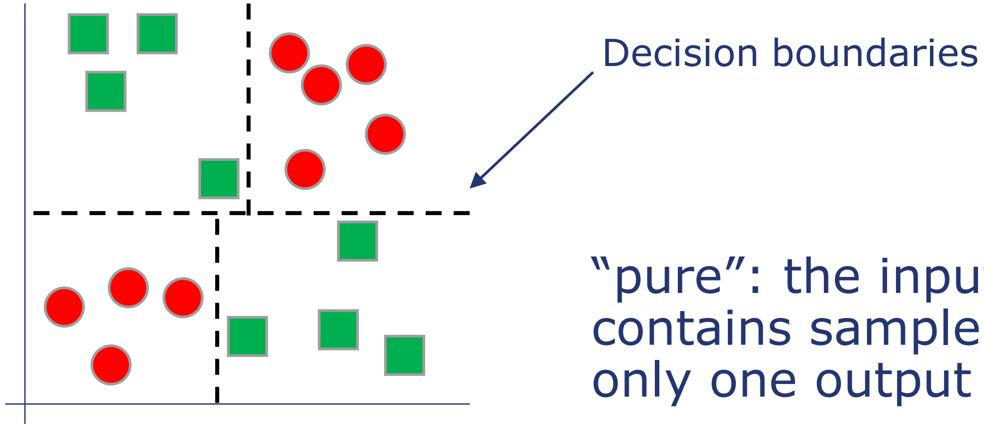
- Decision trees
- K-nearest neighbors
- Naïve Bayes
- Support Vector Machine
- Random Forest
- Artificial Neural Network

- And many others…

Decision trees

- Idea: Split data into subsets with one output class



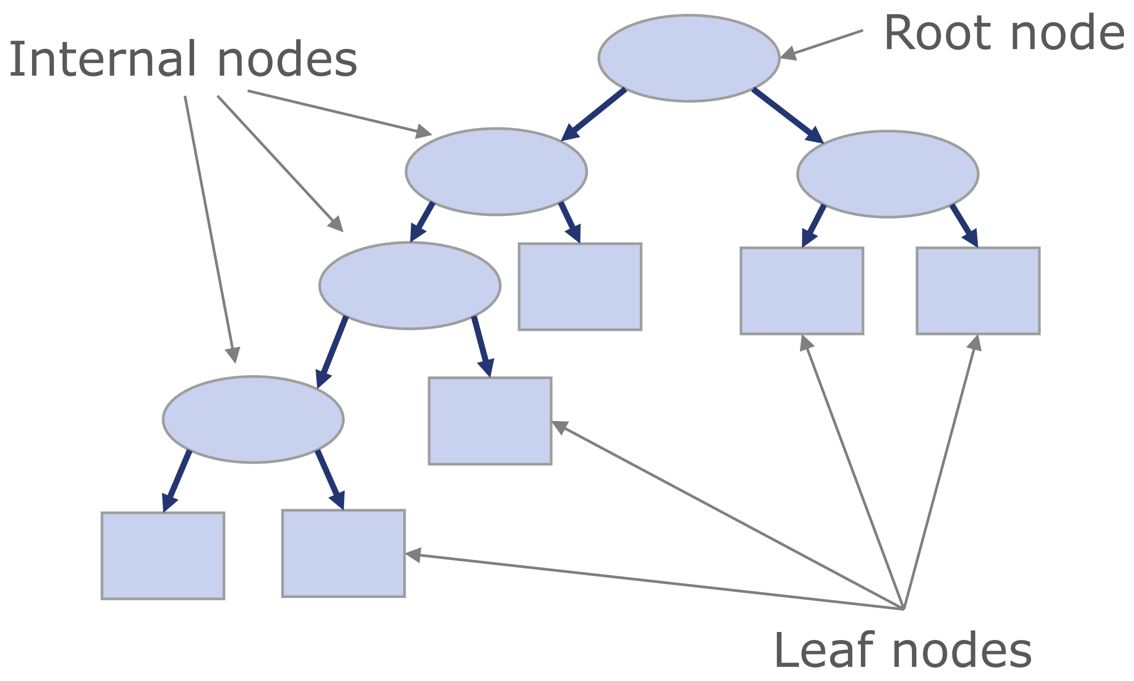


"pure": the input space contains sample that have only one output class

Goal: Split the data into regions that are as pure as possible

Decision trees

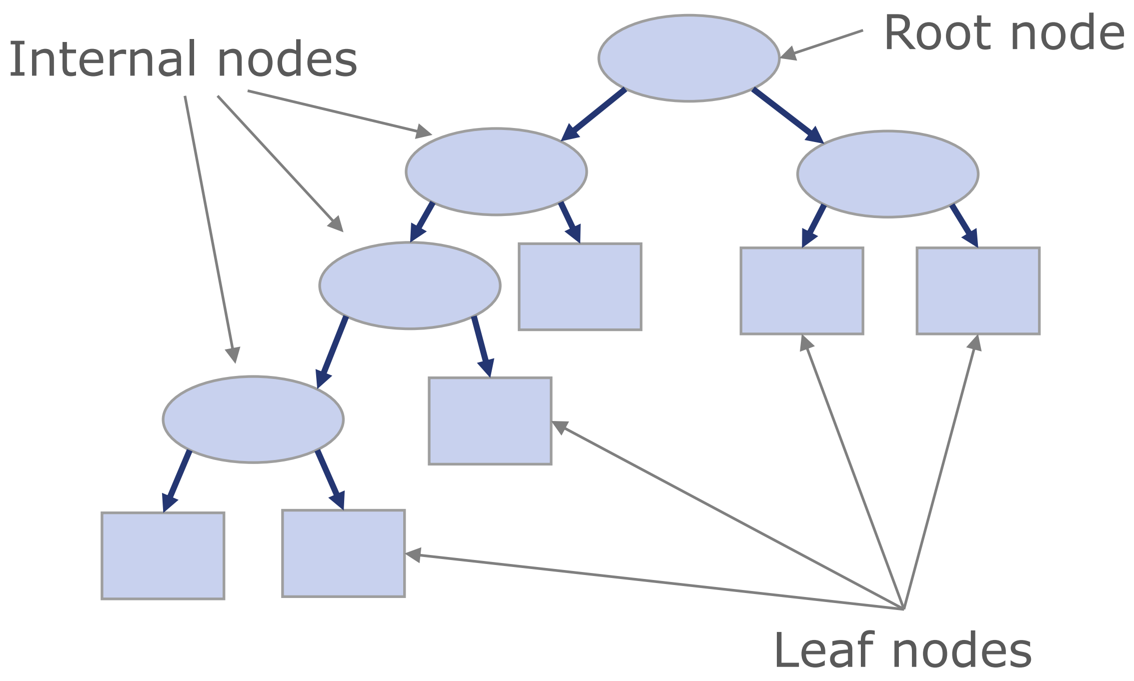
Structure



- Hierarchical structure with nodes and directed edges
- Root and internal nodes have test conditions
- Leaf nodes have class labels associated with it
- The classification is done by traversing the tree from top until a leaf

Decision trees

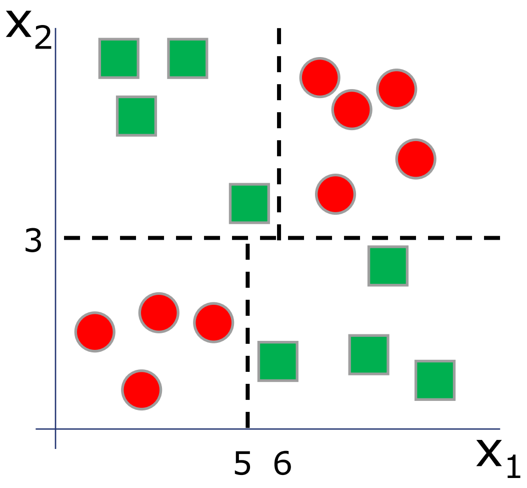
Structure

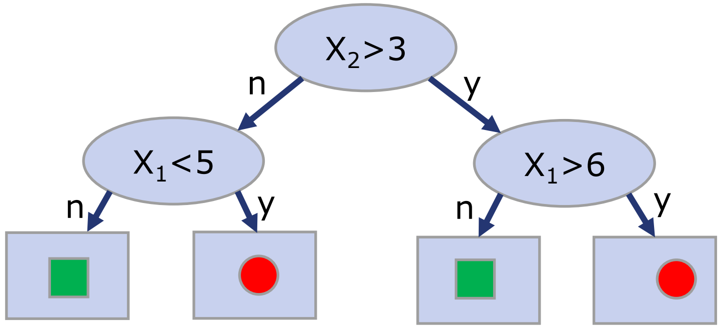


- **Depth of a node**: number of edges from the root to the node
- **Depth of a tree**: Number of edges of the longest path in the tree
- **Size of a tree:** Number of nodes in a tree

Decision trees

Example

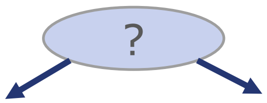




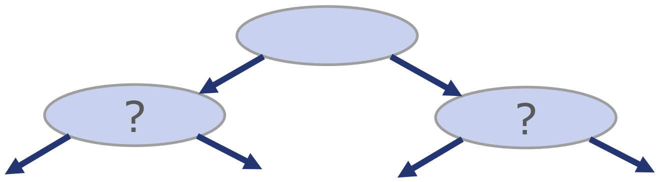
Decision trees

Main principle

- Start with all samples in a node
- Partition such that the subsets are as pure as possible



- Repeat the partitioning

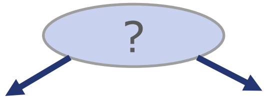


- Until a stopping criterion is satisfied

Decision trees

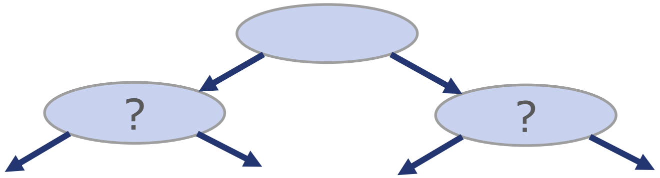
Main principle

- Start with all samples in a node
- Partition such that the subsets are as pure as possible



Question 1. How to partition → Splitting criterion

- Repeat the partitioning



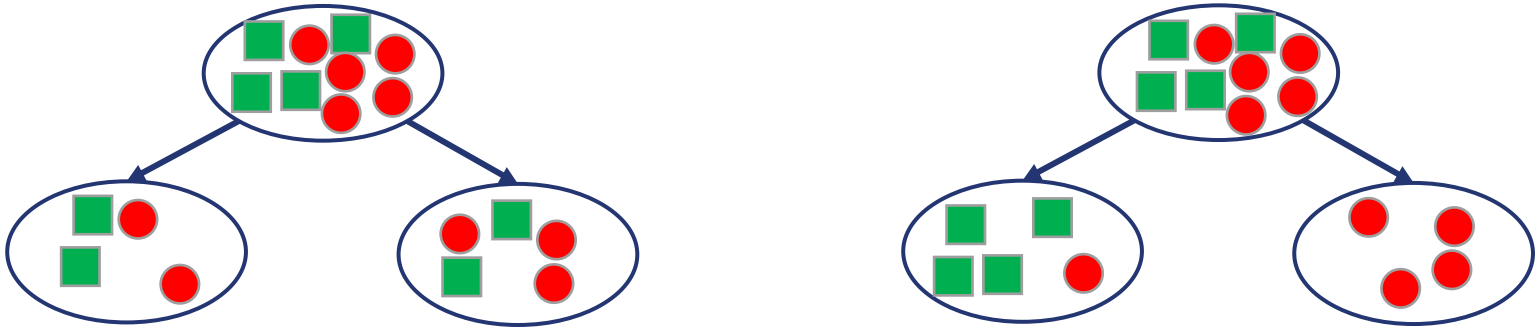
- Until a stopping criterion is satisfied

Question 2. When to stop → Stopping criterion

Decision trees

Splitting criterion

- How to measure purity?

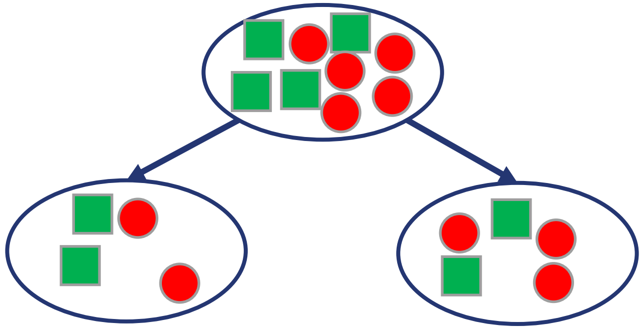


What's the best splitting here?

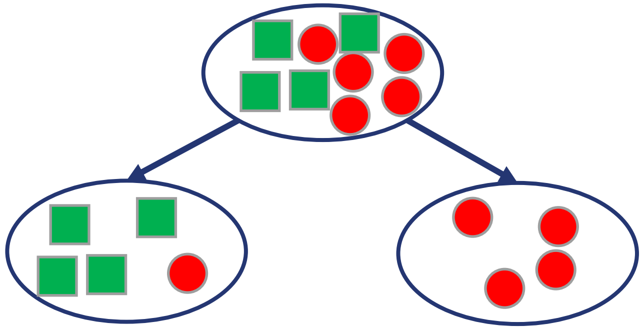
Decision trees

Splitting criterion

- How to measure purity?



X

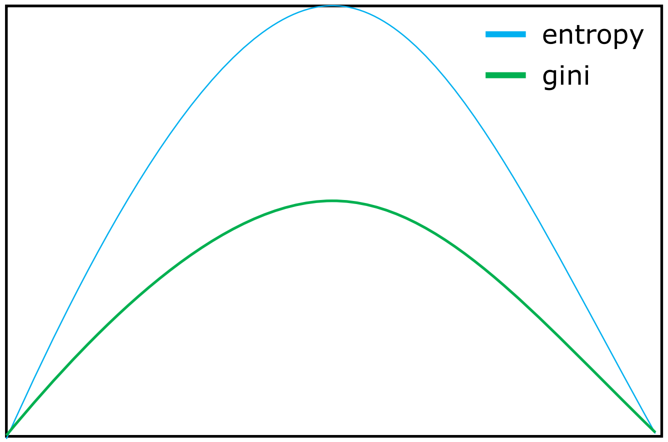


V

Decision trees

How to measure the purity of the subsets

- **Gini impurity index =** $\mathbf{1 - \sum_j p_j^2}$
  - All samples one class:
    $$1 - 1^2 = 0$$
  - Equally distributed in two classes:
    $$1 - (0.5^2 + 0.5^2) = 0.5$$
- **Entropy =** $\mathbf{-\sum_j (p_j * \log_2 p_j)}$
  - All samples one class:
    $$-1 * \log_2 1 = 0$$
  - Equally distributed in two classes:
    $$-0.5 * \log_2 0.5 - 0.5 * \log_2 0.5 = 1$$

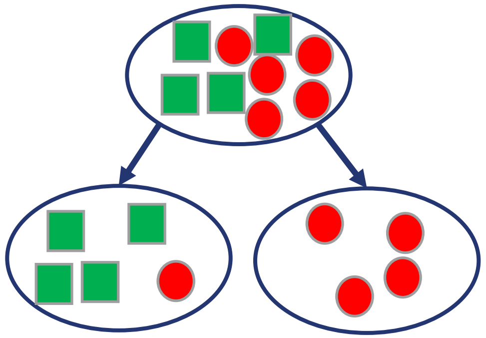


Decision trees

When to stop growing the tree

Most common criteria

- A percentage of samples have the same class
- The number of samples reaches a minimum
- Change in purity is smaller than a threshold
- Max tree depth is reached



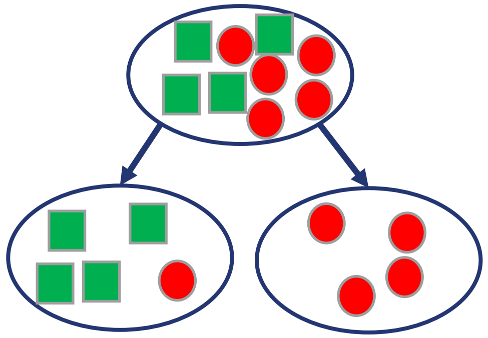
Example of condition	Parameter in *rpart*
Class > perc_threshold	-
N_samples < min_samples	*Minsplit*
Purity gain < threshold	*CP*
Depth == max_depth	*Maxdepth*

Decision trees

Or grow it all and “prune” it later

Common approach:

- Grow the entire tree
- Plot the results at different “levels” (e.g., different purity gains or different depth)
- Find the level where crossvalidation error is minimal
- “Prune” the tree!

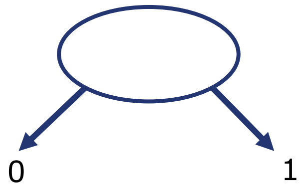


Example of condition	**Parameter in *rpart***
Class > perc_threshold	-
N_samples < min_samples	*Minsplit*
Purity gain < threshold	*CP*
Depth == max_depth	*Maxdepth*

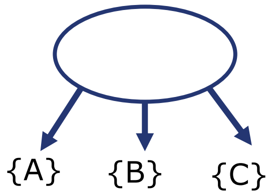
Decision trees

How to split on different types of variables

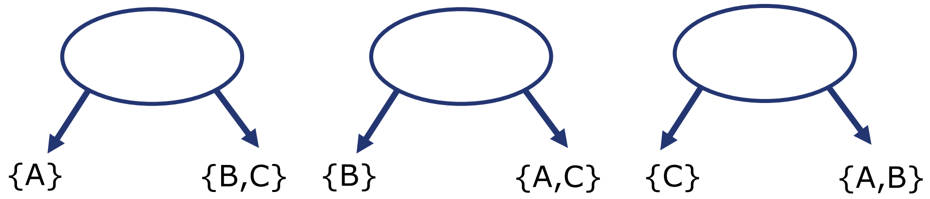
- Binary



- Categorical



Multi-way split

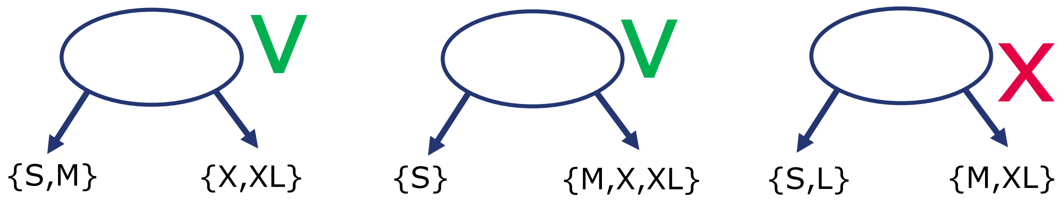


Binary split

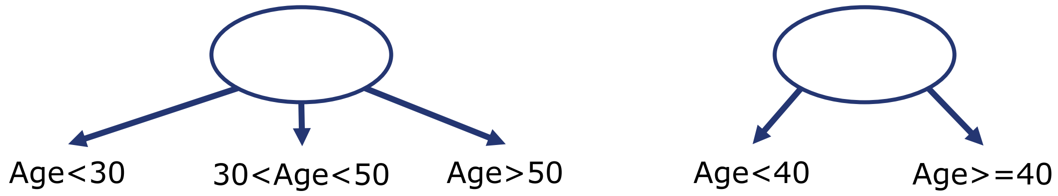
Decision trees

How to split on different types of variables

- Ordinal
    - Like categorical but attention to the order



- Numerical



Range query

Binary

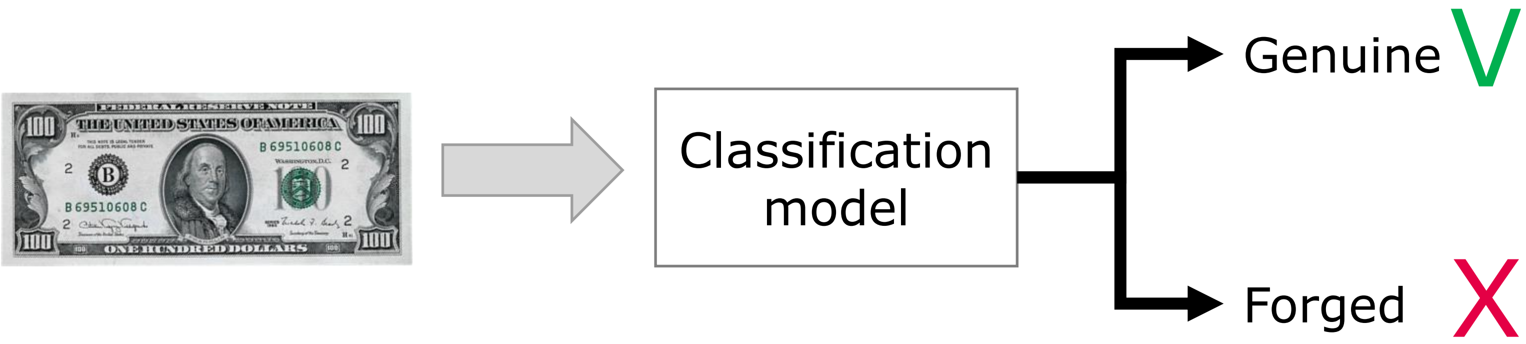
Decision trees

Considerations

- Advantages:
  - Very simple
  - Works with non-linear problems
  - Interpretable
  - Computationally cheap
- There are some drawbacks
  - Unstable (A change in the data can lead to completely different model)
  - Mostly based on heuristic with no solid statistical background
  - Prone to overfit
- Tree ensemble techniques (Random Forest, XGBoost) will be covered in coming lectures.

Decision tree

Practical Example – Detection of forged banknotes



Decision tree

Practical Example – Detection of forged banknotes

Genuine
Genuine
Genuine
Genuine
Forged
⋮
Forged

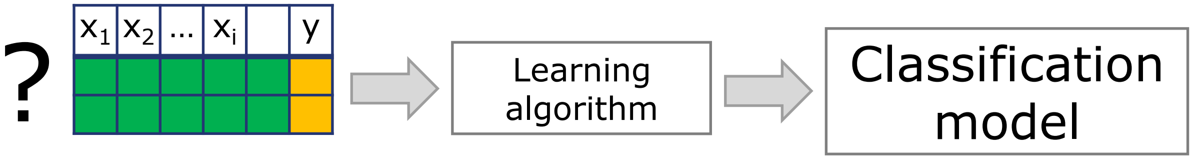
Input dataset

Decision tree

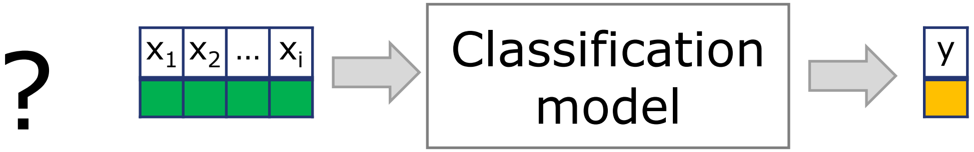
Practical Example – Detection of forged banknotes

Genuine
Genuine
Genuine
Genuine
Forged
⋮
Forged

Input dataset



Training phase



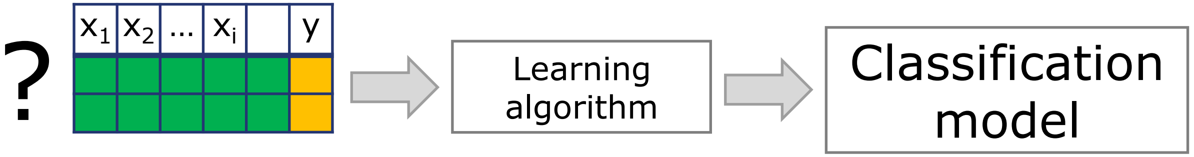
Operational phase

Decision tree

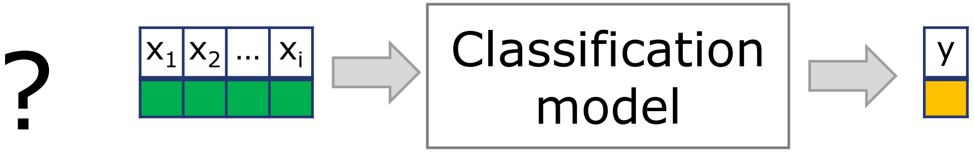
Practical Example – Detection of forged banknotes

Wavelet Transform

Variance of the wavelet
Skewness of the wavelet
Curtosis of the wavelet
Entropy of the wavelet



Training phase



Operational phase

Decision tree

Practical Example – Detection of forged banknotes

Wavelet Transform

Variance of the wavelet
Skewness of the wavelet
Curtosis of the wavelet
Entropy of the wavelet

variance	skewness	Curtosis	entropy	isForged
var_1	$skew_1$	cur_1	$entr_1$	y_1
var_2	$skew_2$	cur_2	$entr_2$	y_2
var_3	$skew_3$	cur_3	$entr_3$	y_3
var_j	$skew_j$	cur_j	$entr_j$	y_j

Learning algorithm

Classification model

Training phase

variance	skewness	curtosis	entropy

Classification model

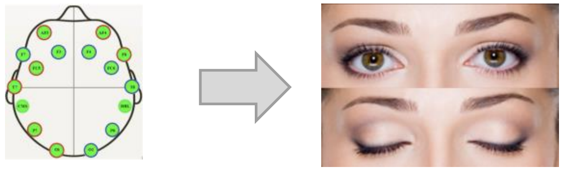
isForged
?

FH KREMS
UNIVERSITY OF APPLIED SCIENCES / AUSTRIA

Let’s do it!

Exercise 2 – **Due on 26.03.2023 23.59 CET**

(late submission +1 week, max 6 points)



- Import the EEG data from the assignment
  - 16 EEG numerical features
  - 1 label (1 = eyes open, 2 = eyes closed)
  - Train and test are already split (*eeg_training.csv*, *eeg_test.csv*)
- Create a classifier that can detect if the patient has open or closed eyelid
- Part 1: Decision Trees
- Part 2: Random Forest, AdaBoost, and XGBoost
- **Attention: This is not the same data that can be found online. Do not copy!**
- Hints: Are there NAs? Is it balanced? Are there useless features? Are there outliers?

Lecture E-LEARNING

- Register to datacamp.com using your IMC email address following this invite.
- You'll find one assignment already: "Support Vector Machines with R"
- **Due date: March 31st**

Lecture E-LEARNING

Support Vector Machines