Bioinformatical analysis of omics expression data Part 6

Dr. Michael Turewicz[1,2]

[1]Institut für Klinische Biochemie und Pathobiochemie,
Deutsches Diabetes-Zentrum, Leibniz-Zentrum für Diabetesforschung an der Heinrich-Heine-Universität, Düsseldorf, Deutschland
[2]Deutsches Zentrum für Diabetesforschung (DZD), München-Neuherberg, Deutschland

Course schedule

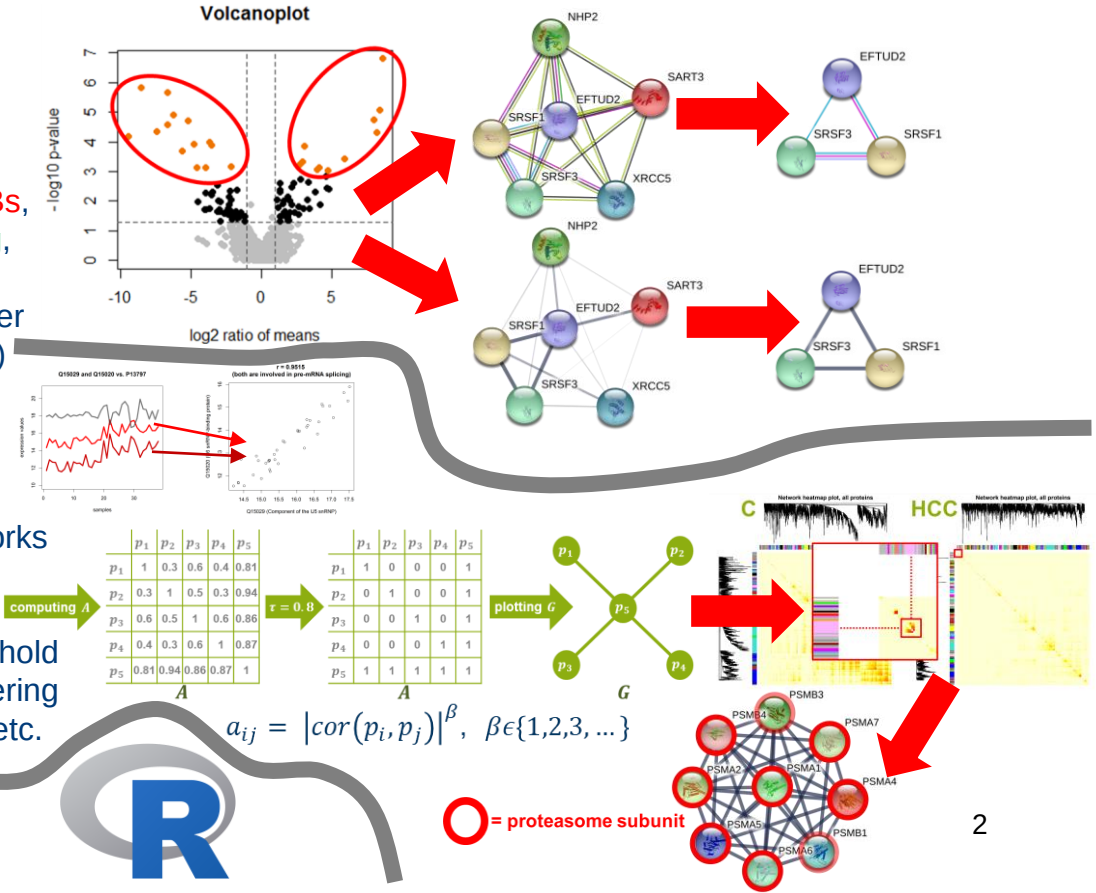
Deutsches Diabetes-Zentrum

- Part 1 (25.10.23)
  - Introduction (omics, example data, programming)
  - Data preprocessing (data inspection, normalization, missing values)
  - Exercises: R programming tutorial (part 1)
- Part 2 (08.11.23)
  - Differential expression analysis (statistics, volcano plot)
  - Exercises: R programming tutorial (part 2)
- Part 3 (15.11.23)
  - Machine learning I: Clustering (clustering, PCA)
  - Exercises: Customized hierarchical clustering & PCA in R
- Part 4 (22.11.23)
  - Overrepresentation analysis (GO, Reactome)
  - Exercises: Own GO- & Reactome analysis in R & other tools
- Part 5 (29.11.23)
  - Network analysis (esp. STRING)
  - Exercises: Own network analysis in R & STRING
- **Part 6 (06.12.23)**
  - **Machine learning II: Tree-based classification algorithms**
  - **Exercise: Own tree-based classification in R**

Recap of previous part

DDZ Deutsches Diabetes-Zentrum

- **PPI networks constructed from DBs**
  - Input: usually list of differential candidates
  - Output: input-specific PPI subnetwork from PPI-DB
  - **STRING:** DB of **known** & **predicted** PPIs
  - Kinds of PPIs: experimentally determined, curated DBs, gene neighborhood/fusions/co-occurrence, textmining, co-expression, gene homology
  - Set confidence score (c) & kinds of PPIs: reliability filter
  - Lax (c > 0.15, all PPIs) vs. strict (c > 0.9, known PPIs)
  - Evidence view vs. confidence view
  - Also ORA available
- **PPI networks inferred from data**
  - Protein co-expression networks: **predicted** PPI networks
  - → metric for co-expression: correlation
  - Idea: correlation matrix for all pairs of genes/proteins
  - Output: adjacency matrix for set/computed corr.-threshold
  - WGCNA: Modules & hubs based on topological clustering
  - Confirm biological relevance with ORA, known PPIs, etc.
- **Programming (example: R)**
  - Own STRING- & WGCNA-based ORA network construction in R & STRING web application

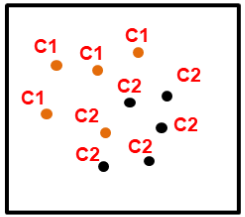


$$a_{ij} = \left|cor(p_i, p_j)\right|^{\beta}, \quad \beta \epsilon \{1,2,3,\dots\}$$

Machine learning: tasks & example algorithms

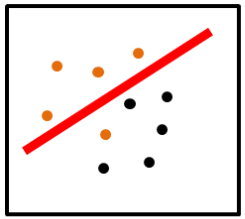
DDZ Deutsches Diabetes-Zentrum

- **Classification** (output = class label): Support vector machines, decision trees, neural networks…

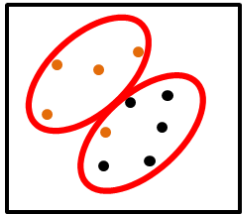


→ **Discussed today!**

- **Regression** (output = (continuous) numbers): Support vector regression, regression trees…

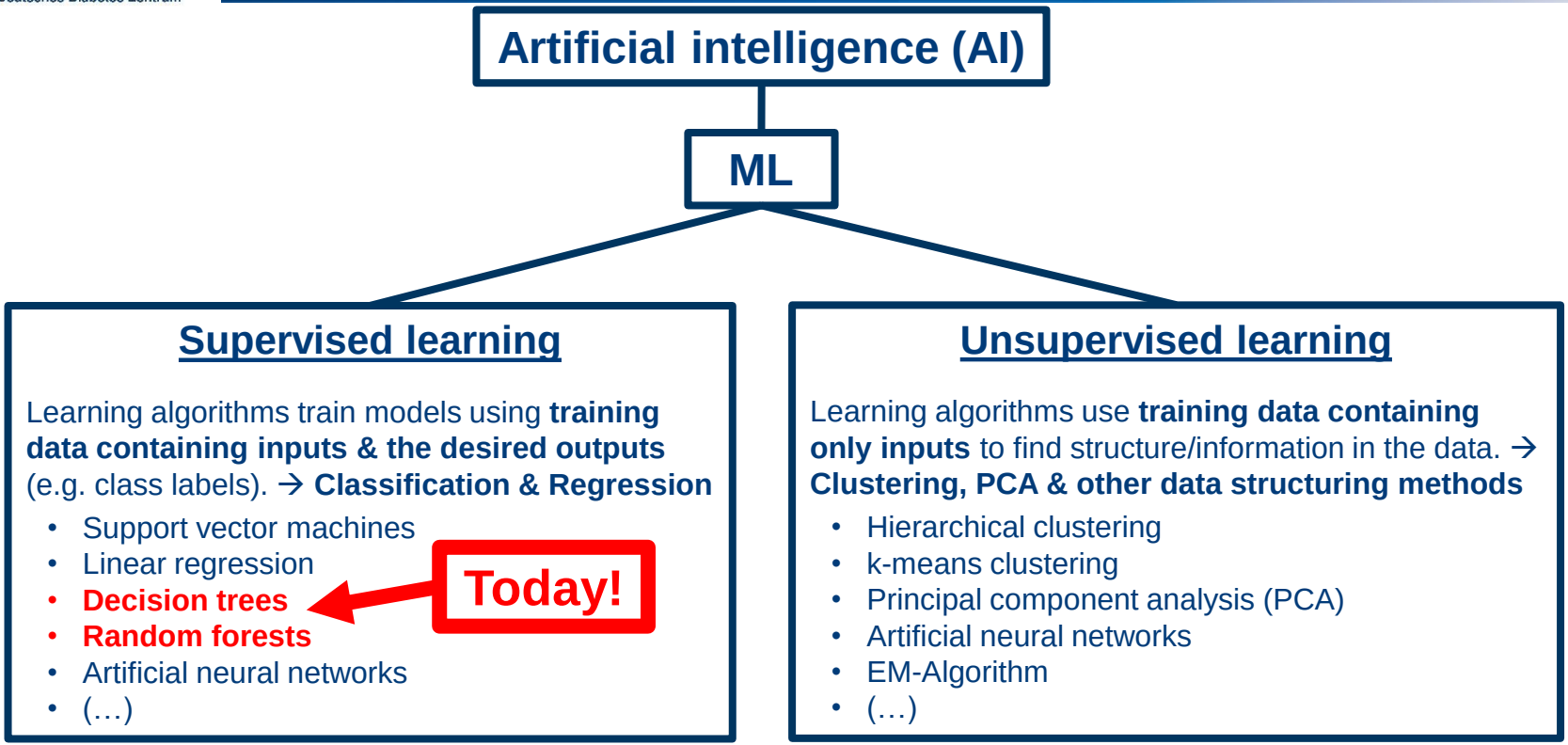


- **Clustering** (output = clusters of data points): k-Means, hierarchical clustering, …



Machine learning (ML)

DDZ Deutsches Diabetes-Zentrum



Biomarker discovery

DDZ Deutsches Diabetes-Zentrum

- Which genes/proteins are differentially expressed between two or more groups?
- Application: A gene/protein with a specific expression level for a disease (or another state/condition) can be used for diagnosis (or state/condition indicator). → so called **“biomarker”**.
- Discovery of differentially expressed genes/proteins → **biomarker discovery**

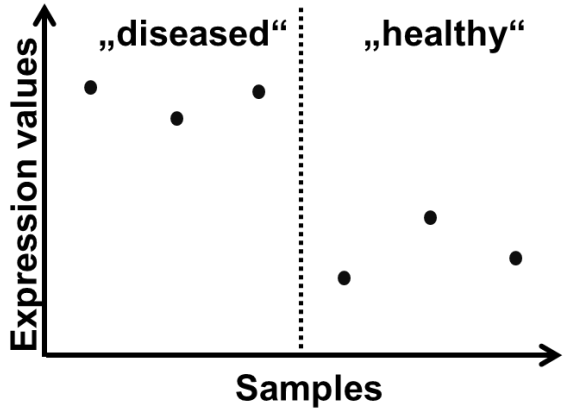
Protein expression values

Diseased samples

Healthy samples

	D1	D2	D3	H1	H2	H3
P1	50	100	50	100	50	100
P2	300	250	270	90	120	110
P3	500	600	550	590	490	610
…	…	…	…	…	…	…

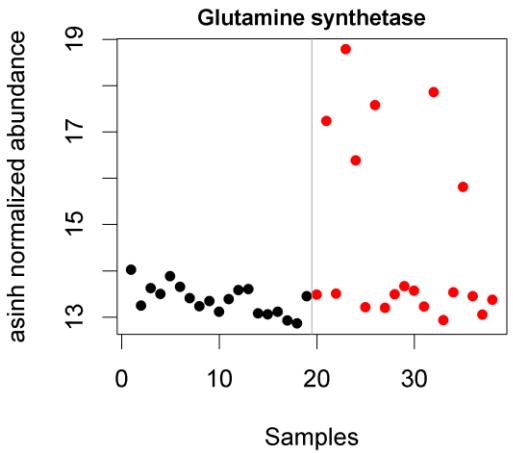
Proteins

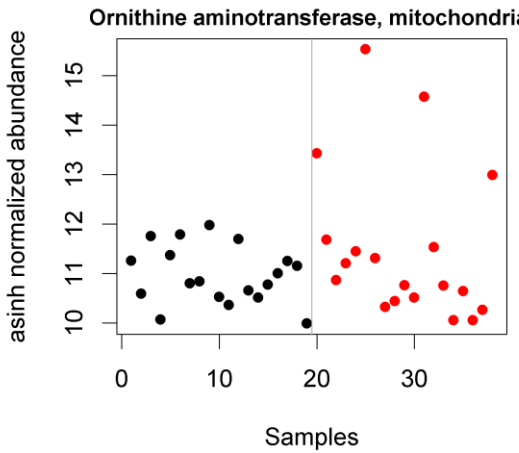


Biomarker panels

DDZ Deutsches Diabetes-Zentrum

- Often there is no gene/protein overexpressed in all samples of the diseased group
- Alternative: **biomarker panel**, a set of **complementary genes/proteins**
- Biomarker panels provide usually better classification results than single biomarkers





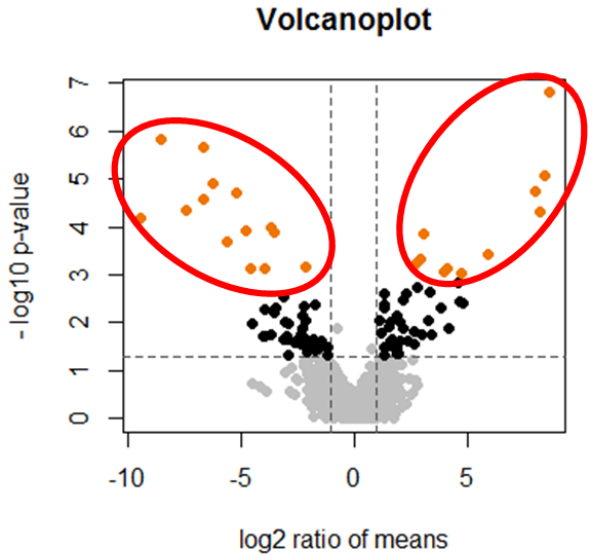
Example: Panel of two proteins

	D1	D2	D3	D4	H1	H2	H3	H4
P1	50	100	50	100	50	100	50	100
P2	800	950	40	60	50	30	60	70
P3	500	600	550	580	590	490	610	530
P4	40	50	700	790	30	60	40	20
…	…	…	…	…	…	…	…	…

Univariate vs. multivariate selection methods

DDZ Deutsches Diabetes-Zentrum

1. Differential analysis: selecting candidates with p-values & fold changes → **consider each single gene/protein independently of the others (univariate methods)**



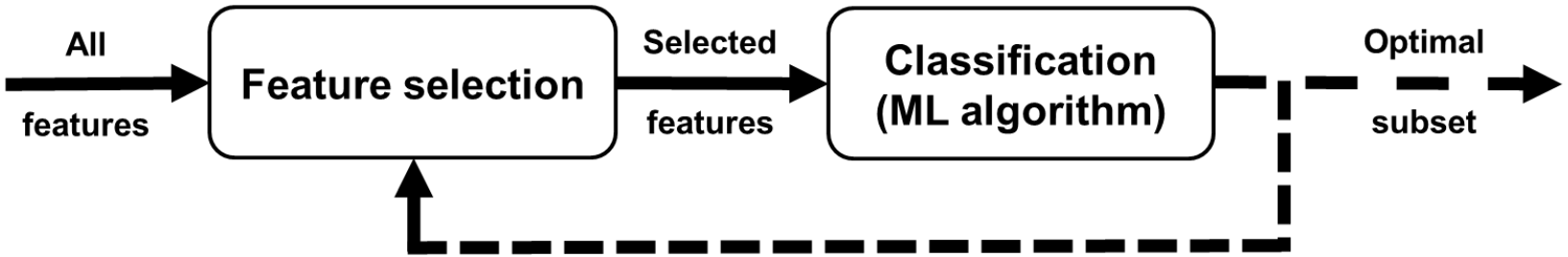
2. Machine learning approaches → **consider multiple genes/proteins together (multivariate methods)**

ML-based approach

DDZ Deutsches Diabetes-Zentrum

- Task: Select optimal differential subset of genes/proteins
- For this **'feature selection'** methods can be used (**'features' = genes/proteins = biomarkers/candidates**)
- In ML, feature selection was developed for dimensionality reduction to improve model performance & runtimes, i.e. reducing **"*n << p – problem*"**
- Assessment of subset: **classification algorithm (classification accuracy)**
- Optimization problem: **minimal feature number** & **maximal classification accuracy**

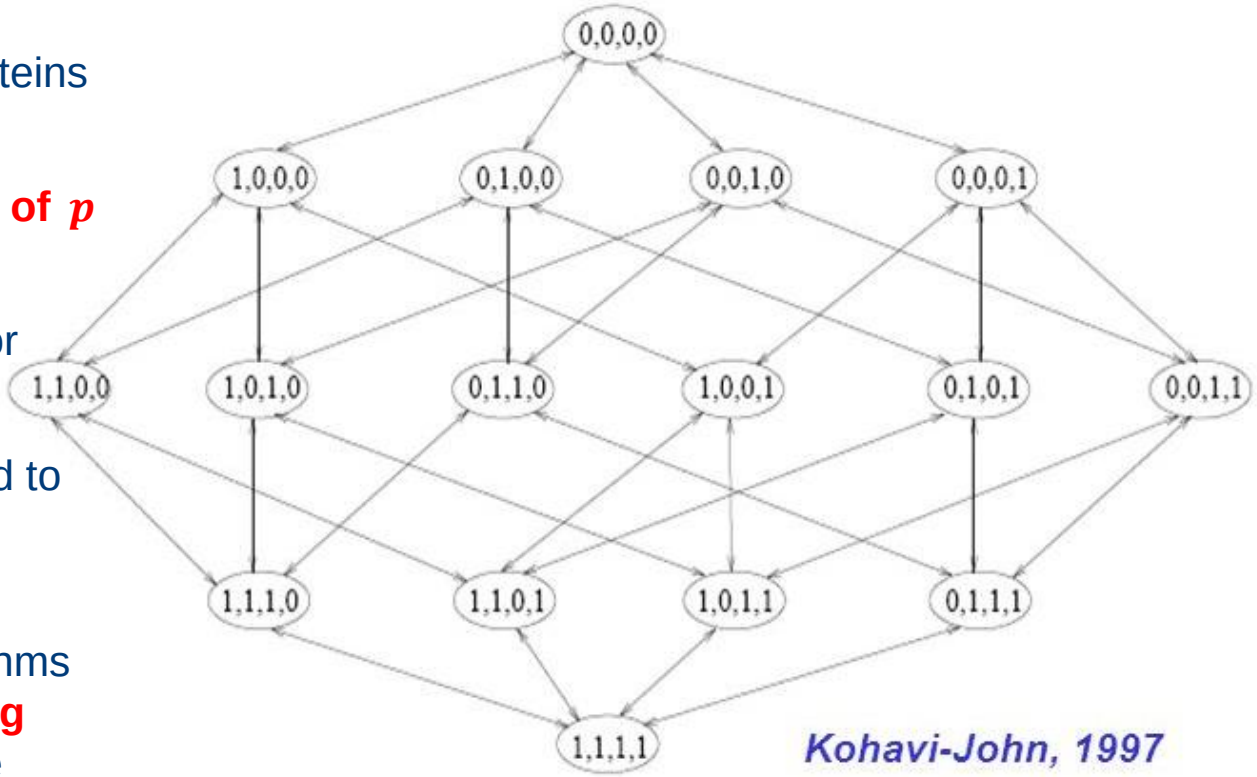
General approach:



ML: Computational challange

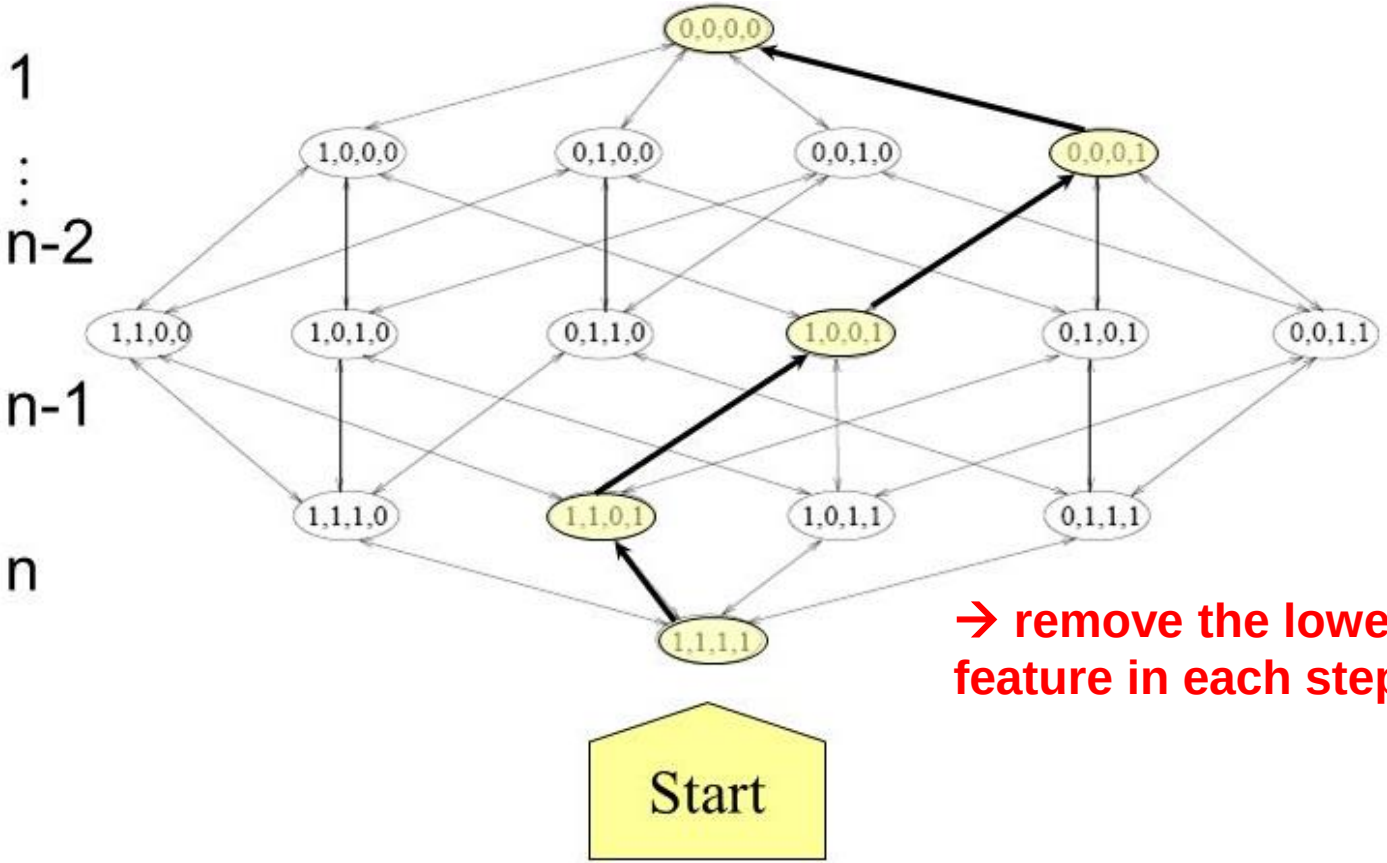
DDZ Deutsches Diabetes-Zentrum

- Optimization problem:
  all combinations of genes/proteins must be assessed
- There are **2^p combinations of p features**!
- → computational challenge for thousands of genes/proteins!
- → Heuristic strategies needed to skip most subsets & reduce runtimes!
- → Some classification algorithms perform an own **feature rating** which can be used for feature selection



Backward / recursive feature elimination

DDZ
Deutsches Diabetes-Zentrum



→ **remove the lowest rated feature in each step**

Classification algorithms

DDZ Deutsches Diabetes-Zentrum

- k-nearest neighbor (k-NN) classifier
- Naive Bayes classifier
- Deep learning / artificial neuronal networks
- Linear discriminant analysis (LDA)
- Support vector machine (SVM)
- **Decision trees**
- **Random forest (RF)**
- (…)

Provide feature importance rating & discussed today…

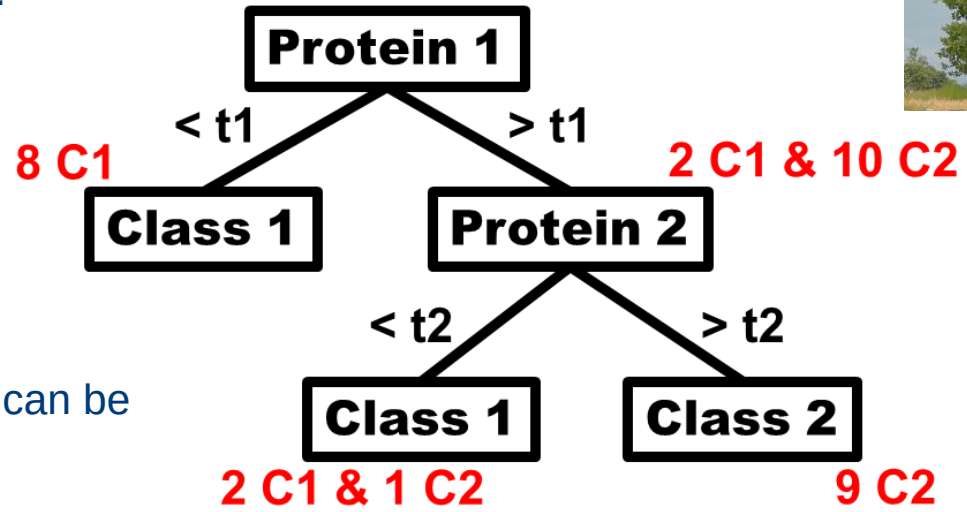
Decision trees

DDZ Deutsches Diabetes-Zentrum

- As part of the training of decision trees (DTs) most discriminative features for the remaining cases are chosen.

- Feature importance metrics:
  - Entropy
  - Gini impurity measure
  - Classification accuracy

- Main DT advantages
  - Classification paths → rules can be formulated
  - Easy interpretation



- Main DT disadvantage
  - Growing trees = growing **overfitting** problems → relatively high error rates

Decision tree learning (Gini impurity measure)

DDZ
Deutsches Diabetes-Zentrum

$$I_G = \sum_{i=1}^{C} p_i \ (1 - p_i)$$

where C is the number of classes and p_i is the probability of class i for the considered variable

$$p_1 = p_2 = \frac{1}{2}$$

$$I_G = \left[\frac{1}{2}\left(1 - \frac{1}{2}\right)\right] + \left[\frac{1}{2}\left(1 - \frac{1}{2}\right)\right] = \frac{1}{4} + \frac{1}{4} = \frac{1}{2}$$

$$p_1 = 1;\ p_2 = 0$$

$$min(I_G) = [1(1-1)] + [0(1-0)] = 0 + 0 = 0$$

$$max(I_G) = 1 - \frac{1}{C}$$

...so, $max(I_G)$ depends on the number of classes C

Decision tree learning: basic algorithm

DDZ Deutsches Diabetes-Zentrum

BuildTree(depth h, data D_i, node n_j)

- For each variable v_k find threshold t_k minimizing $I_G(v_{k1}) + I_G(v_{k2})$ for split $v_{k1} < t_k < v_{k2}$
- Select split at v_k with the highest I_G-reduction: $I_G(v_k) - (I_G(v_{k1}) + I_G(v_{k1}))$.
- Stop if stopping criterion == TRUE
- Split D with respect to t_k of v_k and create two child nodes of n: n_{k1} and n_{k2}
- BuildTree(h+1, D_{k1}, n_{k1})
- BuildTree(h+1, D_{k2}, n_{k2})

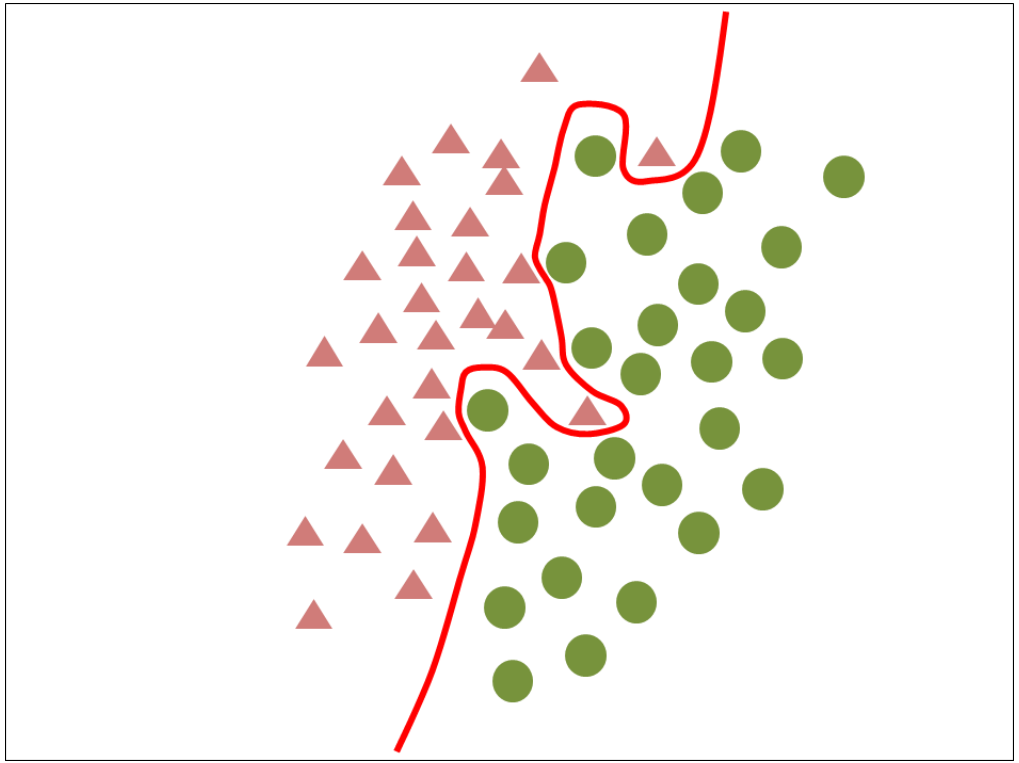
Call: BuildTree(0, D, n_{root})
Stopping criterion: depth threshold, information gain threshold, all leafs only 1 class

Prominent learning algorithms:

- CART (similar to above idea, Leo Breiman et al., 1984)
- ID3 (J. Ross Qinlan, 1986)
- C4.5 (J. R. Quinlan, 1993)

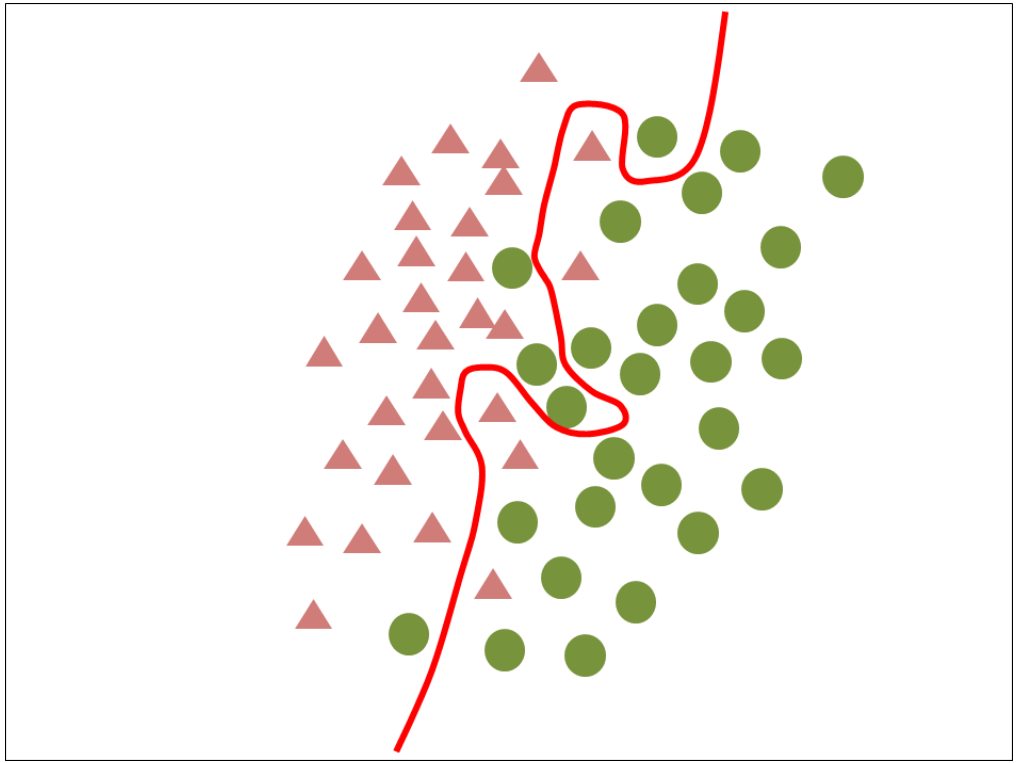
Overfitting

DDZ Deutsches Diabetes-Zentrum



- During training, ML algorithms "learn“ / “fit” parameters of their underlying mathematical functions using **given training data**
- Training data needs to be representative!
- However, usually, omics training data is not ideally representative…

Overfitting



Another omics dataset (also technical replicates) is always (at least slightly) different due to technical noise, natural variability, (…)

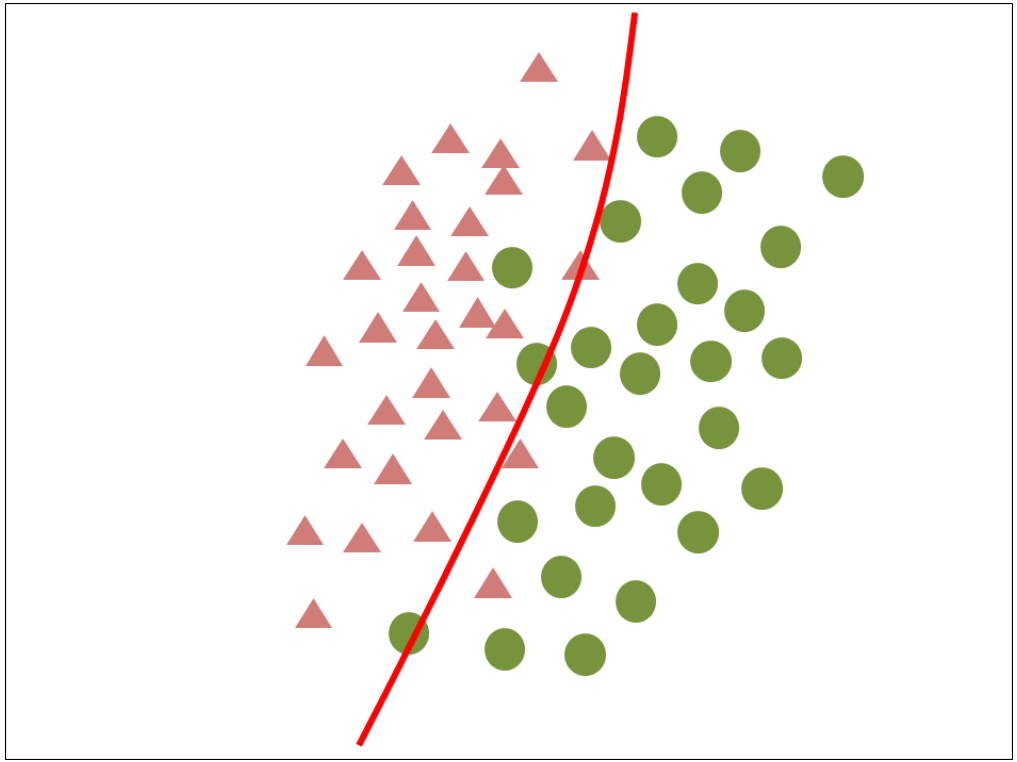
→ Due to “**overfitting**” to training data the classifier may have poor performance with other data

→ **Overfitting must be always checked**!

→ **Independent test dataset** is needed

→ **“In-silico validation”**

Overfitting

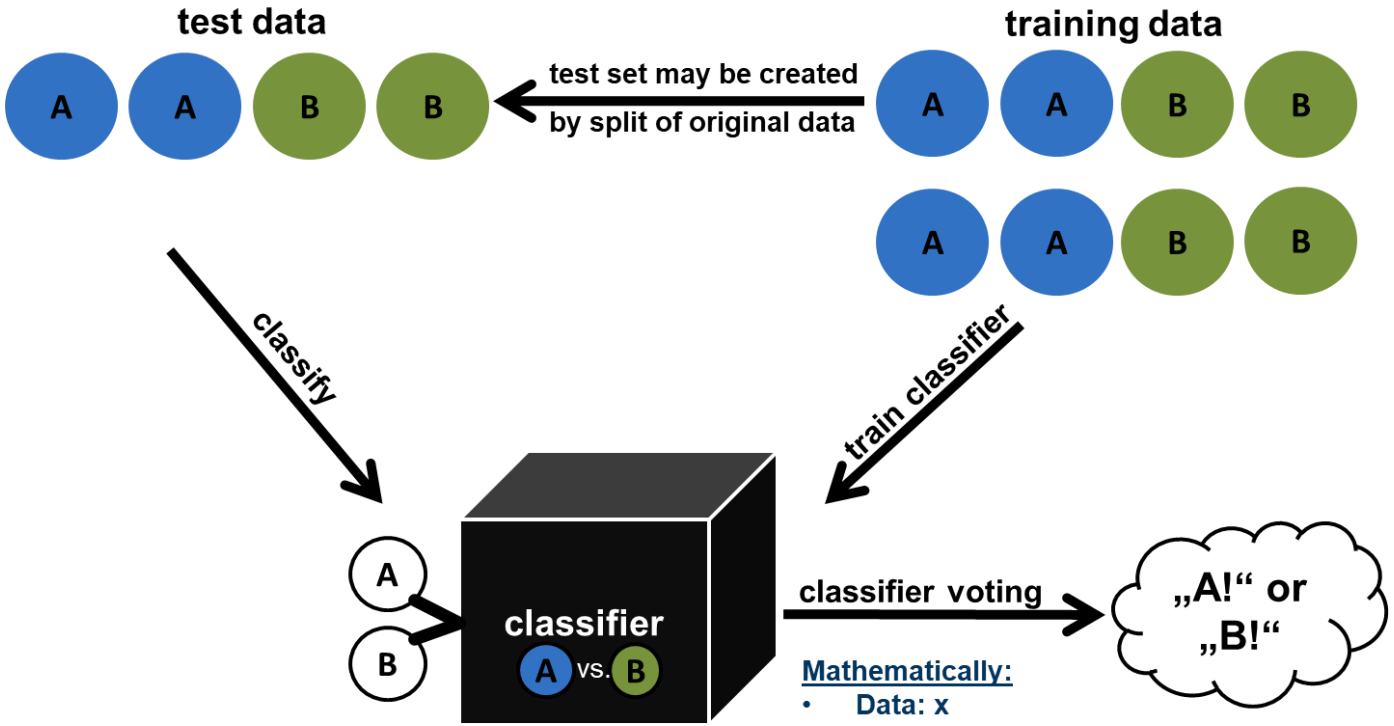


If training data is not representative, often classifiers with relatively rough boundaries have a better average performance with multiple datasets than models with very fine classification boundaries

→ **KISS principle: “Keep it simple, stupid!”**

Controlling overfitting: Training & test data

DDZ Deutsches Diabetes-Zentrum



Mathematically:

- Data: x
- Class label: y ϵ {“A”, “B”}
- Classifier: f(x) = y

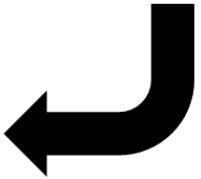
Classification performance: accuracy

DDZ Deutsches Diabetes-Zentrum

$$accuracy := \frac{TP + TN}{TP + FN + FP + TN}$$

$$accuracy = \frac{50 + 50}{50 + 0 + 0 + 50} = 100\%$$

TP:50	FN:0
FP:0	TN:50



examples

$$accuracy = \frac{25 + 25}{25 + 25 + 25 + 25} = 50\%$$

TP:25	FN:25
FP:25	TN:25

$$accuracy = \frac{0 + 0}{0 + 50 + 50 + 0} = 0\%$$

TP:0	FN:50
FP:50	TN:0

A classifier vote can be:

- TP: true positive (an 'A' classified as 'A')
- FN: false negative (an 'A' classified as 'B')
- FP: false positive (a 'B' classified as 'A')
- TN: true negative (a 'B' classified as 'B')

Classification performance: ROC curve

DDZ Deutsches Diabetes-Zentrum

- To compute a **receiver operating characteristic (ROC) curve** the decision threshold of a classifier is varied → classification results changed.
- E.g. a random forest votes '**A**' when $\geq 50\%$ of its trees vote '**A**'. Setting this threshold to 60% chan-ges classification results.
- For each threshold **TPR** & **FPR** is computed to plot the ROC curve. Data points: (FPR, TPR).
- **True positive rate (TPR) = 'sensitivity':**

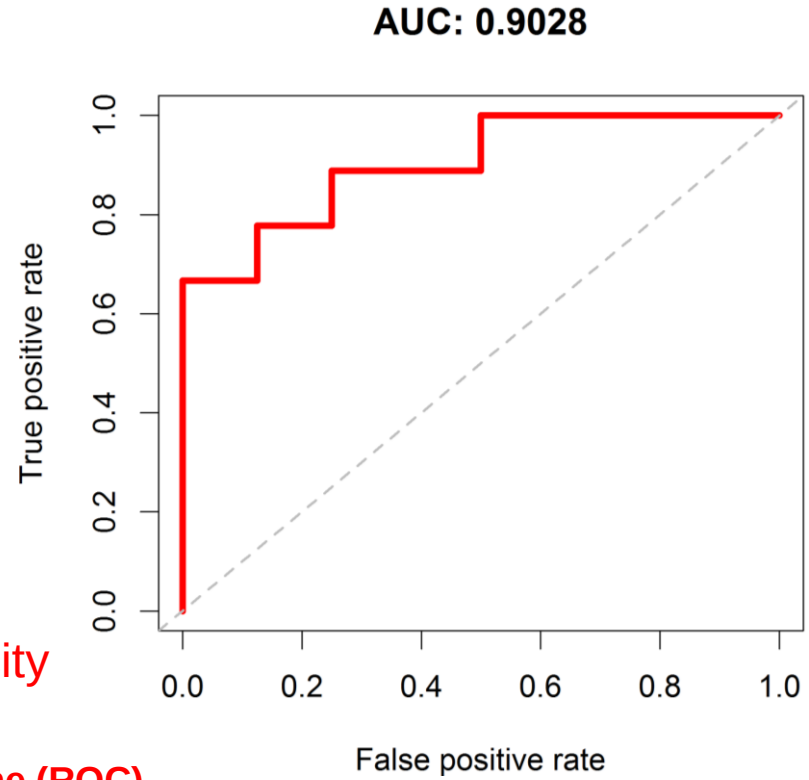
$$TPR := \frac{TP}{TP + FN}$$

- **False positive rate (FPR) = 1-'specificity':**

$$FPR := \frac{FP}{FP + TN} = 1 - \frac{TN}{TN + FP}$$

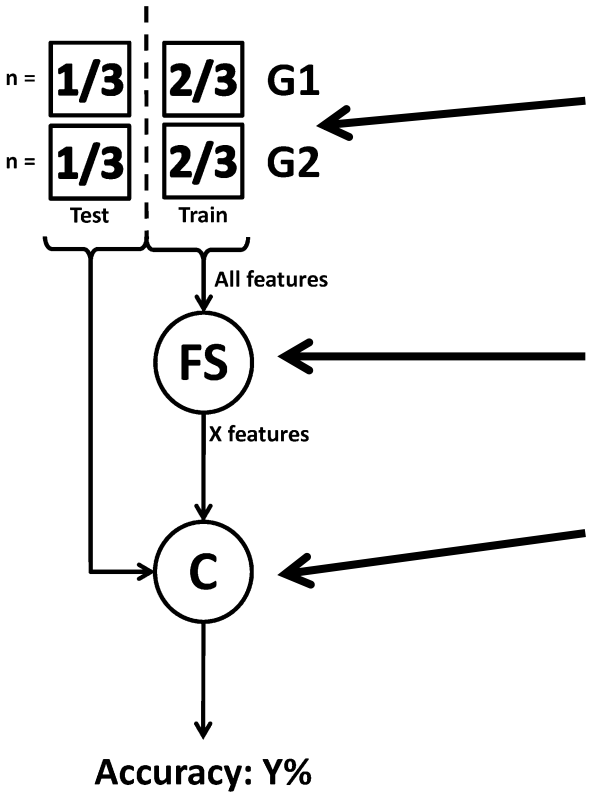
specificity

- Classification performance is represented by the **area under the (ROC) curve**: **'AUC'** or **'AUROC'**



ML: in silico validation

DDZ Deutsches Diabetes-Zentrum

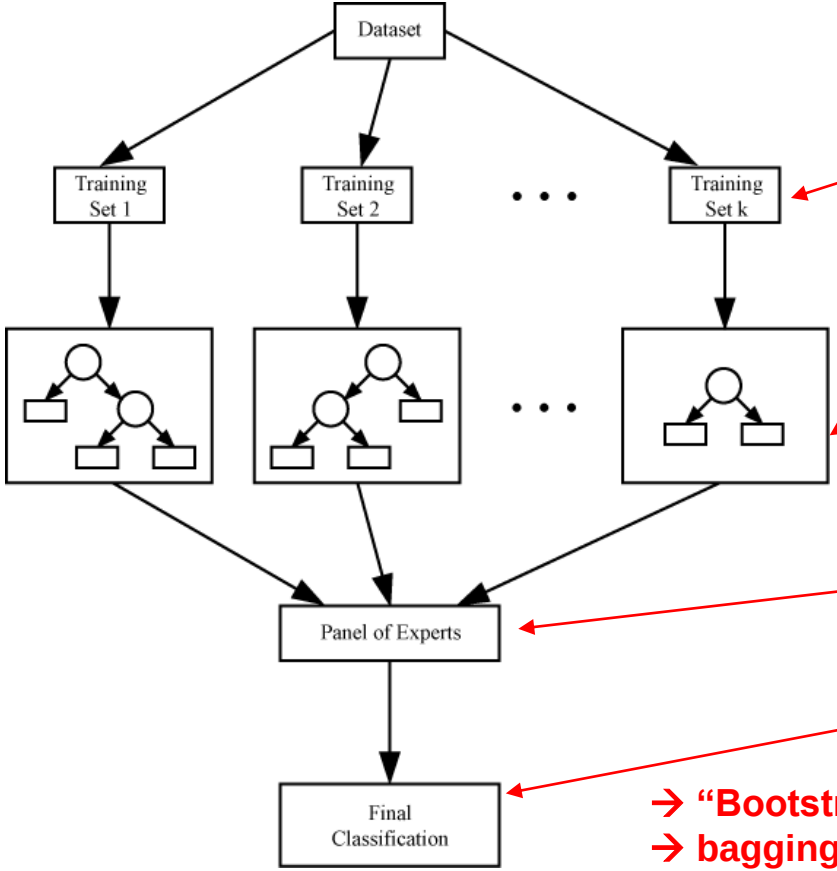


Generate training & test sets by splitting original omics data (e.g., 2/3 of samples for training)

Feature selection

Classifier

Bagging: improving DTs



Generating **k random training sets with same number of samples** as original dataset **via bootstrapping** = **drawing of samples with replacement**. → by drawing some samples again only ca. 2/3 of samples per set are unique → remaining 1/3 = test set

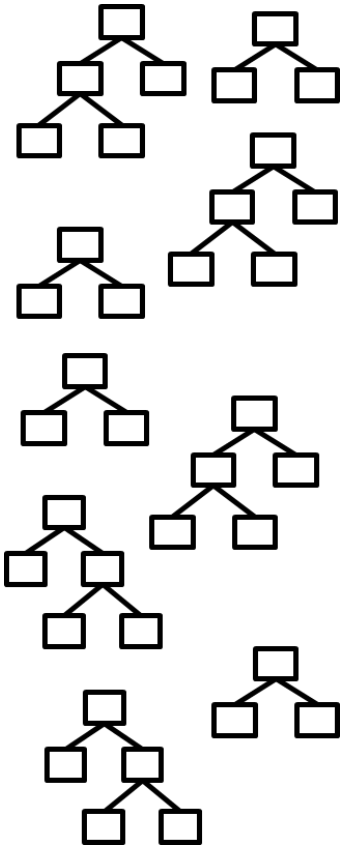
For each training set a distinct decision tree is trained → **k decision trees**

Each decision tree is used for classification of its test set → **k votes combined by an aggregation function** (e.g., mean of votes or majority of votes).

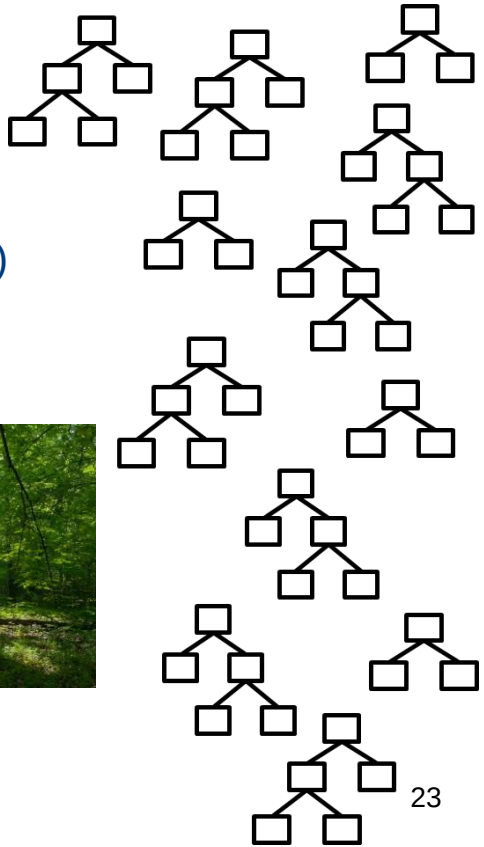
Finally, **one classification result** is returned.

→ **"Bootstrap aggregation" = "bagging", Leo Breiman, 1994**
→ **bagging is a so called ensemble classification method**

Random forest



- Proposed by **Leo Breiman in 1999**
- Based on bagging
- Can be used for classification & regression
- Important parameter: number of trees (often: 500)
- Ø importance measures: Gini impurity, accuracy
- Main advantages
  - More trees = less overfitting
  - Better interpretability than deep learning methods
  - Good classification results
- Main disadvantages
  - Less interpretable than decision trees
  - More trees = growing training times

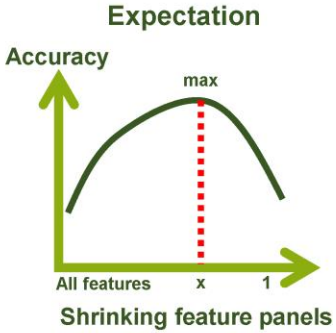


Putting all parts together: Random forest recursive feature elimination (RF-RFE)

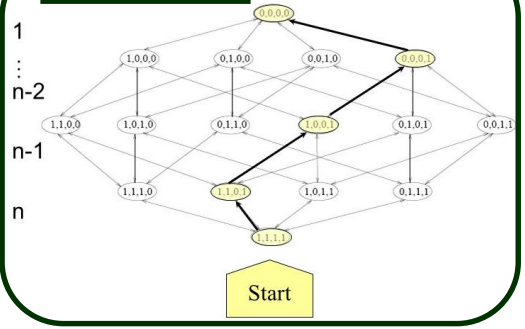
DDZ Deutsches Diabetes-Zentrum

All features

Random forest recursive feature elimination



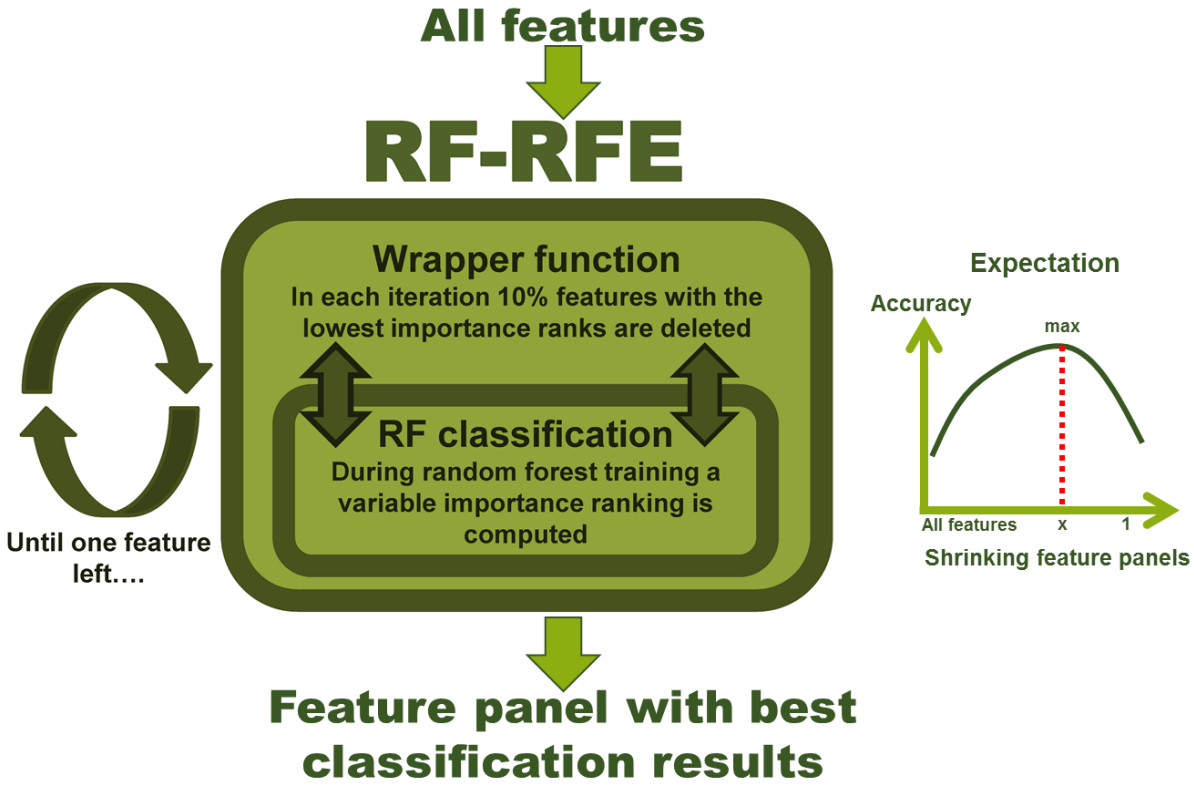
Reminder:



Feature panel with best classification results

RF-RFE

DDZ Deutsches Diabetes-Zentrum

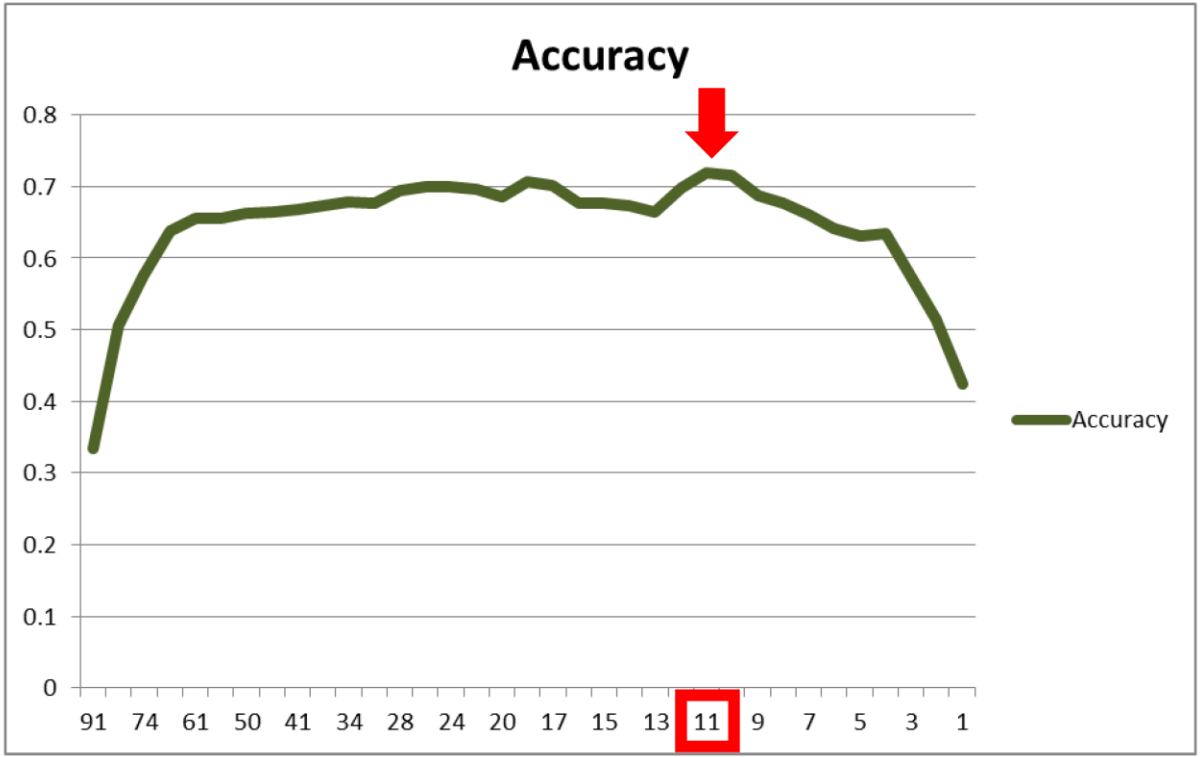


RF-RFE: real-world example (clinical data)

Deutsches Diabetes-Zentrum

- 11 features
  - "X9c.Exc"
  - "doorg.0"
  - "icd1.C18.7"
  - "praether"
  - "chemo1"
  - "polyp1"
  - "opart2a.NA"
  - "lk1"
  - "weither"
  - "chemo2"
  - "radia2"

- Classification accuracy for the panel with 11 features: 71.95 %



Hands on part!

Exercises

DDZ Deutsches Diabetes-Zentrum

- **Exercise 6**
  - https://drive.google.com/drive/folders/1vmewprs0gkpakU8idbgtexDIwmGVUJz3?usp=sharing
  - Use our example dataset from GitHub for the following exercises
  - **Exercise 6.1:** With data sets as small as our example data set, splitting samples into 2/3 training data set and 1/3 test data set for in silico validation is generally not practical. Alternatively, a Leave-One-Out Cross Validation (LOOCV) can be carried out. This involves leaving out a sample, training the model and classifying the omitted sample. This is repeated until every sample has been left out once. The combined classification results are then used to calculate the accuracy. Please run the LOOCV procedure for decision tree training on our example dataset. Please send me your solutions as an ".R"-file

  - **Deadline: 17.12.2024**

Thank you!