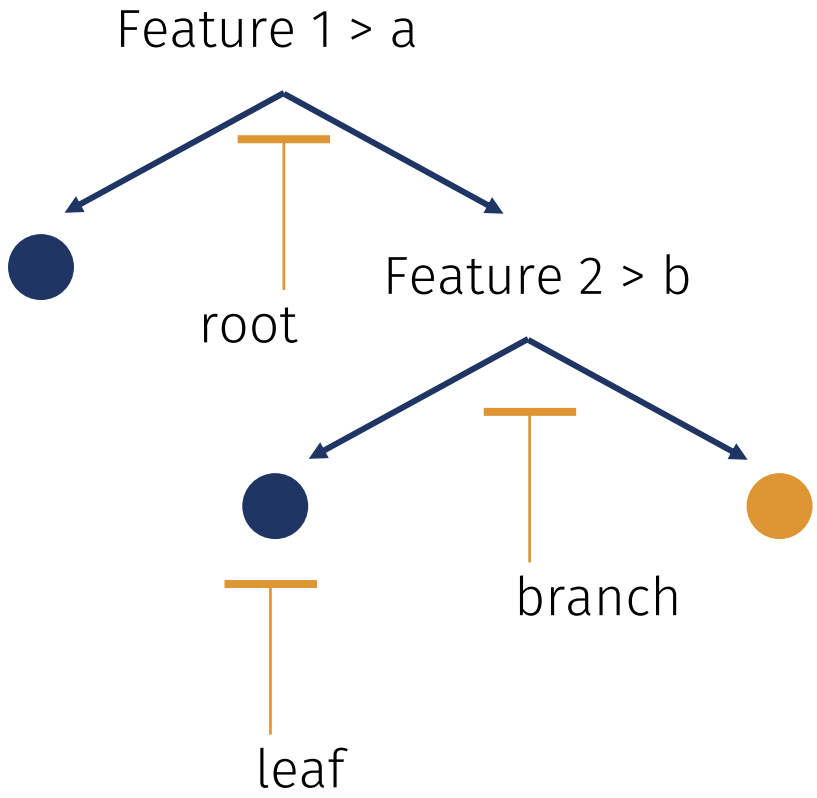
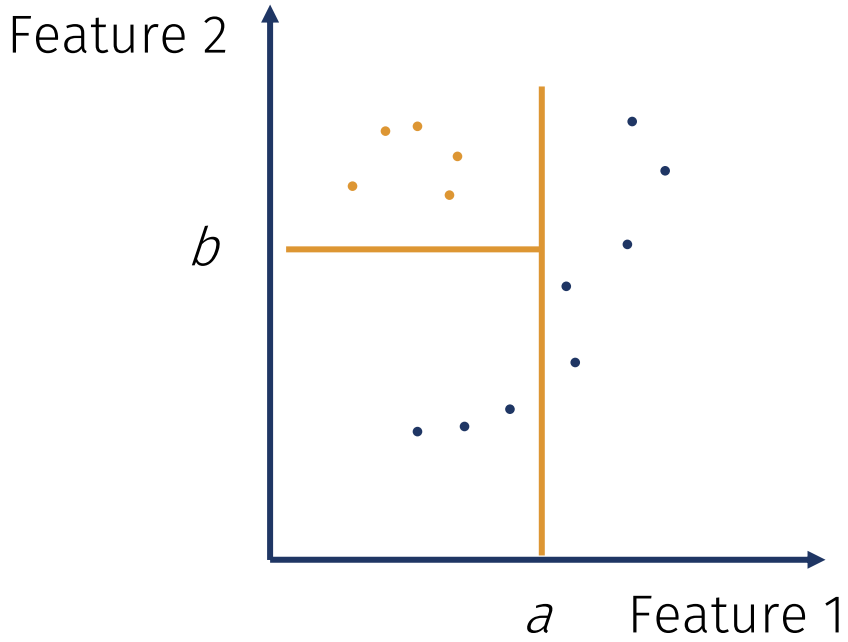
Decision Trees

Decision Trees

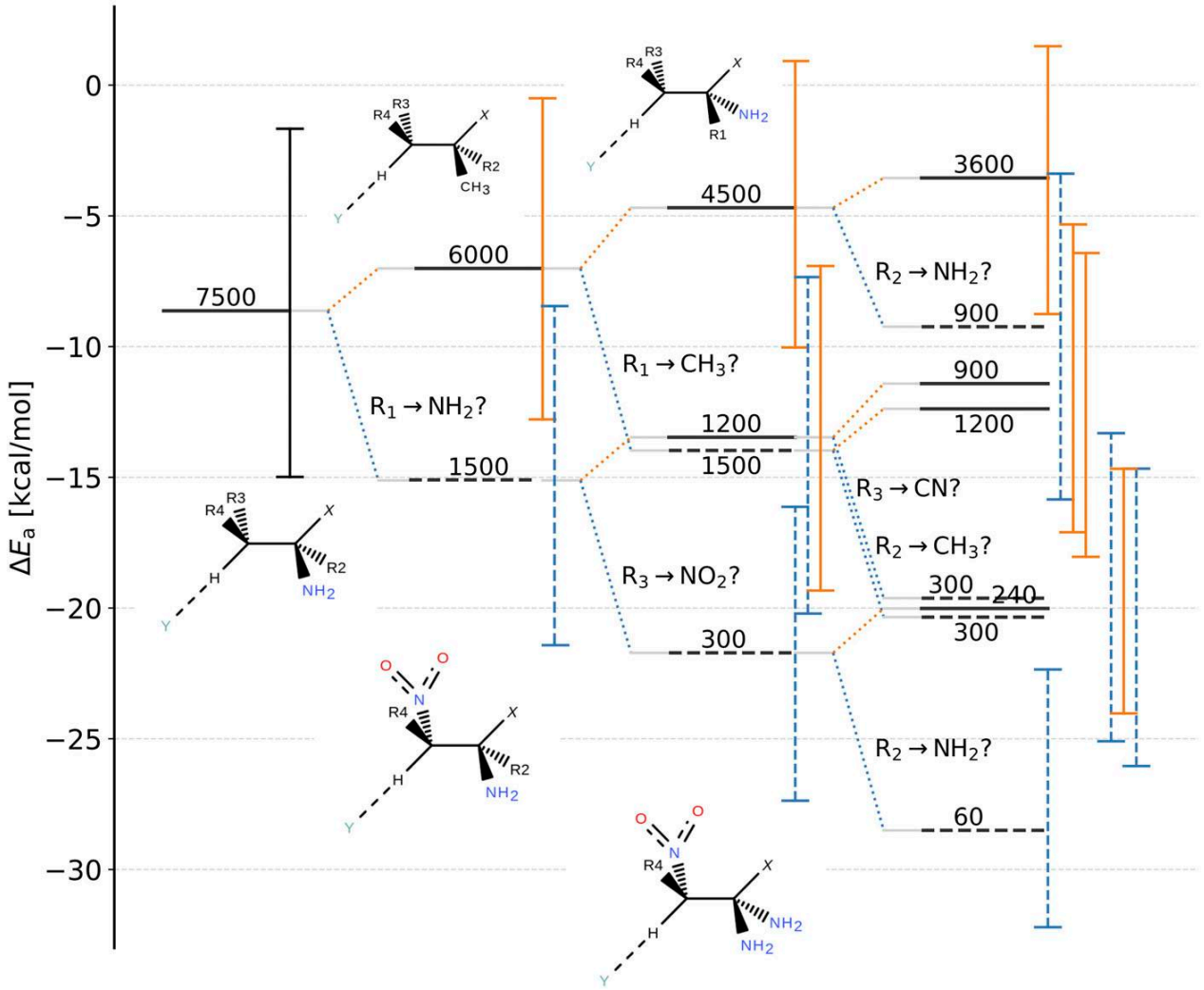
Split data along feature data range, typically binary decisions

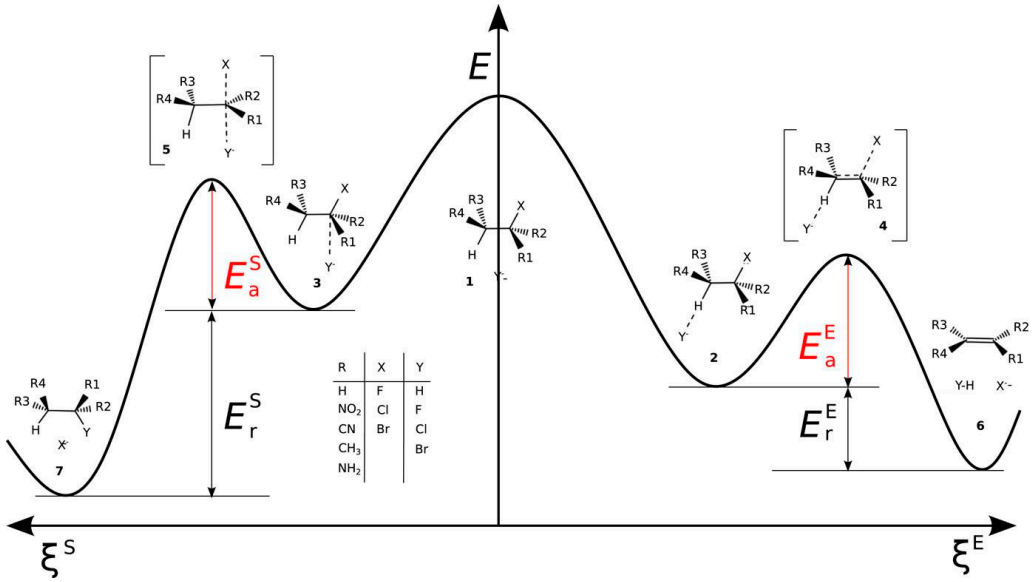
- Scalar: find delimiter
- Discrete: yes/no





Regression example





Heinen et al, *J Chem Phys*, 2021.

When to use

Pros
- Easy to interpret / visualize
- Cheap to use
- No data standardization necessary
- Works well with large amounts of training data (need exponentially more data for another level)
- Flexible: both regression and classification

Cons
- Costly to train
- Prone to overfitting
- Overfitting in high dimensions
- Struggles with „diagonal data“
- Struggles with imbalanced data sets
- Instable under changed of training / randomization

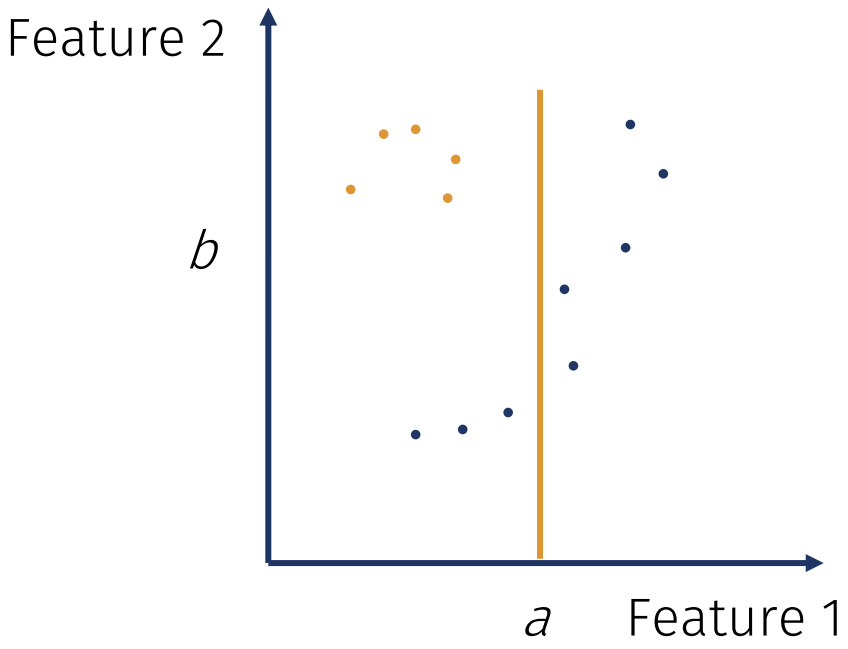
How to fix
Consider random forests
Restrict depth of tree
Subselect features
Transform features with principal components
Subsample
Consider random forests

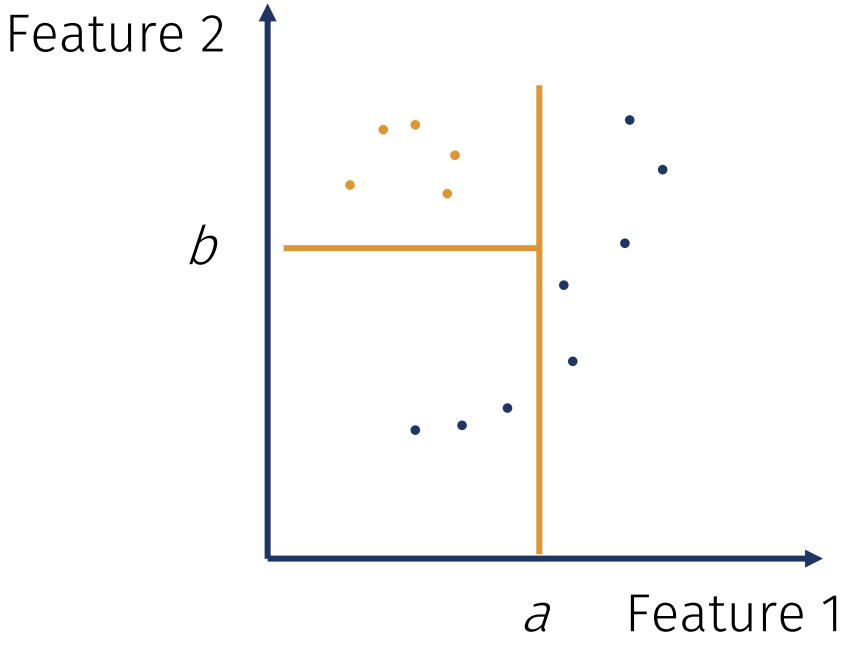
How to build a tree?

Finding the minimal tree is costly.

Heuristic

- Choose feature
- Choose delimiter
- Repeat





Node splitting

A good tree is
- Small
- Important decisions at the root

Typically: maximise information gain from entropy

$$IG(T, A) \equiv H(T) - H(T|A)$$

One decision

Information entropy

Conditional entropy

$$H(T) \equiv -\sum_{t \in \mathcal{T}} p(t) \log p(t)$$

Possible values

Probability

$$H(T|A) \equiv -\sum_{a \in \mathcal{A}, t \in \mathcal{T}} p(a, t) \log \frac{p(a, t)}{p(a)}$$

Conditional probability

Information gain example

Two fair dice, looking for the sum. Which one if rolled first tells us more?

$$Q \equiv D_1 + D_2$$

D_1

0	1

D_2

0	1	2

$$H(T) \equiv -\sum_{t \in \mathcal{T}} p(t) \log p(t)$$

D_1	D_2	Q
0	0	0
0	1	1
0	2	2
1	0	1
1	1	2
1	2	3

$$H(Q) = -\left[\frac{2}{6}\log_2\frac{1}{6} + \frac{4}{6}\log_2\frac{1}{3}\right] \simeq 1.918$$

$$H(D_1) = -\log_2\frac{1}{2} = 1$$

$$H(D_2) = -\log_2\frac{1}{3} \simeq 1.585$$

Information gain example

$Q \equiv D_1 + D_2$

D_1

0	1

D_2

0	1	2

$$H(T|A) \equiv -\sum_{a \in \mathcal{A}, t \in \mathcal{T}} p(a,t) \log \frac{p(a,t)}{p(a)}$$

Q/D_1	0	1	2	3
0	1	1	1	
1		1	1	1

$$H(Q|D_1) = -6 \cdot \frac{1}{6} \log_2 \frac{1/6}{1/2} \simeq 1.585$$

Q/D_2	0	1	2	3
0	1	1		
1		1	1	
2			1	1

$$H(Q|D_2) = -6 \cdot \frac{1}{6} \log_2 \frac{1/6}{1/3} = 1$$

Information gain example

$Q \equiv D_1 + D_2$

D_1

0	1

D_2

0	1	2

$$IG(T, A) \equiv H(T) - H(T|A)$$

$$IG(Q, D_1) = \frac{1}{3}$$

$$IG(Q, D_2) \simeq 0.918$$

Choose the second dice first yields the higher information gain.

Node splitting

Maximize information gain from entropy

$$IG(T, A) \equiv H(T) - H(T|A)$$

- H(T|A)=H(T) if A and T uncorrelated, then IG(T, A) = 0
- Adding or removing unused values t has no effect on H

$$H(T) \equiv -\sum_{t \in \mathcal{T}} p(t) \log p(t)$$

Possible values

Probability

Fixing overfitting

Pruning
- More data
- Simpler trees
  - Restrict during construction (depth, number of data points per leaf)
  - Remove least degrading nodes afterwards

Random forests
- Have many randomized decision trees
- Take majority vote
- An example of an ensemble method

Summary Decision Trees

- Idea: Hierarchy of binary predicate decisions
- Interpretable model, easy to explain to non-experts
- Popular for classification, also works for regression
- Sensitive to actual features
- Hard to condition well
- Works best with plenty data and few features
- Extension: random forests