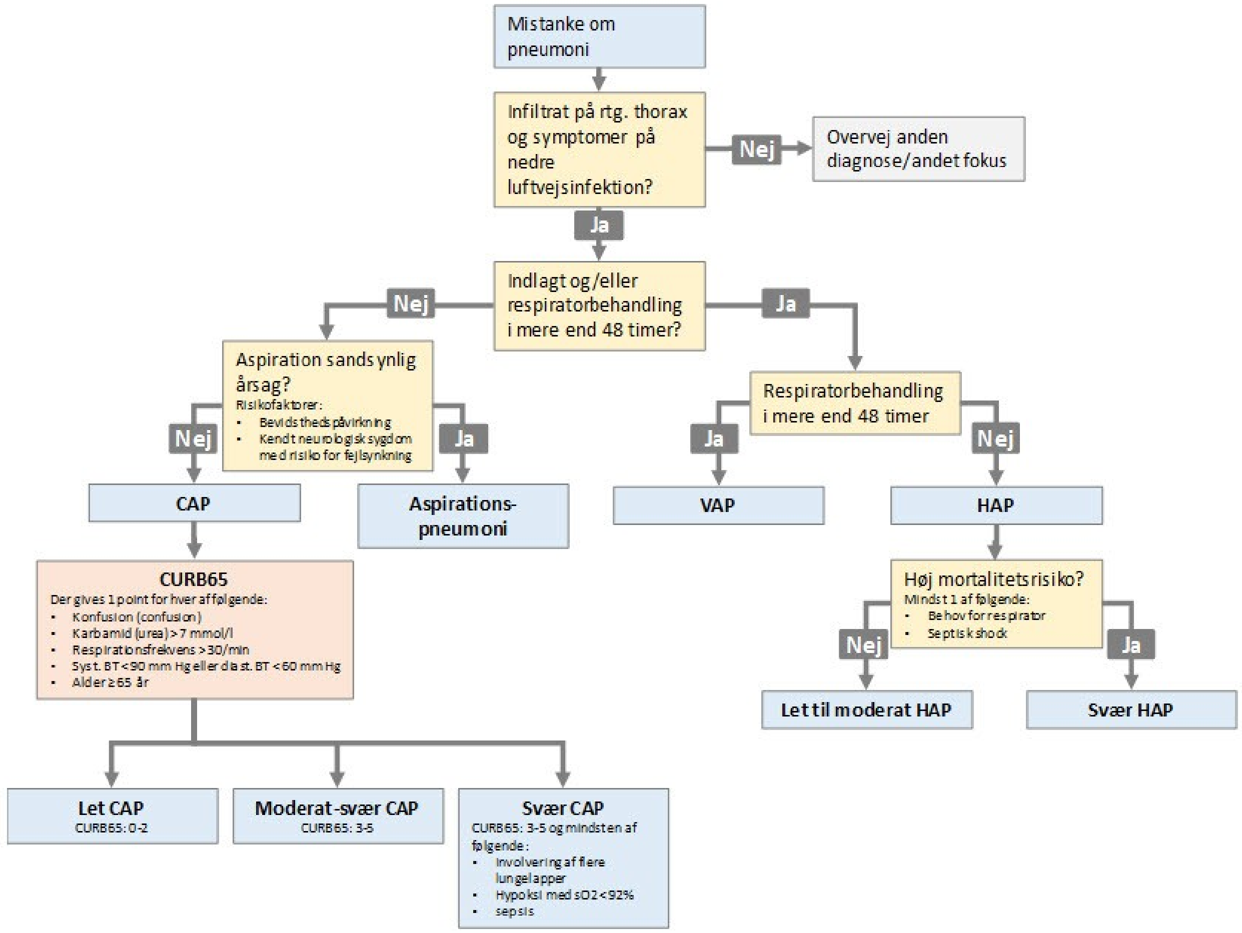
Decision Trees

Image credit: Olga Ernst

“Pneumonia Algorithm”

Community-, Ventilator-, or Hospital-Acquired Pneumonia



Overview

1) Decision trees
2) Univariate
3) Tree induction
4) Classification trees
5) Regression trees
6) Multivariate trees
7) Pruning

Decision Trees

- Nonparametric method
  - Classification
  - Regression

Decision Trees

- Nonparametric method
  - Classification
  - Regression
- Automatic feature extraction
  - Discrete
  - Continuous

Decision Trees

- Nonparametric method
  - Classification
  - Regression
- Automatic feature extraction
  - Discrete
  - Continuous
- Trained tree is efficient

Decision Trees

- *Interpretable*
  - Can be expressed as a set of IF-THEN rules (e.g. IF income < 100K AND savings < 10K THEN High-risk=True)

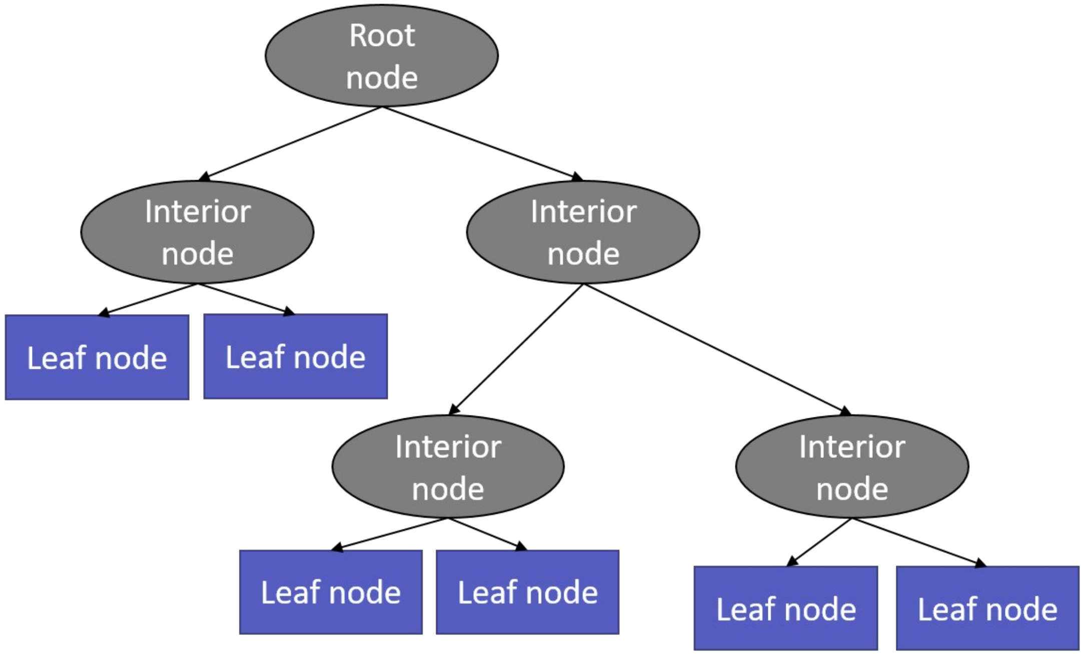
Decision Trees

- *Interpretable*
  - Can be expressed as a set of IF-THEN rules (e.g. IF income < 100K AND savings < 10K THEN High-risk=True)

- Popular
  - Simplicity over performance
  - Many, many variants

Decision Trees

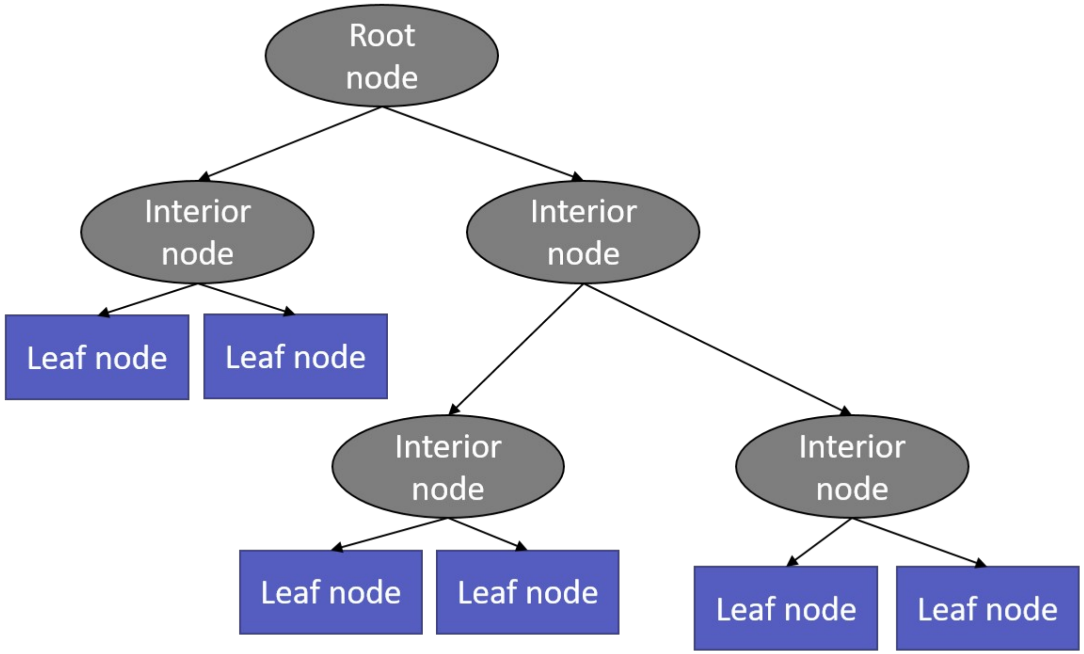
- **Internal (decision) node**
- Splits the data based on test
- Univariate: test is on one attribute of the input, eg:
  - Income > 100K (binary split)
  - Eyecolor {Blue, Brown, Green} (3-way split)



Decision Trees

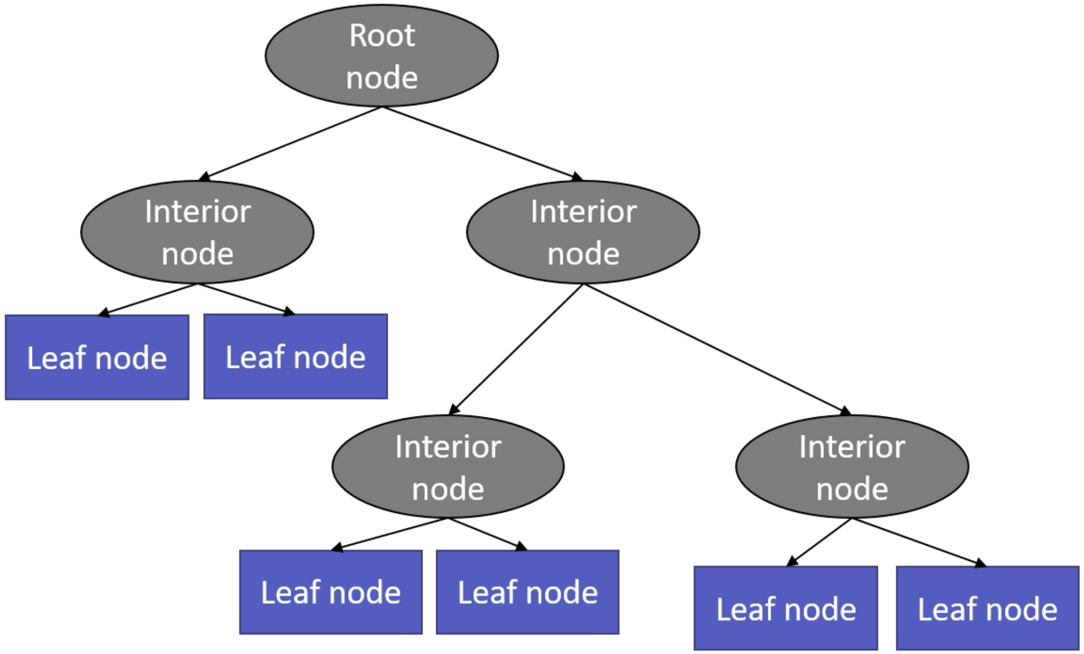
- **Internal (decision) node**
- Splits the data based on test
- Univariate: test is on one attribute of the input, eg:
  - Income > 100K (binary split)
  - Eyecolor {Blue, Brown, Green} (3-way split)

One-level decision tree: *stump*



Decision Trees

- **Internal (decision) node**
- Splits the data based on test
- Univariate: test is on one attribute of the input, eg:
  - Income > 100K (binary split)
  - Eyecolor {Blue, Brown, Green} (3-way split)



One-level decision tree: *stump*

- **Leaf node**
- Contains the final label
- All data points ending up in the same node should ideally have the same label
  - Regression: y = (avg of all training ys in that node)
  - Classification: Class Label

Decision Trees

Mini-exercise: Build a tree

- Based on the examples below
- Classify Imported vs Local fruits/veggies

	Elongated	Green	Big	Class
Watermelon	N	Y	Y	Imported
Orange	N	N	N	Imported
Banana	Y	N	N	Imported
Cucumber	Y	Y	N	Local
Pumpkin	N	N	Y	Local
Pear	N	Y	N	Local

Decision Trees

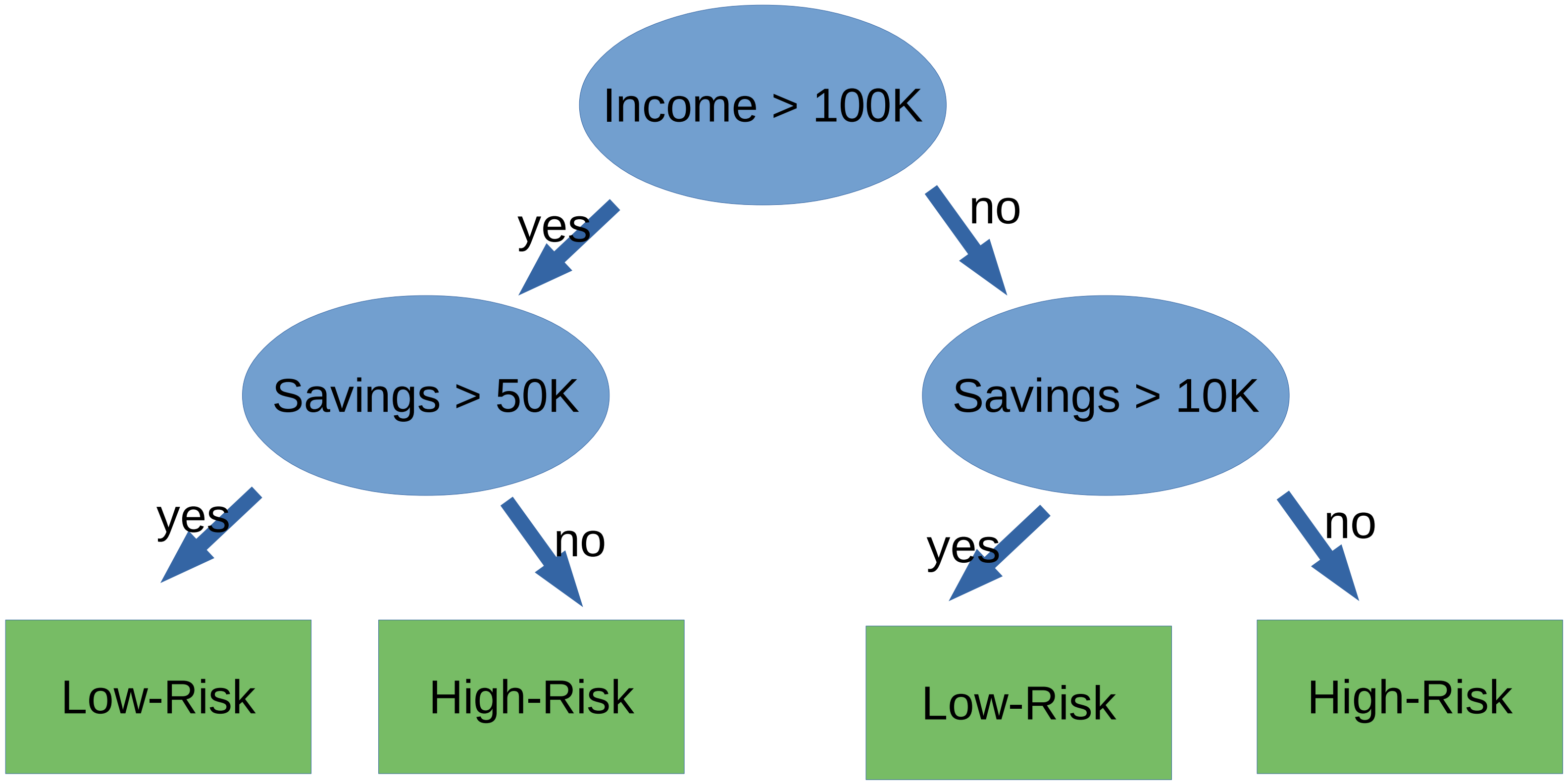
Mini-exercise: Build a tree

- Based on the examples below
- Classify Imported vs Local fruits/veggies

	Elongated	Green	Big	Class
Watermelon	N	Y	Y	Imported
Orange	N	N	N	Imported
Banana	Y	N	N	Imported
Cucumber	Y	Y	N	Local
Pumpkin	N	N	Y	Local
Pear	N	Y	N	Local

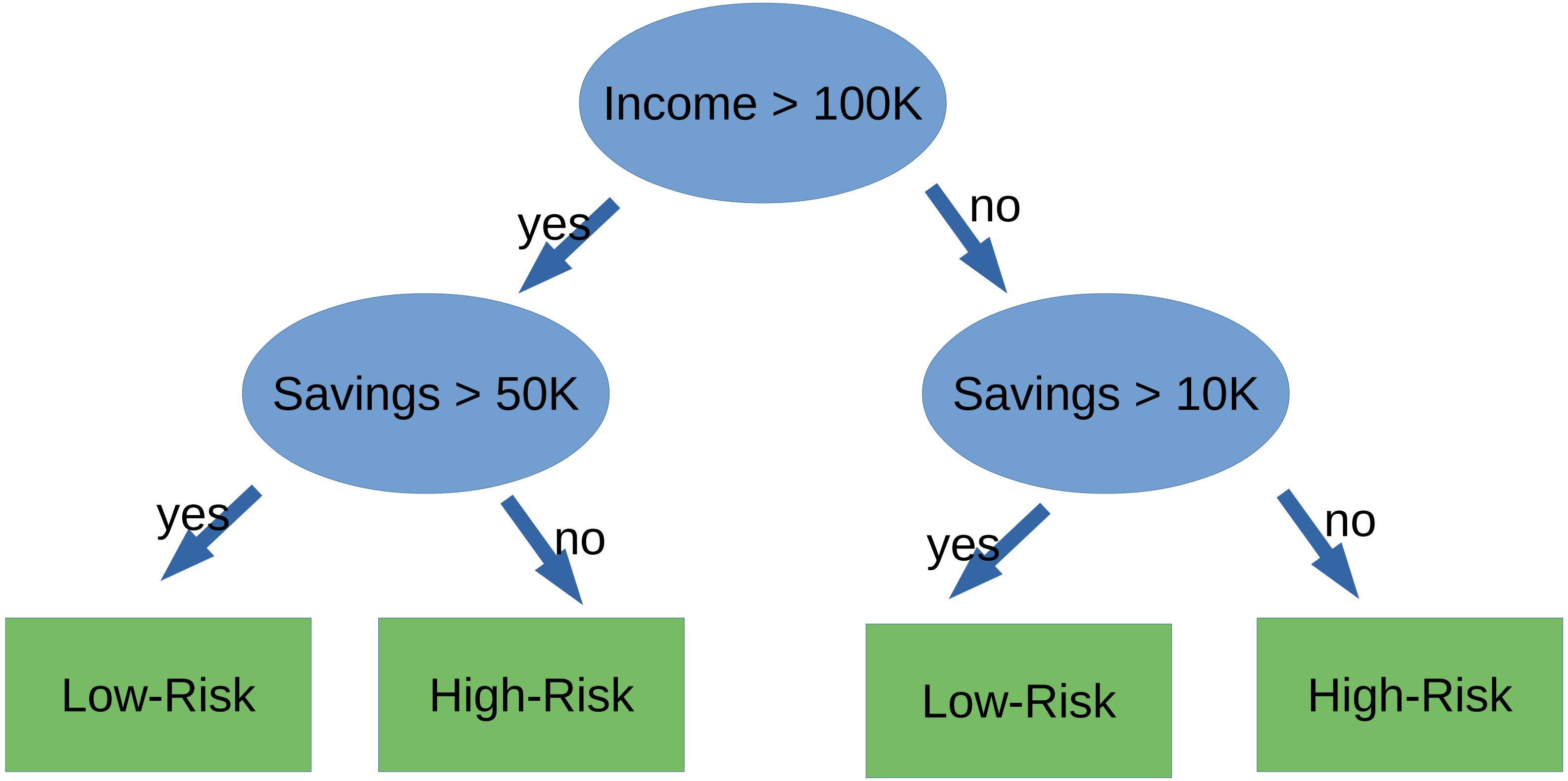
Several possible trees that solve this

Decision Trees



To predict output on a new data instance:
Follow the path from root to leaf

Decision Trees

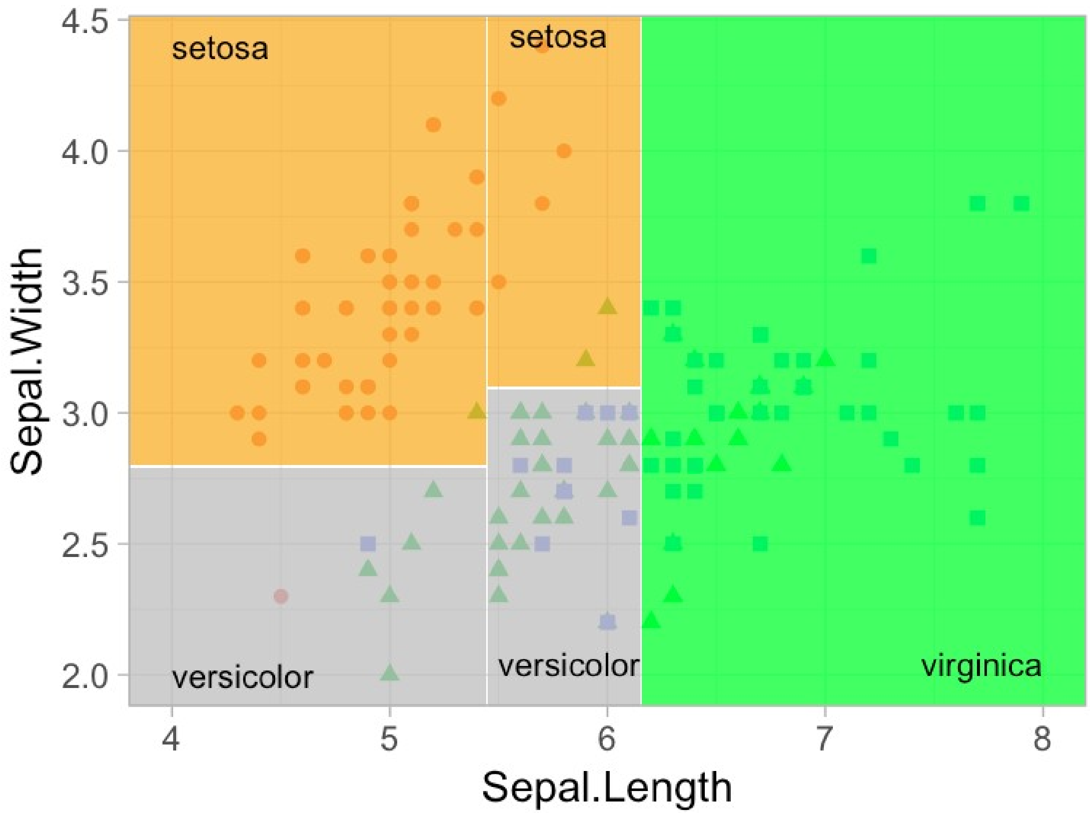


To predict output on a new data instance:
Follow the path from root to leaf

- (Usually) quick traversal
- Need only to save the tree itself

Decision Trees

Leaf nodes divide feature space into bins



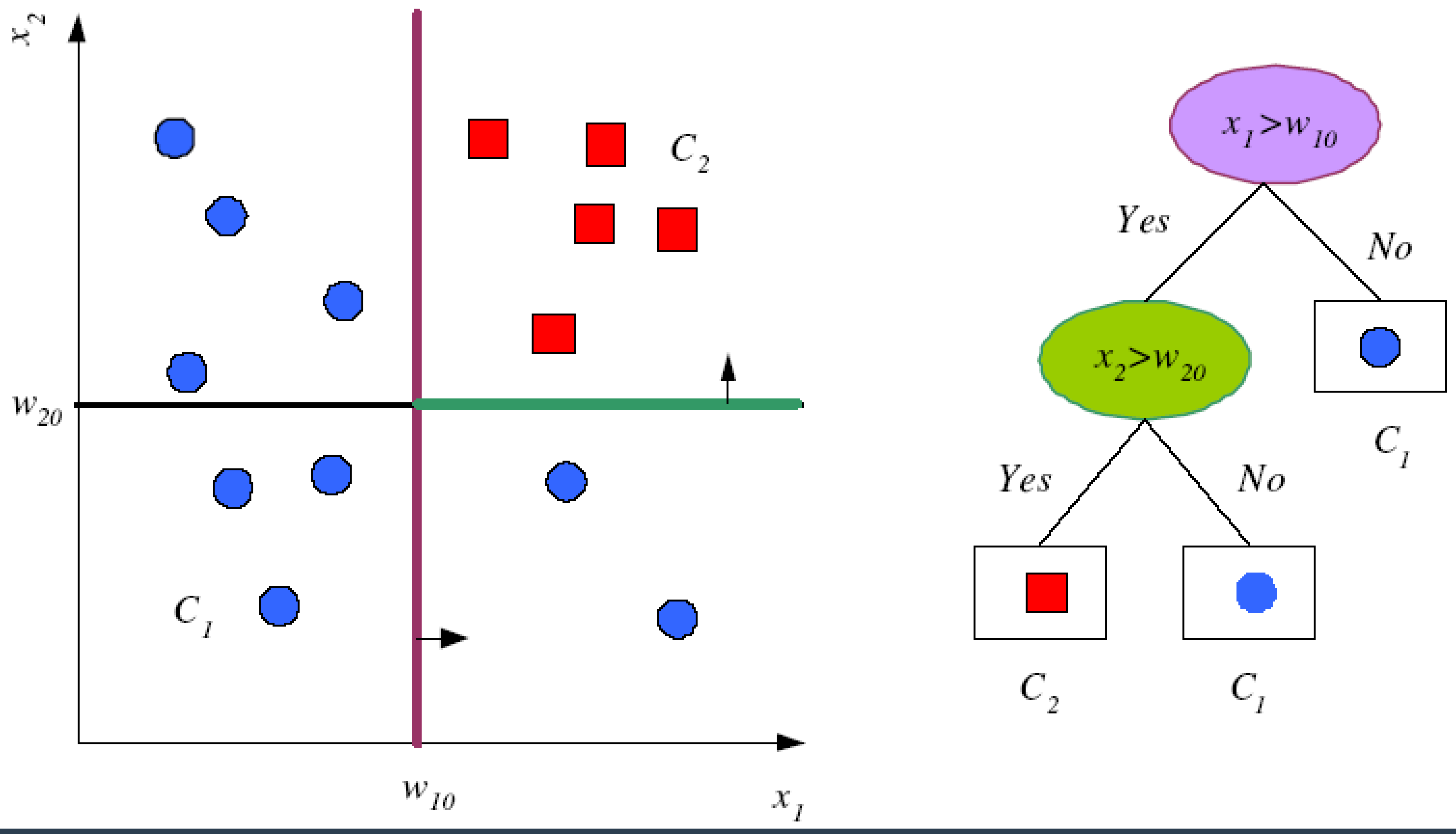
bradleyboehmke.github.io/HOML/DT.html

Univariate Trees

Splits look at only one feature

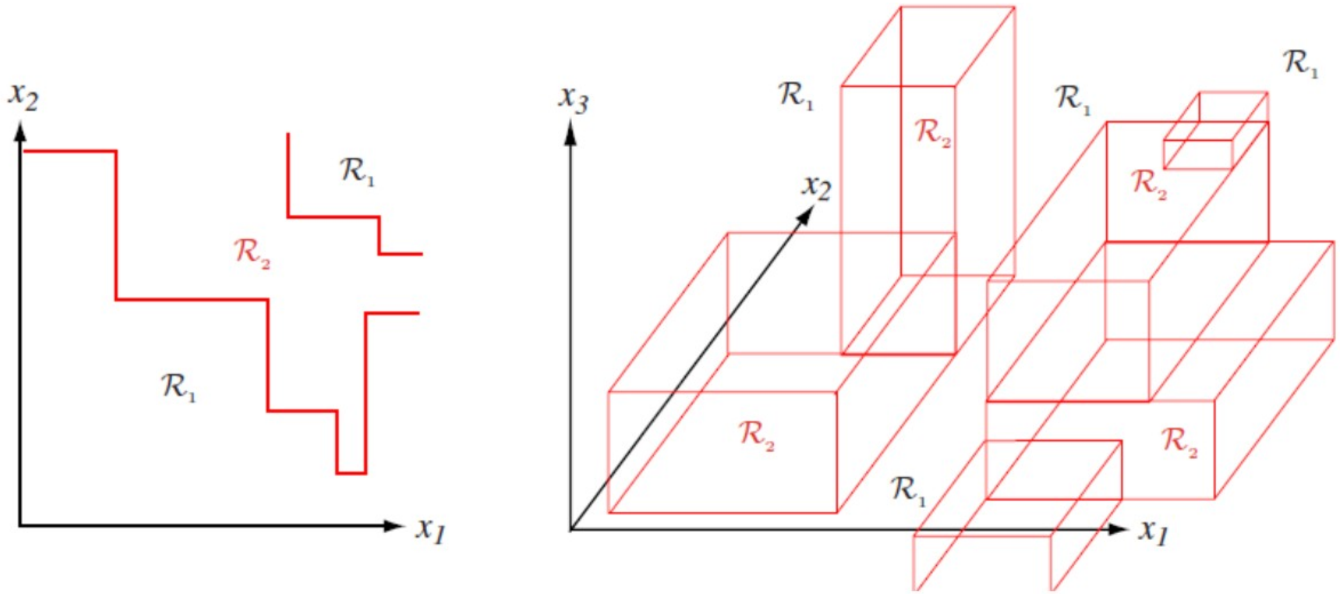
Orthogonal splits: $f_m(\boldsymbol{x}) : x_j > w_{m0}$

Divide the input into two: $L_m = \{\boldsymbol{x} | x_j > w_{m0}\}$ and $R_m = \{\boldsymbol{x} | x_j \leq w_{m0}\}$



Univariate Trees

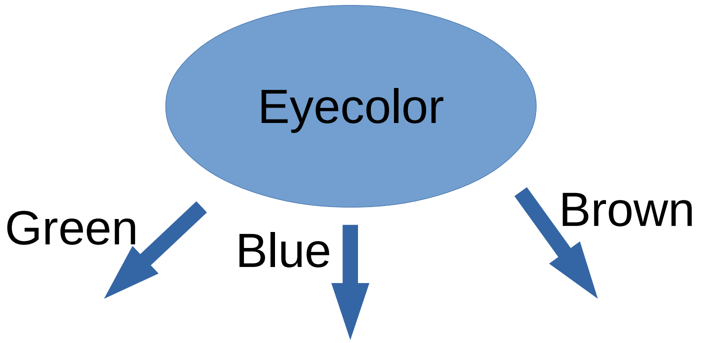
- Example in higher dimension
- Note decision regions need not be continuous



Duda, Hart, Stork; Pattern Classification

Univariate Trees

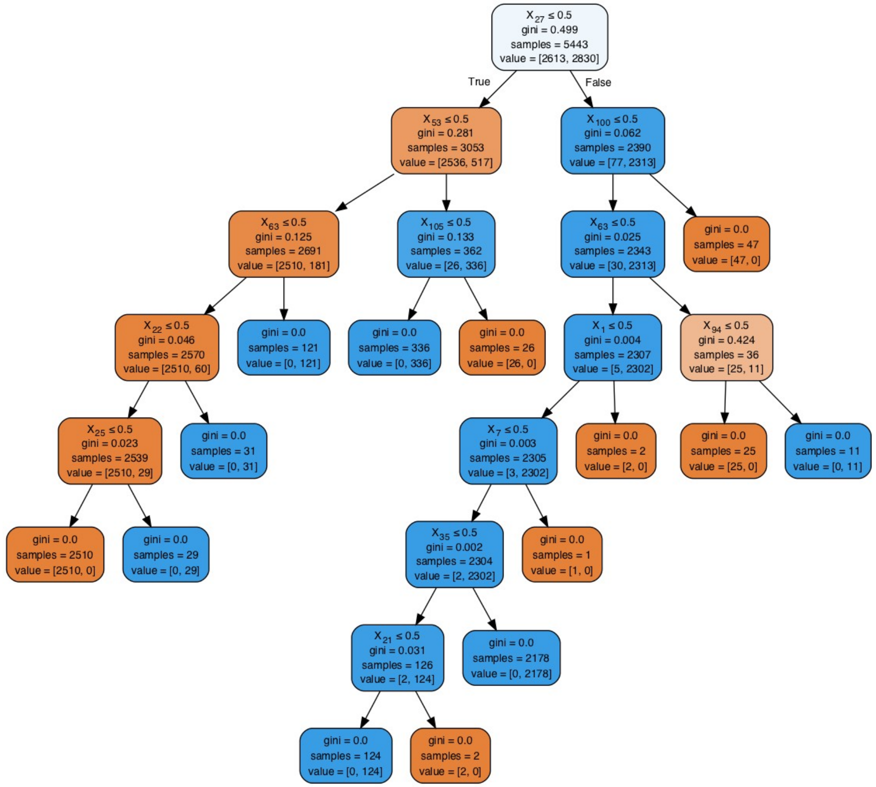
Discrete features: *n*-way splits



Tree induction

Constructing a tree from a given sample

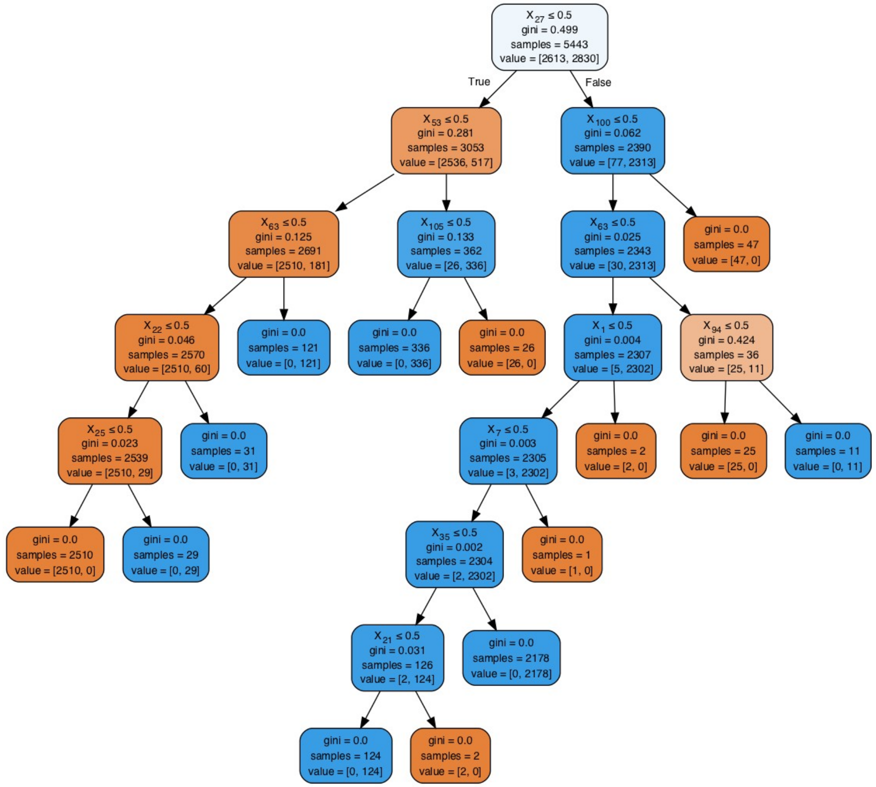
- Find:
    - Number of nodes
    - Tree structure
    - Test in each node
    - Labels of leaf nodes



Tree induction

Constructing a tree from a given sample

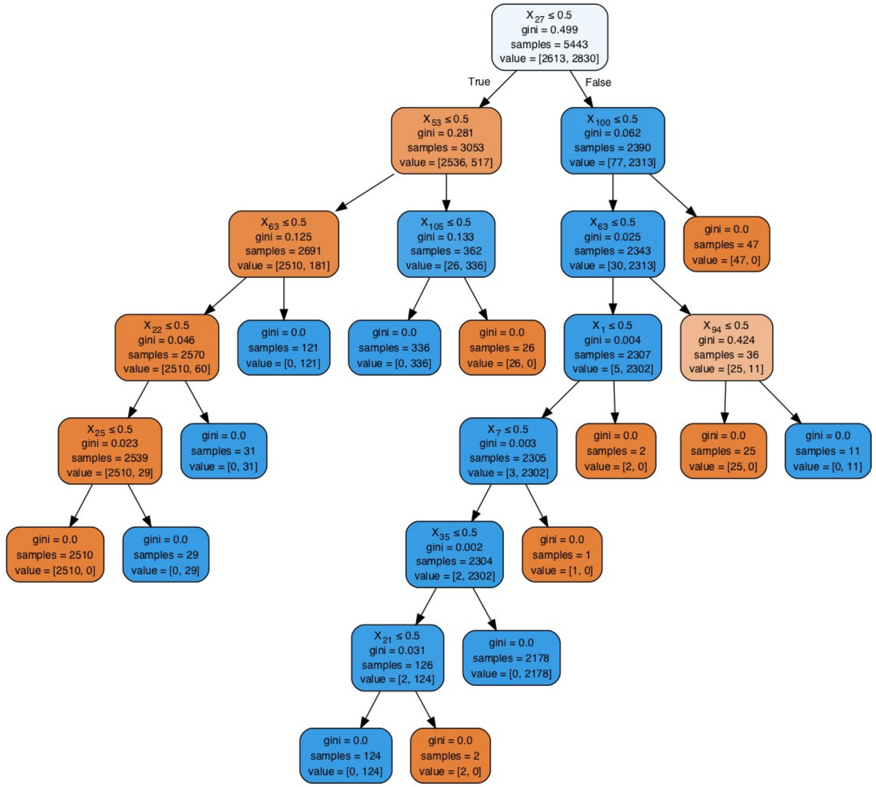
- Find:
  - Number of nodes
  - Tree structure
  - Test in each node
  - Labels of leaf nodes
- There are many possible trees with no error
  - (if they are big enough)



Tree induction

Constructing a tree from a given sample

- Find:
    - Number of nodes
    - Tree structure
    - Test in each node
    - Labels of leaf nodes
- There are many possible trees with no error
    - (if they are big enough)
- Goal: Find the shortest tree that still performs well
- This is NP-complete!



Tree induction

Use greedy/heuristic algorithm based on purity criterion

Tree induction

Use greedy/heuristic algorithm based on purity criterion

- Start at root node with whole training set:
  1) Select best splitting criterion
  2) Divide training set among child nodes accordingly

Tree induction

Use greedy/heuristic algorithm based on purity criterion

- Start at root node with whole training set:
  1) Select best splitting criterion
  2) Divide training set among child nodes accordingly
- If data in child node is pure → leaf
- Else: Repeat recursively for training subset in each child

Tree induction

Use greedy/heuristic algorithm based on purity criterion

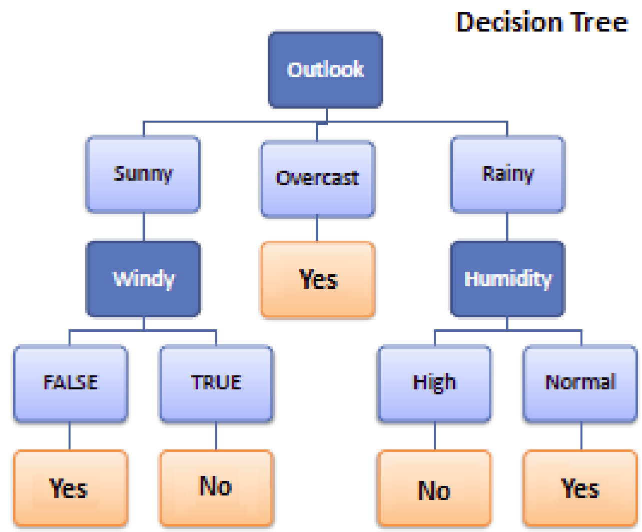
- Start at root node with whole training set:
  1) Select best splitting criterion
  2) Divide training set among child nodes accordingly
- If data in child node is pure → leaf
- Else: Repeat recursively for training subset in each child

Many different implementations, e.g ID3, CART, C4.5

Classification Trees

Predictors | Target

Outlook	Temp.	Humidity	Windy	Play Golf
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes
Rainy	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Sunny	Mild	High	True	No

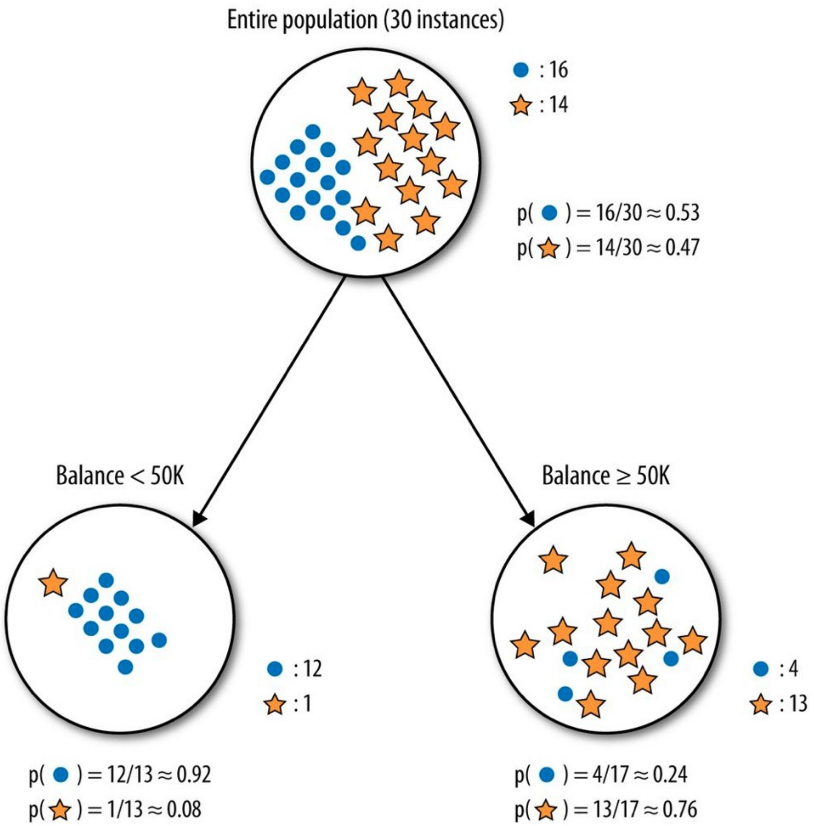


www.saedsayad.com

Classification Trees

Probability of each class at node m:

$$\hat{P}(C_i|\mathbf{x}, m) \equiv p_m^i = \frac{N_m^i}{N_m}$$



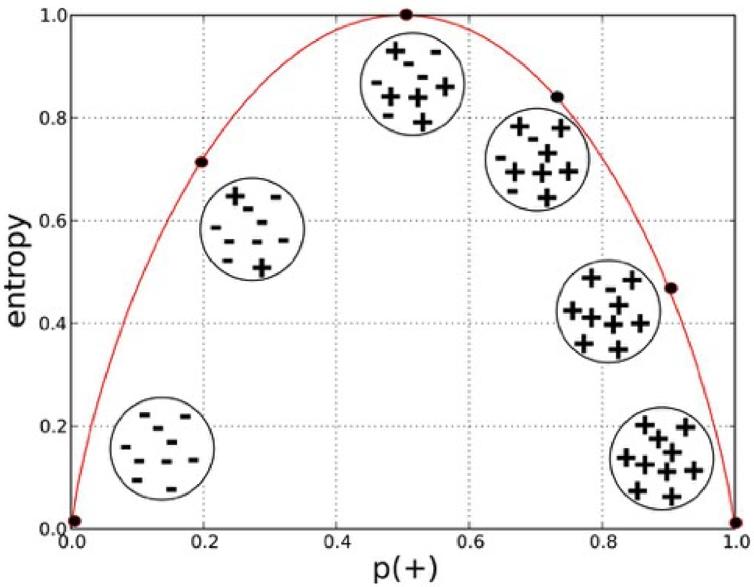
Towardsdatascience.com

Classification Trees

Impurity criterion: Entropy

$$\mathcal{I}_m = -\sum_{i=1}^{K} p_m^i \log_2 p_m^i$$

Alternatives:
- Gini Index
- Misclassification error



Towardsdatascience.com

Classification Trees

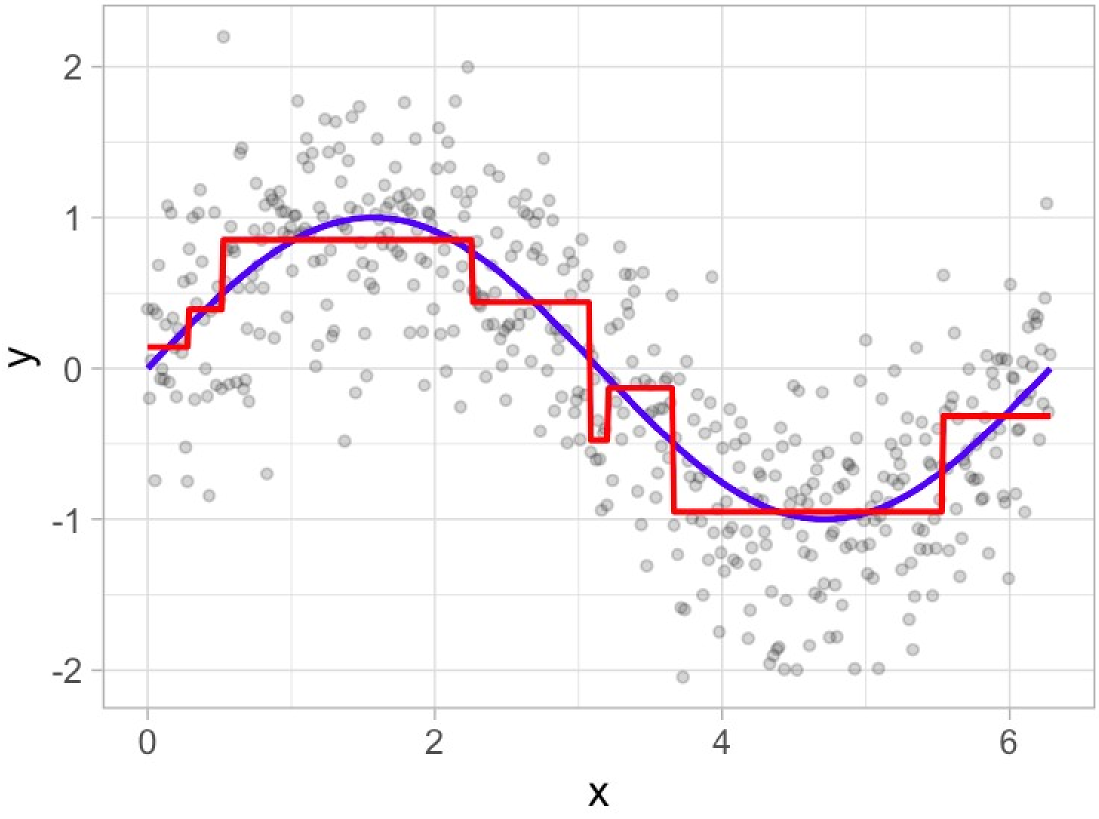
- When a node is sufficiently pure:
  - Stop splitting
  - Make it a leaf node
  - Leaf node can hold most likely class label or class probabilities

Classification Trees

- When a node is sufficiently pure:
  - Stop splitting
  - Make it a leaf node
  - Leaf node can hold most likely class label or class probabilities
- If node is not sufficiently pure:
  - For all features:
    - For all possible splits:
      - Pick split with lowest impurity:
        - For numerical features: N_m-1 possible splits (mid points)
        - For discrete features: n-way split

$$\mathcal{I}'_m = -\sum_{j=1}^{n} \frac{N_{mj}}{N_m} \sum_{i=1}^{K} p^i_{mj} \log_2 p^i_{mj}$$

Regression Trees



bradleyboehmke.github.io/HOML/DT.html

Regression Trees

Similar to classification trees

Purity criterion: Squared error

Regression Trees

Similar to classification trees

Purity criterion: Squared error

$$b_m(\boldsymbol{x}) = \begin{cases} 1 & \text{if } \boldsymbol{x} \in \mathcal{X}_m\text{: } \boldsymbol{x} \text{ reaches node } m \\ 0 & \text{otherwise} \end{cases}$$

$$E_m = \frac{1}{N_m} \sum_t (r^t - g_m)^2 b_m(\boldsymbol{x}^t)$$

Regression Trees

Similar to classification trees

Purity criterion: Squared error

$$b_m(\boldsymbol{x}) = \begin{cases} 1 & \text{if } \boldsymbol{x} \in \mathcal{X}_m\text{: } \boldsymbol{x} \text{ reaches node } m \\ 0 & \text{otherwise} \end{cases}$$

$$E_m = \frac{1}{N_m} \sum_t (r^t - g_m)^2 b_m(\boldsymbol{x}^t)$$

Set g_m to the mean (or median) of r_m

Regression Trees

A node is pure enough when the error is below some threshold

$$E_m < \theta_r$$

The node is then saved as a leafnode with output value g_m

Regression Trees

A node is pure enough when the error is below some threshold

$$E_m < \theta_r$$

The node is then saved as a leafnode with output value g_m

If a node is not pure enough:

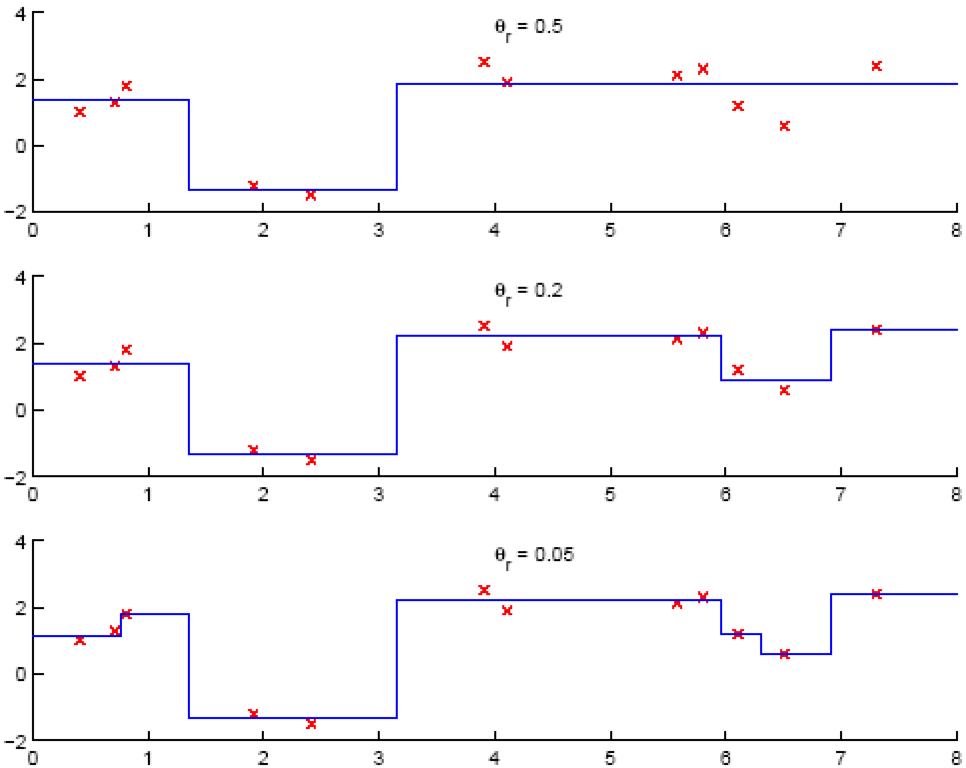
- For all possible features
  - For all possible splits
    - Choose the one with minimum impurity

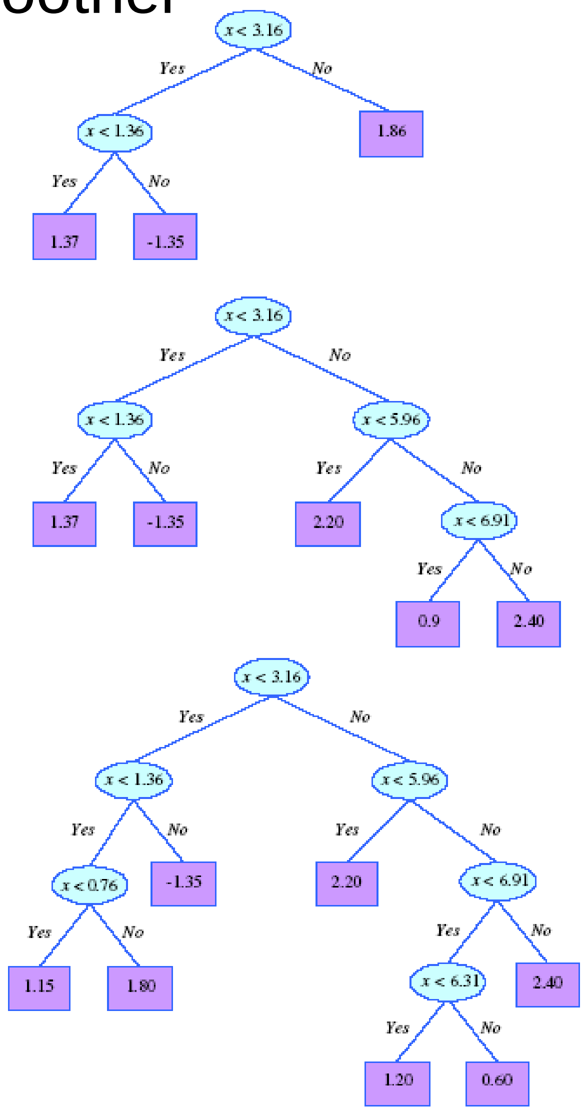
$$E'_m = \frac{1}{N_m} \sum_j \sum_t (r^t - g_{mj})^2 b_{mj}(\mathbf{x}^t)$$

(Sum of the variances (E) of each of the children)

Regression Trees

The threshold value functions as a smoother





Regression Trees

Other possible error function:

$$E_m = \max_j \max_t |r^t - g_{mj}| b_{mj}(\boldsymbol{x}^t)$$

Other possible output function:

$$g_m(\boldsymbol{x}) = \boldsymbol{w}_m^T \boldsymbol{x} + w_{m0}$$

Multivariate Trees

- Allow splits that are not orthogonal to the axes
- Fewer but more complex decision nodes

$$f_m(\boldsymbol{x}) : \boldsymbol{w}_m^T \boldsymbol{x} + w_{m0} > 0$$

Multivariate Trees

- Allow splits that are not orthogonal to the axes
- Fewer but more complex decision nodes

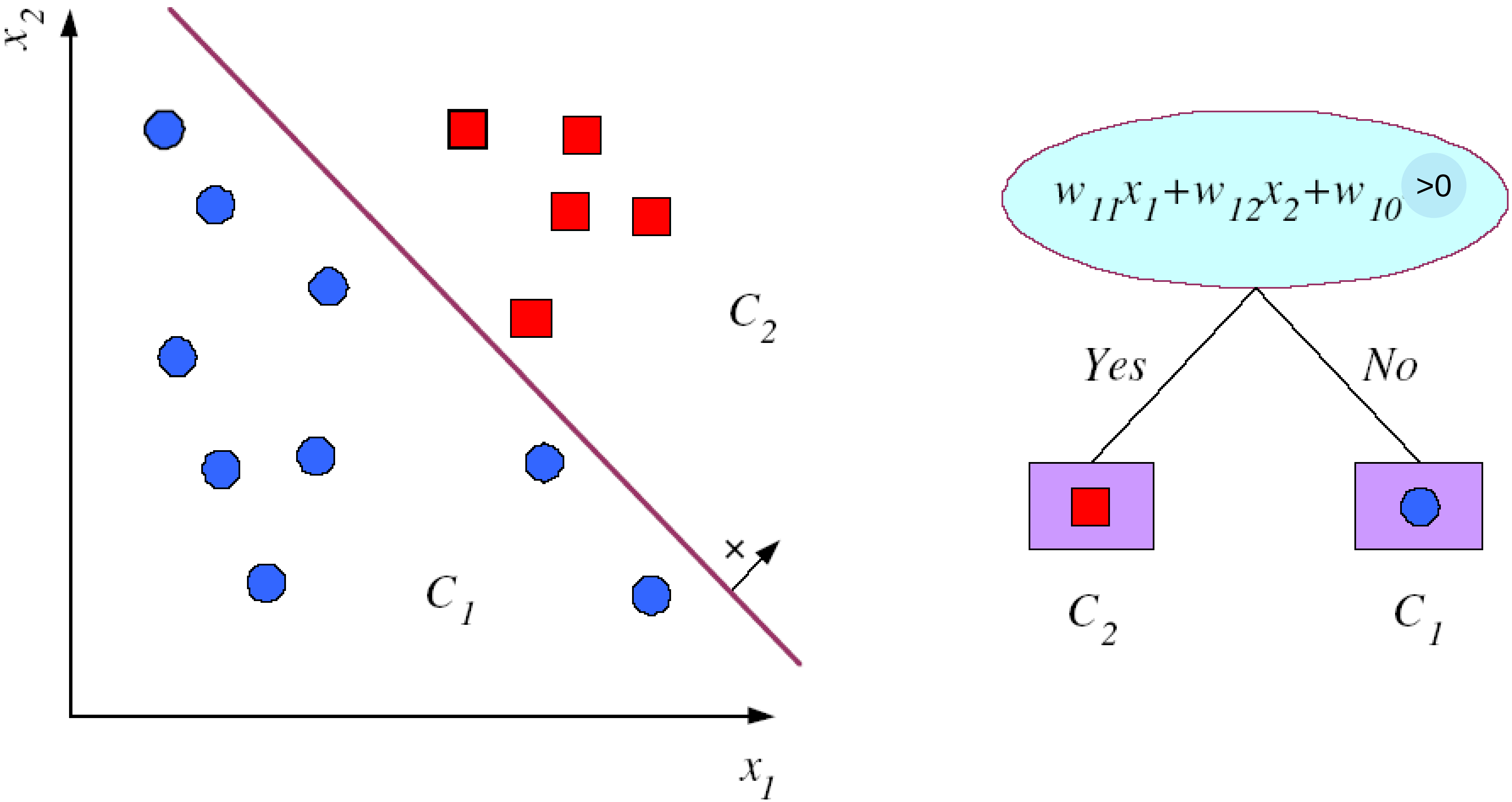
$$f_m(\boldsymbol{x}) : \boldsymbol{w}_m^T \boldsymbol{x} + w_{m0} > 0$$

- Decision hyperplanes
- Decision bins are polyhedra
- Exhaustive search not practical

Multivariate Trees

Allow splits that are not orthogonal to the axes

$$f_m(\boldsymbol{x}) : \boldsymbol{w}_m^T \boldsymbol{x} + w_{m0} > 0$$



Recap

Bias and Variance

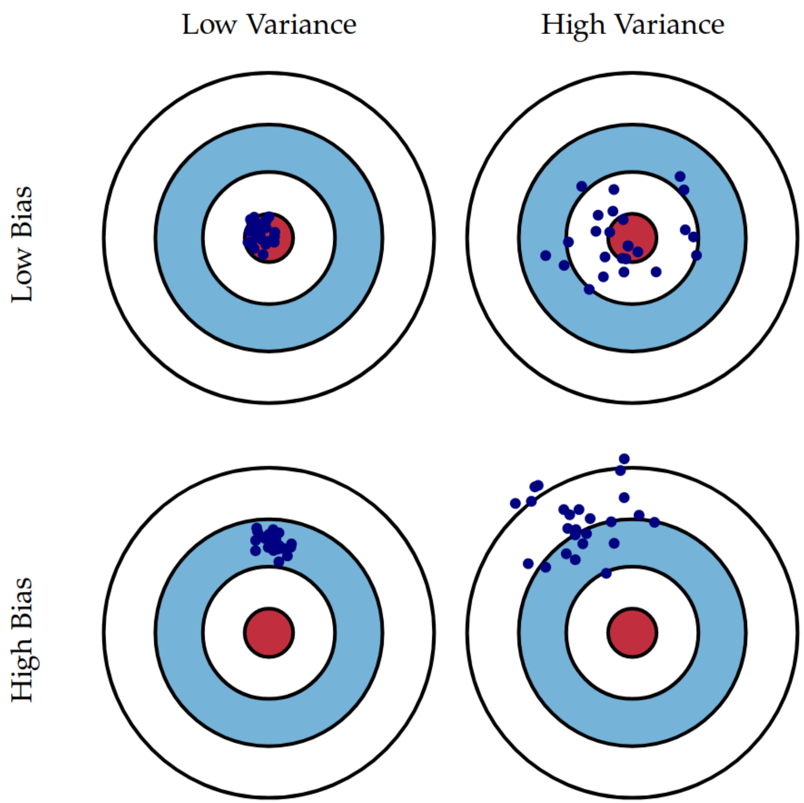
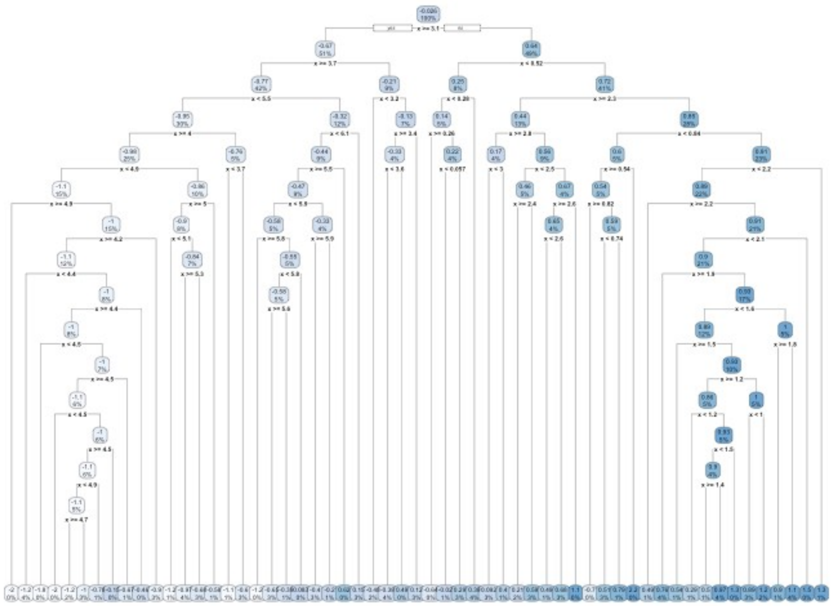


Fig. 1 Graphical illustration of bias and variance.

http://scott.fortmann-roe.com/docs/BiasVariance.html

Variance

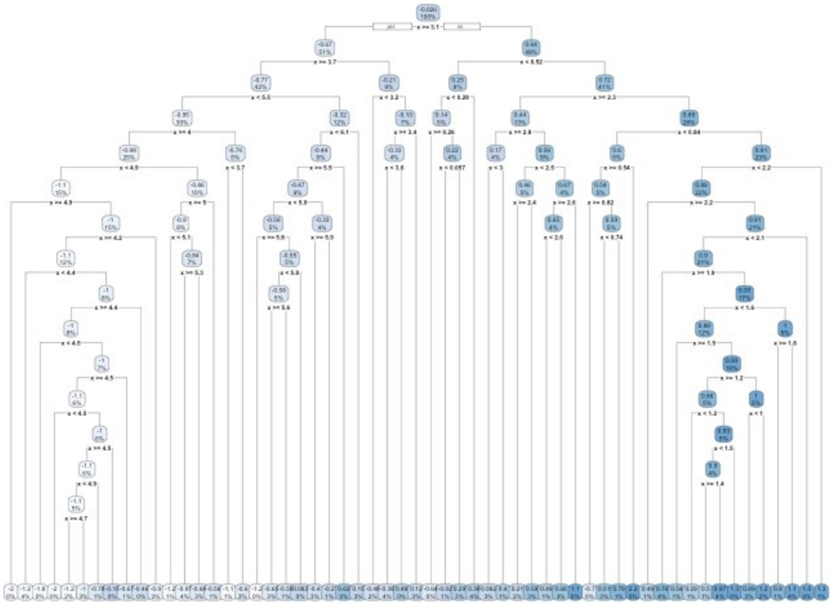
- Decision trees can always be grown to complete purity
- Potentially O(N) nodes!
  - Memorising training set, poor generalization
  - Very prone to overfitting
  - High variance



bradleyboehmke.github.io/HOML/DT.html

Early stopping

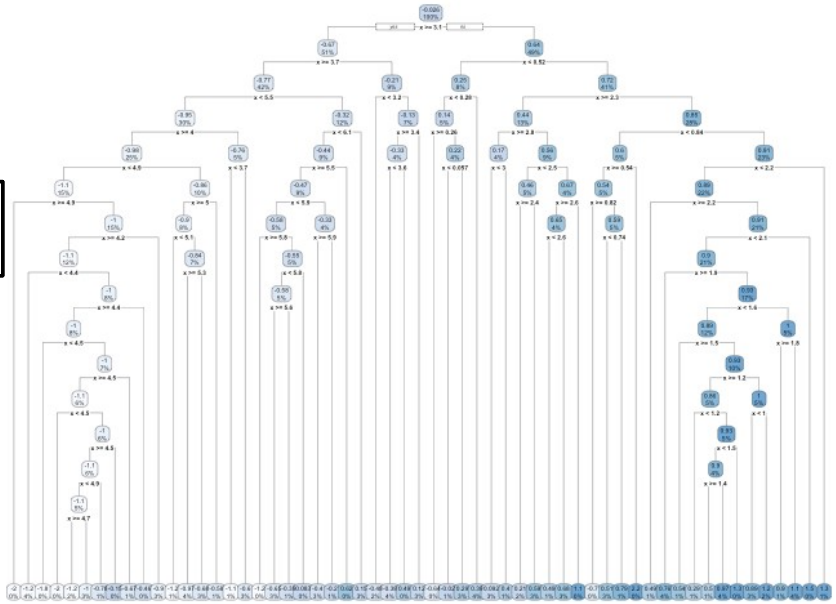
- Stop tree from growing too large ("prepruning")
- A way to prevent overfitting
  - Max tree depth
  - Minimum number terminal node size



Early stopping

- Stop tree from growing too large (“prepruning”)
- A way to prevent overfitting
  - Max tree depth
  - Minimum number terminal node size

Too restricted trees have high bias



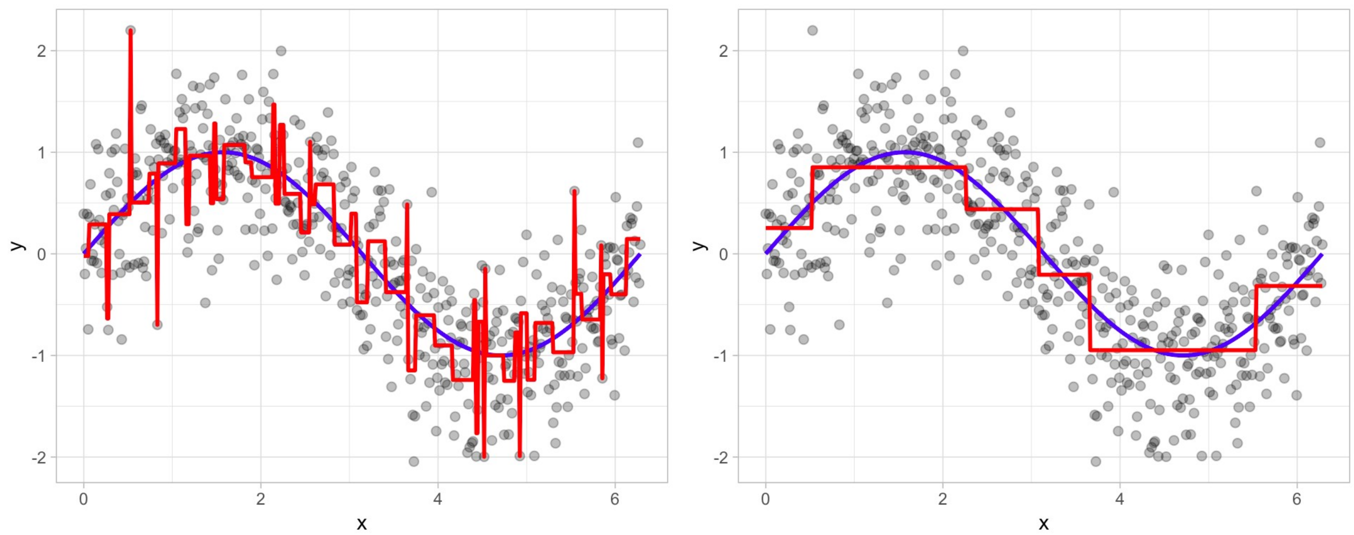
bradleyboehmke.github.io/HOML/DT.html

Pruning

- Grow a large tree, then prune it back (postpruning)
- Remove subtrees that are overfitting

Pruning

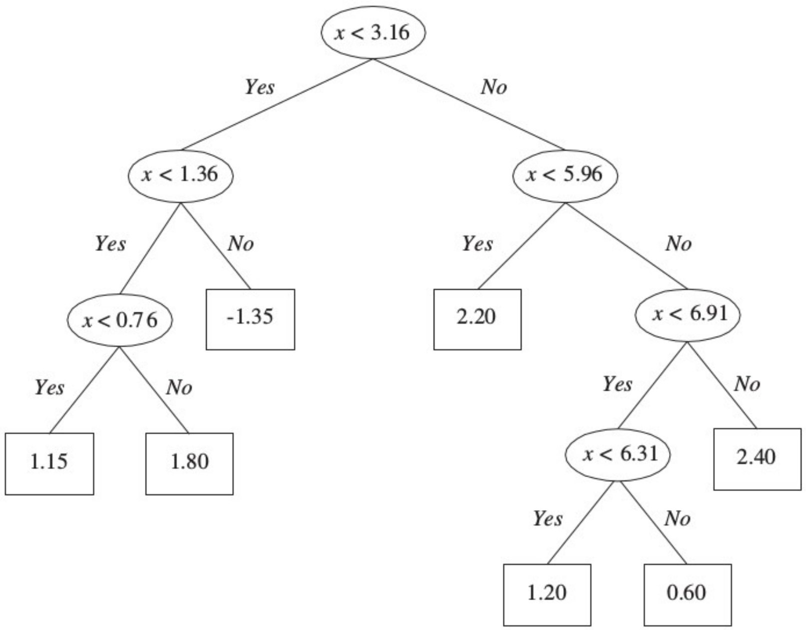
- Grow a large tree, then prune it back (postpruning)
- Remove subtrees that are overfitting
- Set aside a *pruning set* from the training data
- For each subtree:
    - Collapse the subtree to a leafnode
    - Compare performance on the pruning set
    - If the leafnode is as good as the subtree:
        - Discard the subtree, keep the new leafnode

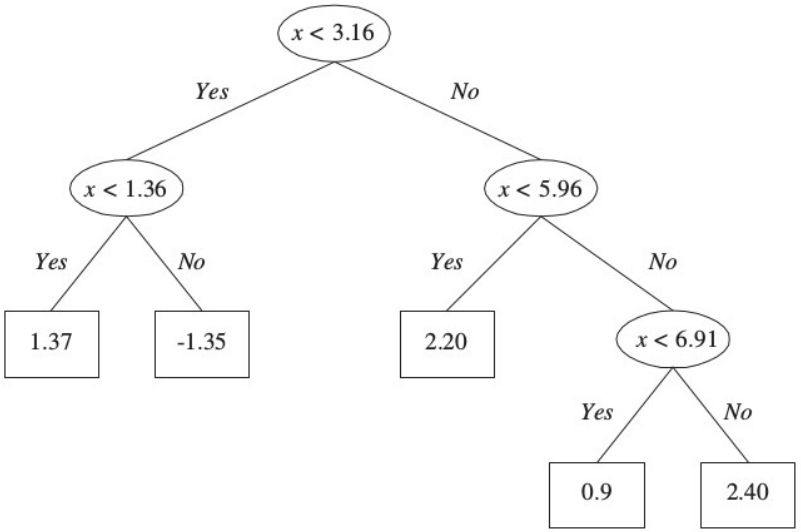


bradleyboehmke.github.io/HOML/DT.html

Pruning

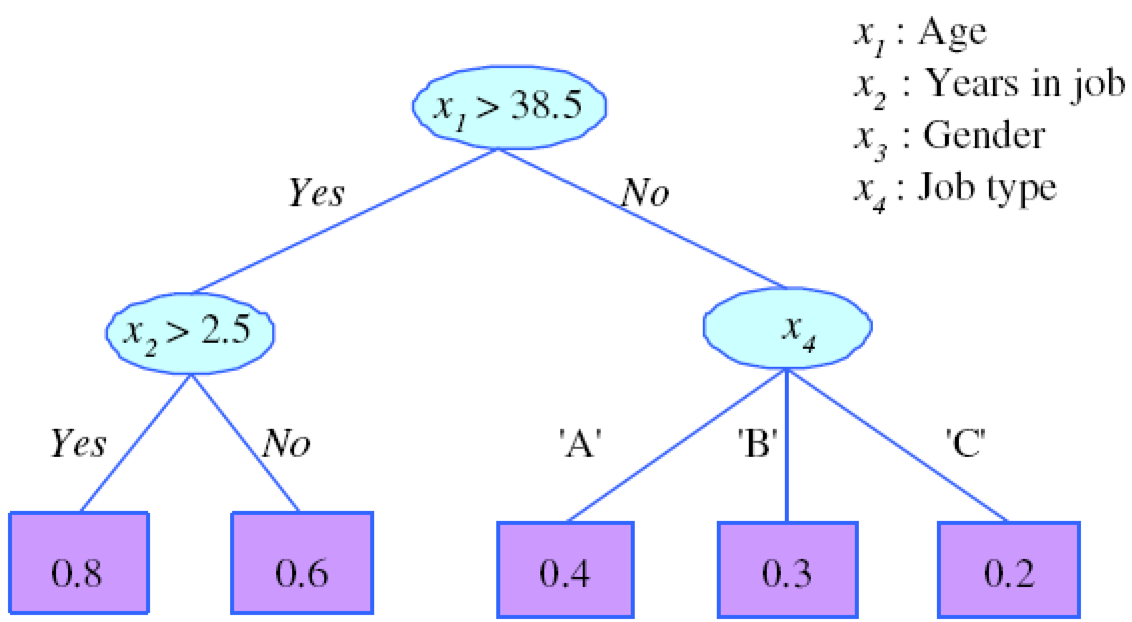
- Early stopping (prepruning) is fastest
- Pruning (postpruning) is usually more accurate





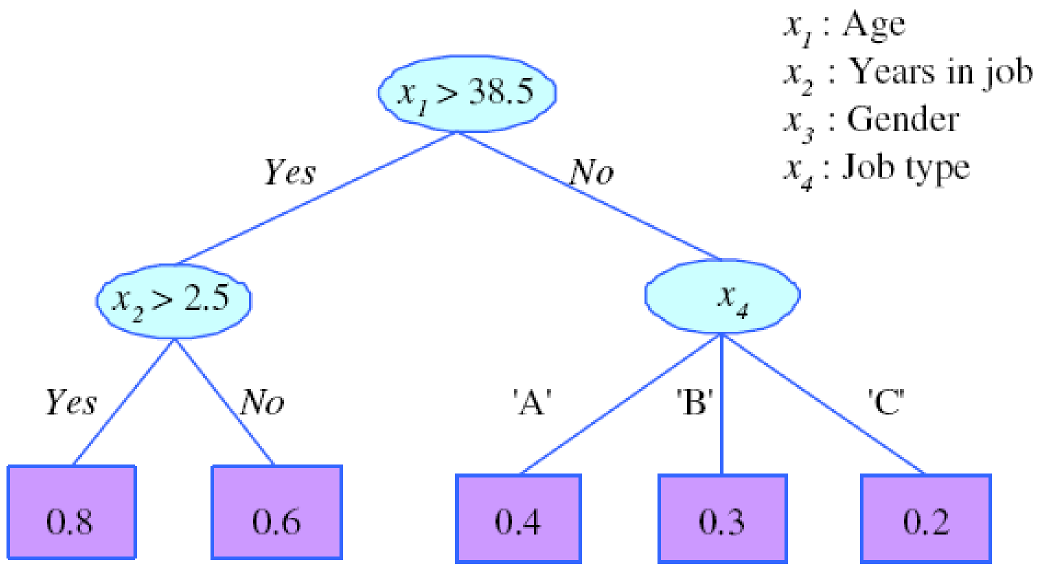
Rule extraction

- Decision trees can easily be interpreted as rules
- Feature importance:
  - Some features may not be used (feature extraction)
  - Basal features more globally important



Rule extraction

- Paths through the tree can be written as IF-THEN rules
- Together, the rules form a *rule base*



R1: IF (age > 38.5) AND (years-in-job > 2.5) THEN $y = 0.8$

R2: IF (age > 38.5) AND (years-in-job ≤ 2.5) THEN $y = 0.6$

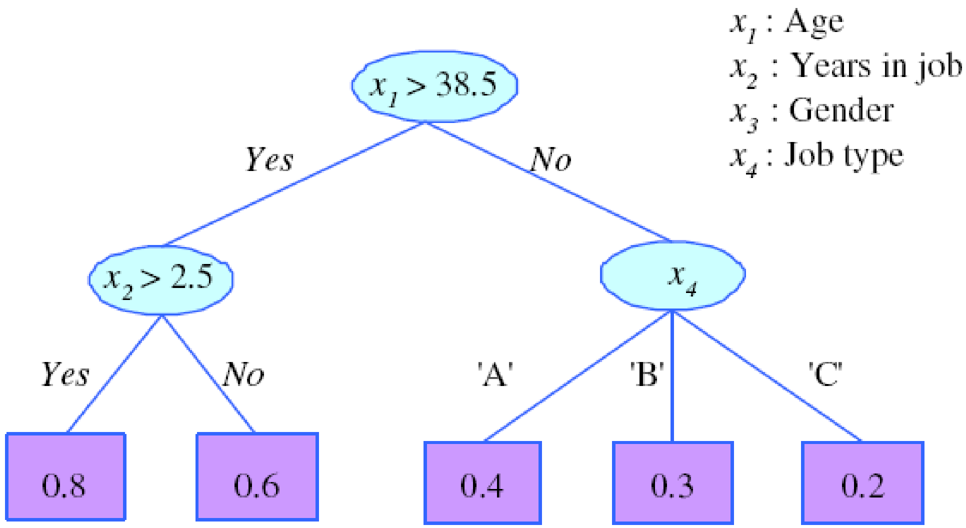
R3: IF (age ≤ 38.5) AND (job-type = 'A') THEN $y = 0.4$

R4: IF (age ≤ 38.5) AND (job-type = 'B') THEN $y = 0.3$

R5: IF (age ≤ 38.5) AND (job-type = 'C') THEN $y = 0.2$

Rule extraction

- Rules can also be pruned
- May break the underlying tree structure



R1: IF (age > 38.5) AND (years-in-job > 2.5) THEN $y = 0.8$
R2: IF (age > 38.5) AND (years-in-job $\leq$ 2.5) THEN $y = 0.6$
R3: IF (age $\leq$ 38.5) AND (job-type = 'A') THEN $y = 0.4$
R4: IF (age $\leq$ 38.5) AND (job-type = 'B') THEN $y = 0.3$
R5: IF (age $\leq$ 38.5) AND (job-type = 'C') THEN $y = 0.2$

For example: $R3'$: IF (job-type='A') THEN y =0.4

Rule Induction

- Rule induction is similar to tree induction but
  - tree induction is breadth-first,
  - rule induction is depth-first; one rule at a time
- Rule set contains rules; rules are conjunctions of terms
- Rule covers an example if all terms of the rule evaluate to true for the example
- Sequential covering: Generate rules one at a time until all positive examples are covered
- IREP (Fürnkrantz and Widmer, 1994), Ripper (Cohen, 1995)

Sources & Resources

- Duda, Hart and Stork, Pattern Classification, Second Edition, Wiley, 2001
- bradleyboehmke.github.io/HOML/DT.html
- towardsdatascience.com/entropy-how-decision-trees-make-decisions-2946b9c18c8
- scikit-learn.org/stable/modules/tree.html
- saedsayad.com