Data Mining

Predictive Modelling

Decision Trees

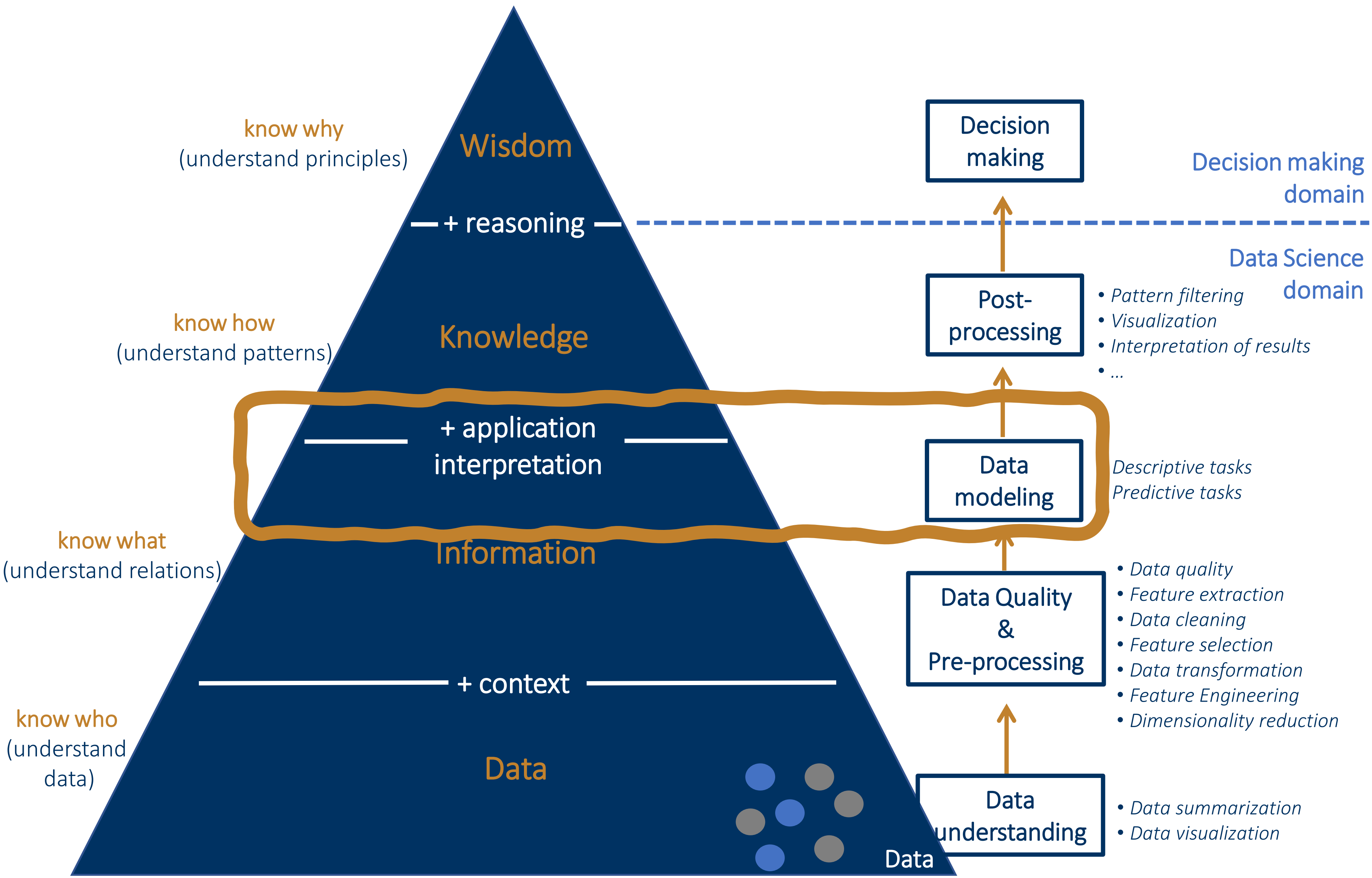
Raquel Sebastião

Departamento de Eletrónica, Telecomunicações e Informática

Universidade de Aveiro

raquel.sebastiao@ua.pt

2022/2023



know why
(understand principles)
Wisdom
+ reasoning
know how
(understand patterns)
Knowledge
+ application
interpretation
know what
(understand relations)
Information
+ context
know who
(understand data)
Data
Data
Decision making
Decision making domain
Data Science domain
Post-processing
• Pattern filtering
• Visualization
• Interpretation of results
• ...
Data modeling
Descriptive tasks
Predictive tasks
Data Quality & Pre-processing
• Data quality
• Feature extraction
• Data cleaning
• Feature selection
• Data transformation
• Feature Engineering
• Dimensionality reduction
Data understanding
• Data summarization
• Data visualization

Prediction Models – approaches

Geometric approaches

- **Distance-based:** kNN
- **Linear models:** Fisher's linear discriminant, perceptron, logistic regression, SVM (w. linear kernel)

Probabilistic approaches

- naive Bayes, logistic regression

Logical approaches

- classification or regression trees, rules

Optimization approaches

- neural networks, SVM

Sets of models (ensembles)

- random forests, adaBoost

Decision Trees

Model to predict a response y based on the feature vector

$$X = \left[x^1 \quad x^2 \quad \ldots x^D\right]^T \in \mathbb{R}^D$$

If the dependent (target) variable y (**predicted**) represents:

- Class label: **classification tree**
- Real number: **regression tree**

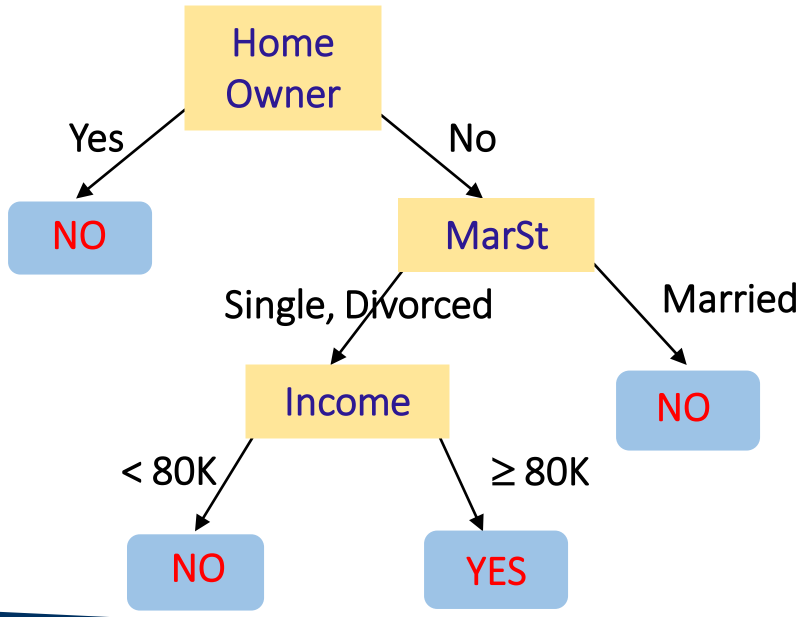
Ensemble algorithms

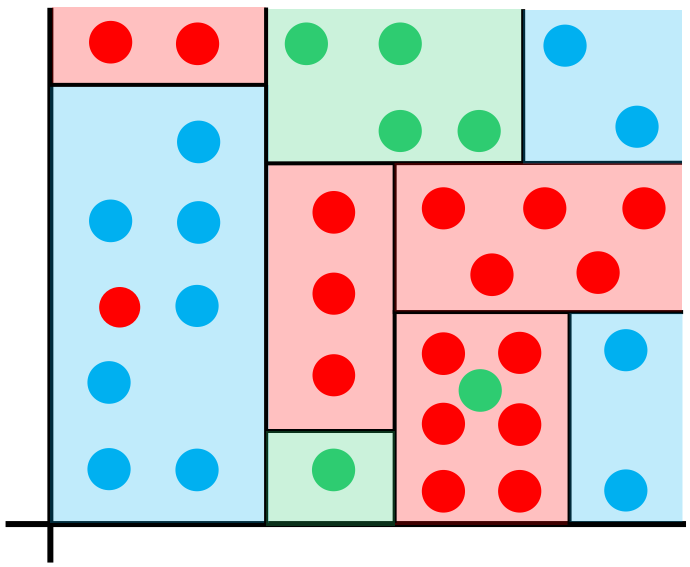
- combining many decision trees

Decision Trees

Divide-and-conquer method

- Set of ***splitting rules***
- **Partitioning** the feature space into smaller (non-overlapping) **regions with similar response values**
  - Each node is a partition of the input space

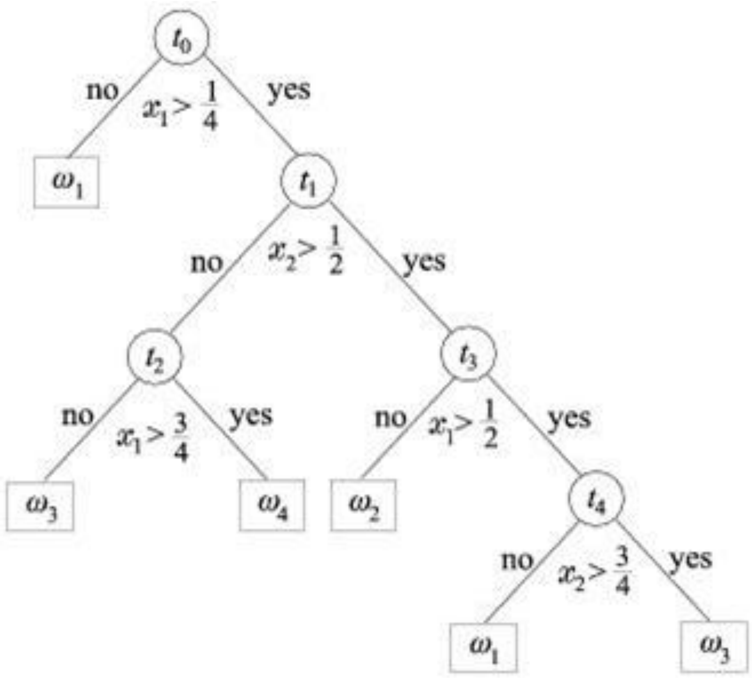


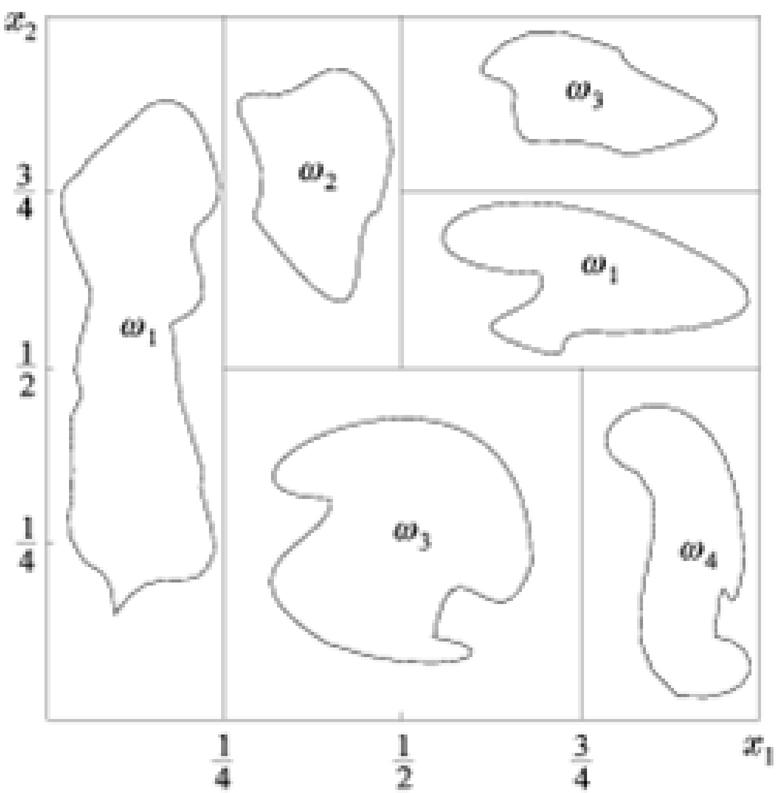


Decision Trees

The decision surfaces

- hyperplanes, splitting the space into regions
- are parallel to the axis of the spaces





Decision Trees

Tree:

- **Root node** – no income edges
- **Internal node** – with a test on an attribute/predictor variable
- **Branch** – represents an outcome of the test in the respective node
- **Leaf node** – contains the value of the target variable (either class label or a numeric value)

Training phase: at each node, one attribute is chosen to split training examples into distinct classes as much as possible

Testing phase: A new case is classified by following a matching path to a leaf node

Decision Trees

Tree:

- **Root node** – no income edges
- **Internal node** – with a test on an attribute/predictor variable
- **Branch** – represents an outcome of the test in the respective node
- **Leaf node** – contains the value of the target variable (either class label or a numeric value)

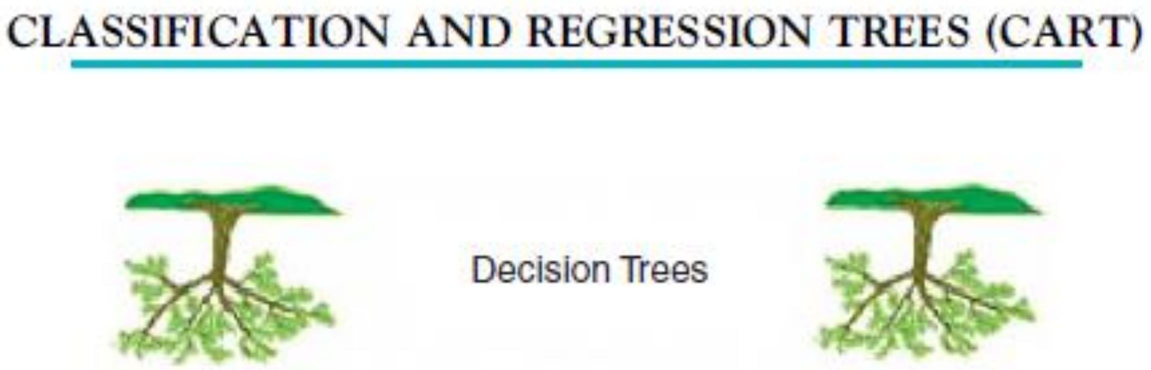
Training phase: at each node, one attribute is chosen to split training examples into distinct classes as much as possible

Testing phase: A new case is classified by following a matching path to a leaf node

Decision Trees

Algorithms:

- CLS (Hunt et al., 1966): Concept Learning System – early algorithm for DT construction
- ID3 (Quinlan, 1979): based on information theory
- **CART** (Breiman et al., 1984)
- C4.5 (Quinlan, 1993): improved extension of ID3
- (...)



Decision Trees: learning

Algorithms to grow a decision tree use a best feature x_i at each node and the related value

- to split the corresponding subset of the training set into more homogeneous groups

The main issues of the algorithm

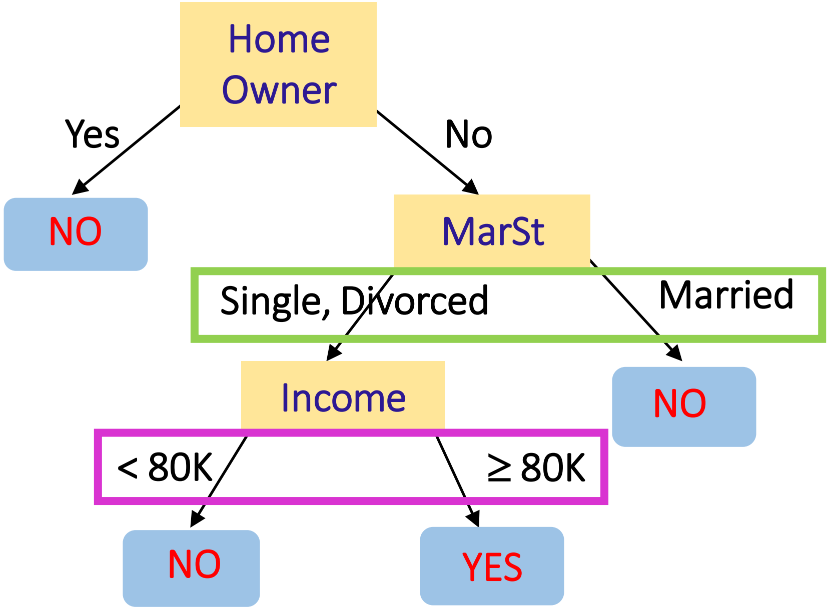
- Splitting Rule: split the cases of the sample to form partitions that are more homogeneous ("purer") than the parent node
  - Using a *homogeneity function* (impurity function) based on the distribution of the classes at each node
- Class Assignment Rule: Assign a class to each leaf (terminal node)
- Pre-pruning Rule: early stop (stop growing the three)

Post-Pruning: removing branches and nodes after training

Decision Trees: learning

Splitting binary rules

- numerical predictors: $x_i \in \mathbb{R}$
- categorical predictors: $x_i \in \{v_1, ..., v_m\}$



Each path from the root till a leaf node is a set of logical tests defining a region on the predictors space

All the examples “falling” on a leaf will get the same prediction (either a class label or a numeric value)

Decision Trees: learning

Splitting binary rules

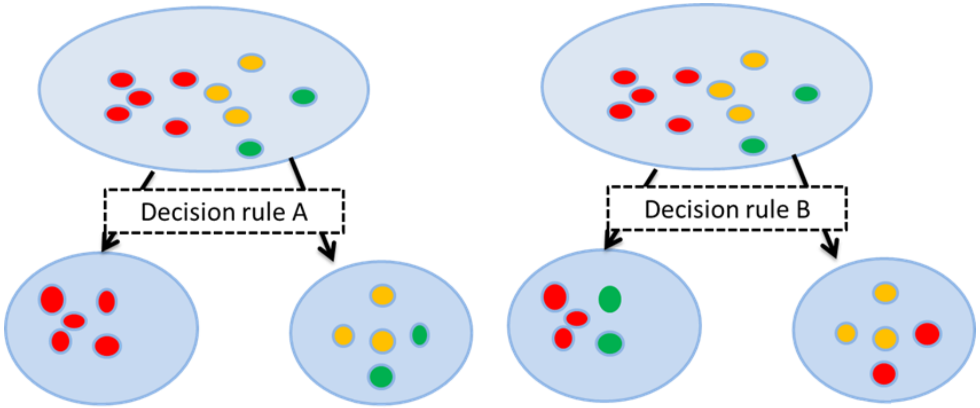
- **How to handle categorical attributes***
  - Evaluate all possible combinations of 2 subsets of values
  - Choose the test that produces more homogeneous partitions

- **How to handle continuous attributes**
  - Discretize continuous values and treat them as categorical values
  - Determine the best split point
    - Sort all values in increasing order
    - Consider all possible binary splits and finds the best cut

* **For ordinal attributes**: preserve order property among attribute values

Decision Trees: homogeneity function

The basic idea is to find out decision rules which turn the subsets more homogenous



- Decision rule A: one of the subsets is homogeneous
- Decision rule B: subsets have samples of two classes almost equiprobable

The rule A is preferred over rule B

Decision Trees: designing facts

Each node k is associated with a subset $\mathbf{X}_k$ of the training set $\mathbf{X}$.

In node k

- $\mathbf{X}_k$ is split into two (binary splits) disjoint descendant subsets:
  - $\mathbf{X}_{k,N} \cap \mathbf{X}_{k,M} = \emptyset$
  - $\mathbf{X}_{k,N} \cup \mathbf{X}_{k,M} = \mathbf{X}_k$
  - the subsets $\mathbf{X}_{k,N}$ and $\mathbf{X}_{k,M}$ correspond to ”YES” and ”NO” branches

How to measure class-homogeneity of set $\mathbf{X}_k$ and subsets $\mathbf{X}_{k,N}$ and $\mathbf{X}_{k,M}$?

- Computing the variation on entropy (information gain)
- Computing the variation on Gini index

Decision Trees: homogeneity of sets

How to measure class-homogeneity of (sub)set $\mathbf{X}_k$ and subsets $\mathbf{X}_{k,N}$ and $\mathbf{X}_{k,M}$?

- Computing the variation on entropy (information gain)

$$\Delta I(k) = I(k) - \frac{N_{k,N}}{N_k} I(k_N) - \frac{N_{k,M}}{N_k} I(k_M)$$

- Computing the variation on Gini index

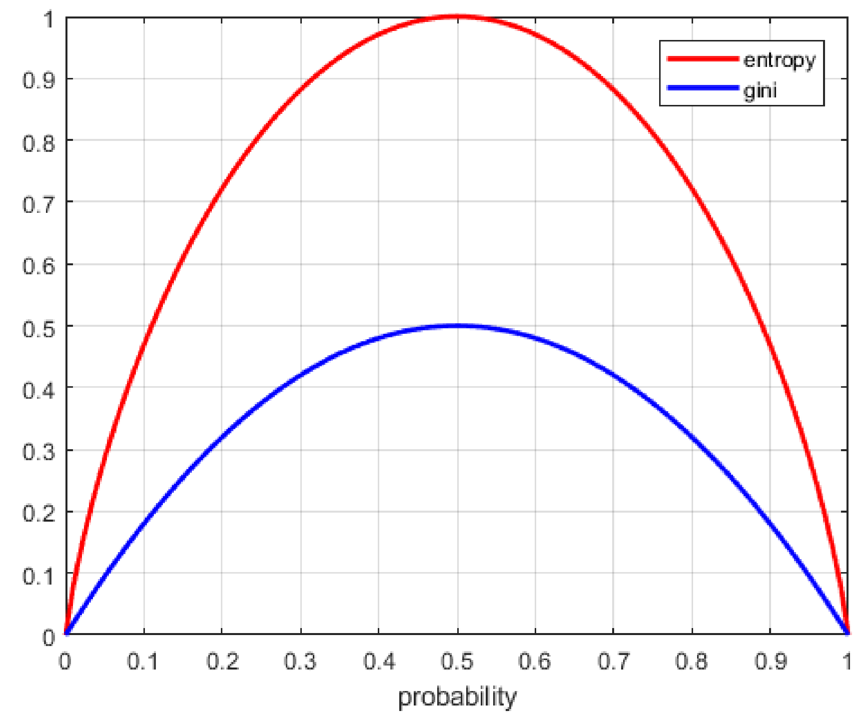
$$\Delta G(k) = G(k) - \frac{N_{k,N}}{N_k} G(k_N) - \frac{N_{k,M}}{N_k} G(k_M)$$

  - Entropy: $I = -\sum_i p_i \, log_2(p_i)$
  - Gini index: $G = 1 - \sum_i (p_i)^2$

  where p_i , $i = 1, \ldots, C$ are the probabilities of the classes

Decision Trees: homogeneity of sets

Data set with two (C=2) classes



- *I* (or *G*) is **maximum** when $p_i = p_j, \forall i, j$
  - maximum uncertainty (randomness, **non-homoegeous**)
- *I* (or *G*) is **minimum** when $p_i = 1, p_j = 0 \quad (i \neq j\ )$ *i* = 1
  - the outcome is already known (set is **homogeneous**)

Decision Trees: splitting criteria

Node k: with N_k elements, belonging to classes ω_i , $i = 1, \dots, C$

- Entropy* of node k

$$I(k) = -\sum_{i=1}^{C} P(\omega_i|k) \ \ log_2\left(P(\omega_i|k)\right)$$

where

- $P(\omega_i|k) = \frac{N_k^{(i)}}{N_k}$
- $N_k^{(i)}$ is the number of examples (in node k) that belongs to class ω_i

* The **Gini index** can be used instead of **entropy**

Decision Trees: splitting criteria

Assuming that the (sub)set $\mathbf{X}_k$ (at node k) is divided into the subsets

- $\mathbf{X}_{k,N}$ with entropy $I(k_N)$
- $\mathbf{X}_{k,M}$ with entropy $I(k_M)$

Then the decrease in impurity with this split is given by the variation on entropy

$$\Delta I(k) = I(k) - \frac{N_{k,N}}{N_k} I(k_N) - \frac{N_{k,M}}{N_k} I(k_M)$$

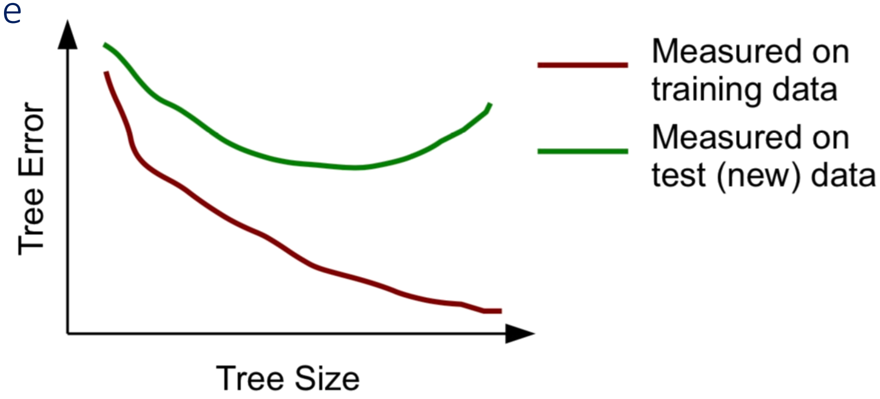
$\Delta I(k)$* is computed for every possible splitting test

The **splitting rule** is chosen for the maximum $\Delta I(k)$

* The **Gini index** can be used instead of **entropy**

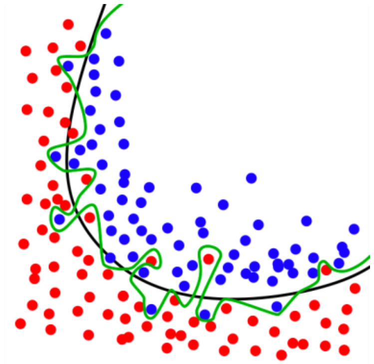
Decision Trees: overfitting and tree pruning

- Overall scores keep improving while growing the DT
- Going down in the tree: the split decisions are made based on smaller and smaller sets
  - Thus, potentially less reliable decisions are made



Overfitting

Too large trees tend to overfit the training data and will perform badly on new data



Decision Trees: overfitting and tree pruning

Pre-pruning: stop growing the tree if our estimate of quality indicates that is not worth continuing

- minimum nr. of examples in a node
- minimum nr. of examples in a leaf
- maximum depth of the tree

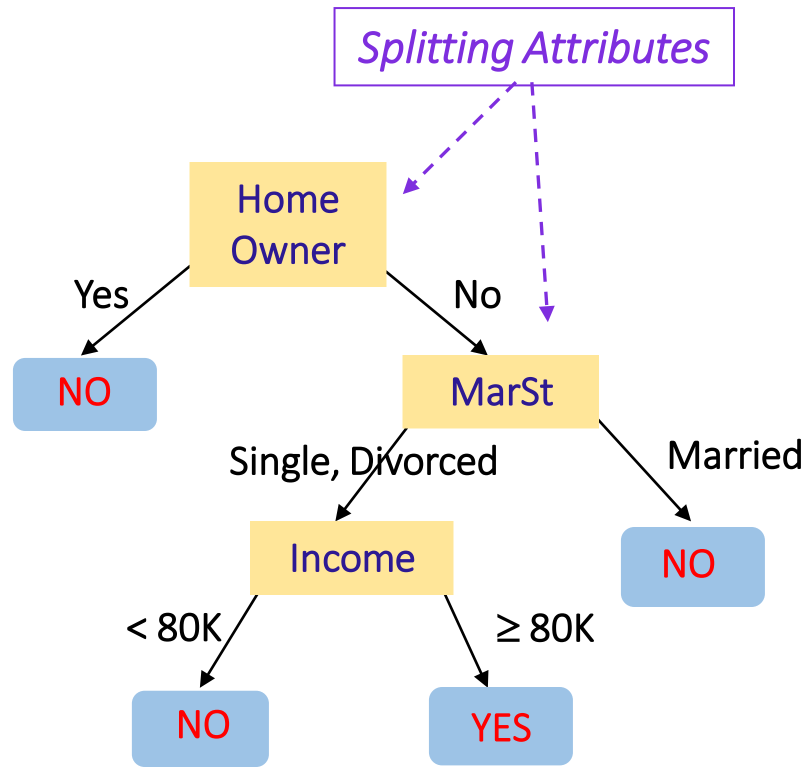
Post-prunning: grow an overly large tree and then use some statistical procedure to prune unreliable branches according to error estimates

- *Minimum error*: The tree is pruned back to the point where the cross-validated error is a minimum
- *Smallest tree*: The tree is pruned back slightly further than the minimum error

Classification Trees: example

ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
	categorical	categorical	continuous	target
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

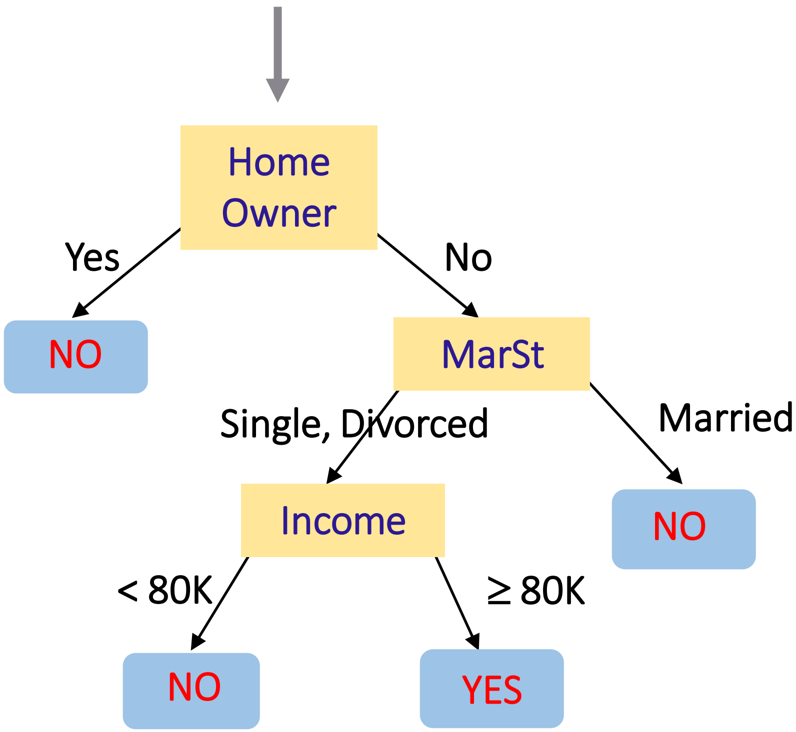
Training Data



Model: Decision Tree

Classification Trees: apply model to test data

Start from the root of tree

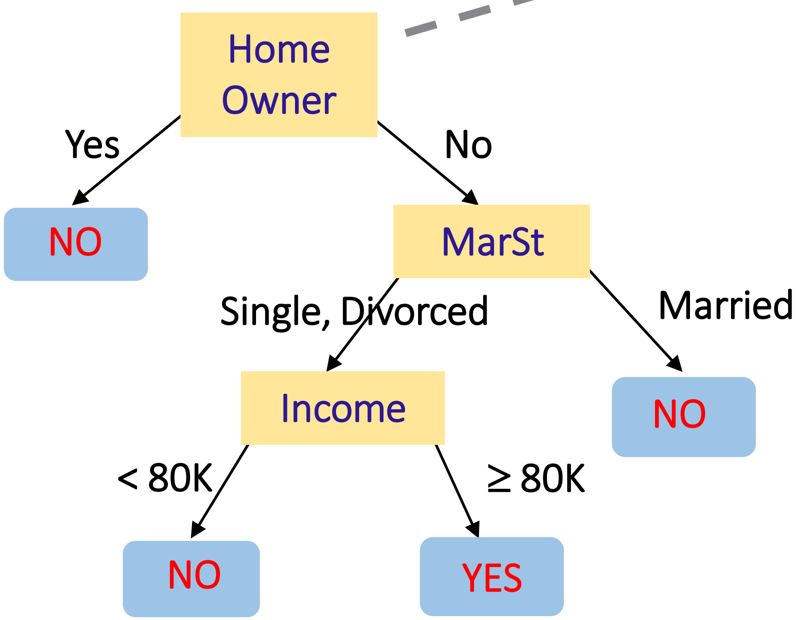


Test Data

Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?

The prediction for a new test case is obtained by following a path from the root till a leaf according to its predictor values

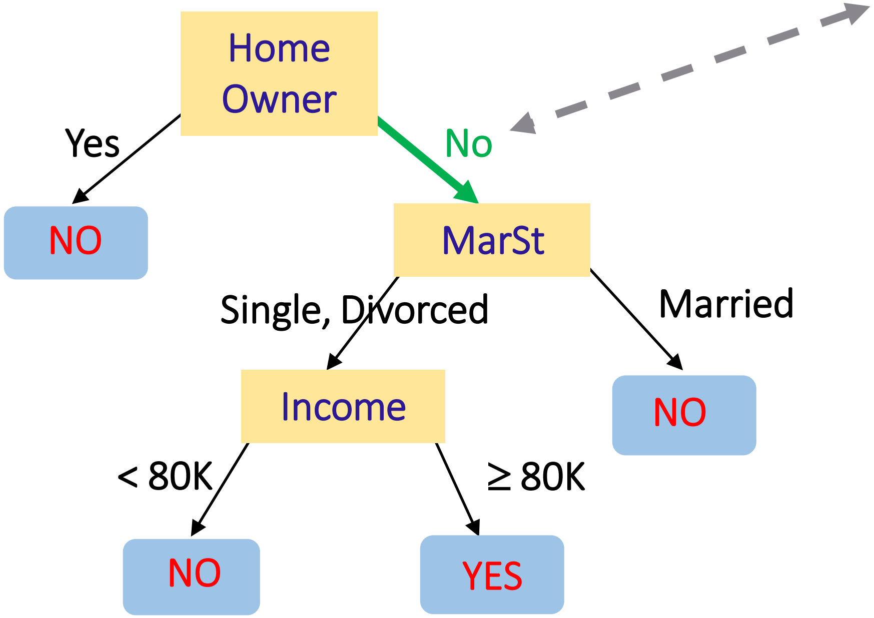
Classification Trees: apply model to test data



Test Data

Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?

Classification Trees: apply model to test data



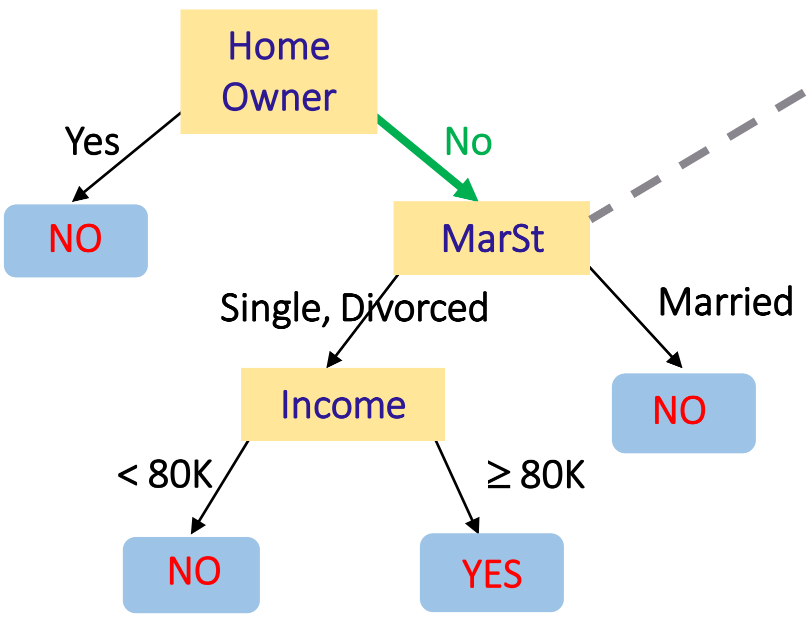
Test Data

Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?

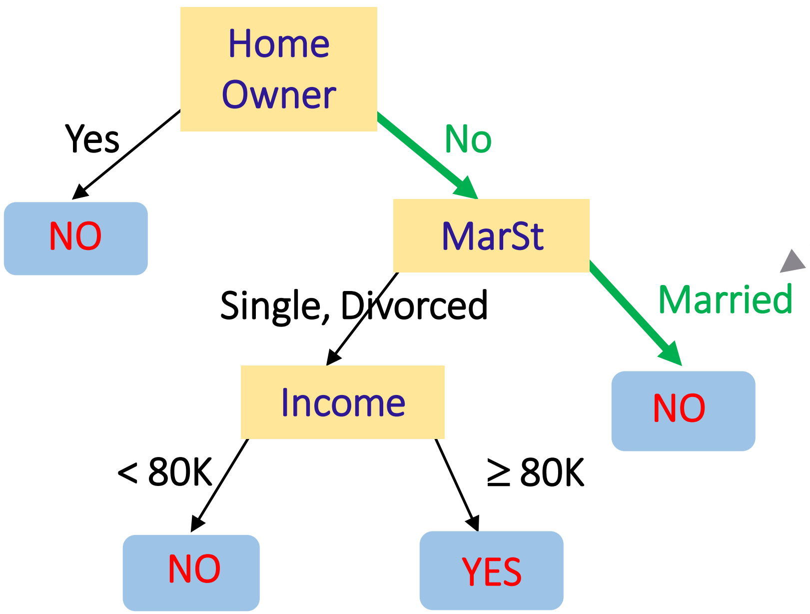
Classification Trees: apply model to test data

Test Data

Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?



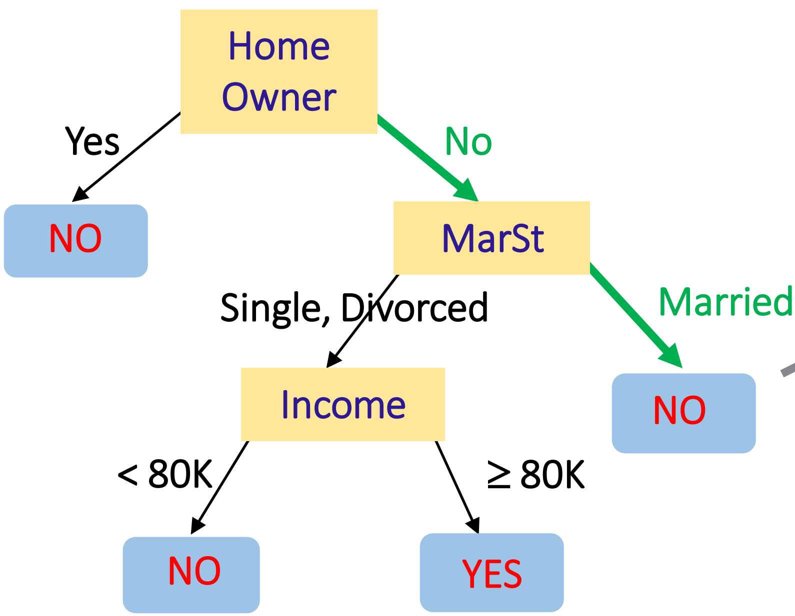
Classification Trees: apply model to test data



Test Data

Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?

Classification Trees: apply model to test data



Test Data

Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?

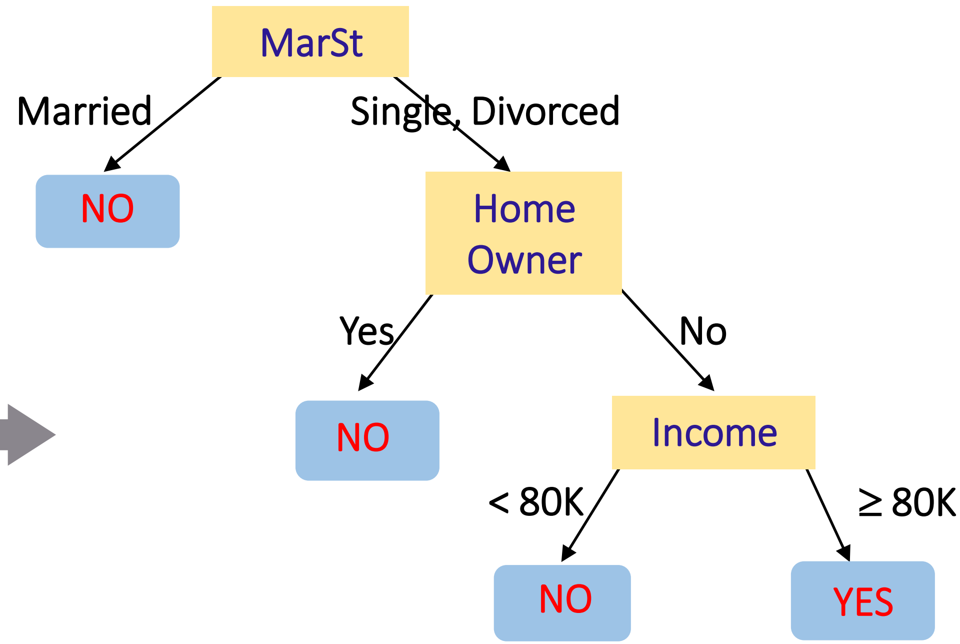
Assign target *Defaulted Borrower* to “No”

Classification Trees: same data, another DT

categorical categorical continuous target

ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Training Data



There could be more than one tree that fits the same data!

Decision Trees: advantages and disadvantages

Advantages

- Easy to interpret models
- Can cope with any data type (categorical and numeric)
- Don’t require feature scaling
- Embedded feature importance and handling of missing values
- Relatively inexpensive to construct
- Extremely fast at classifying unknown examples

Disadvantages

- Not robust: can be very sensitive to small variations in the data -> **High variance**
- **Very complex decision trees** usually have low bias (difficult to incorporate new data):
  - Weak learners: poor classification performance (specially if overfitting occurs)
- Key features might be hidden by more dominant ones: interacting attributes (that can distinguish between classes together but not individually) may be passed over in favor of other attributed that are less discriminating.

Bibliography

Introduction to Data Mining, Pang-Ning Tan, Michael Steinbach, Anuj Karpatne, Vipin Kumar, *Pearson*, 2019 (chap 3.3, 3.4)

Data Mining, the Textbook, Charu C. Aggarwal, *Springer*, 2015 (chap 10.2, 10.3)

Xindong Wu, et al., **Top 10 algorithms in data mining**, Knowl Inf Syst, 14(1):1-37, 2007. https://doi.org/10.1007/s10115-007-0114-2