# Machine Learning for Bioinformatics: Tree-based methods.

**S 1133 — Fall 2016**

**Chloé-Agathe Azencott**
Centre for Computational Biology, Mines ParisTech
chloe-agathe.azencott@mines-paristech.fr

Inserm
Institut national
de la santé et de la recherche médicale

institutCurie

MINES
ParisTech

# Decision trees

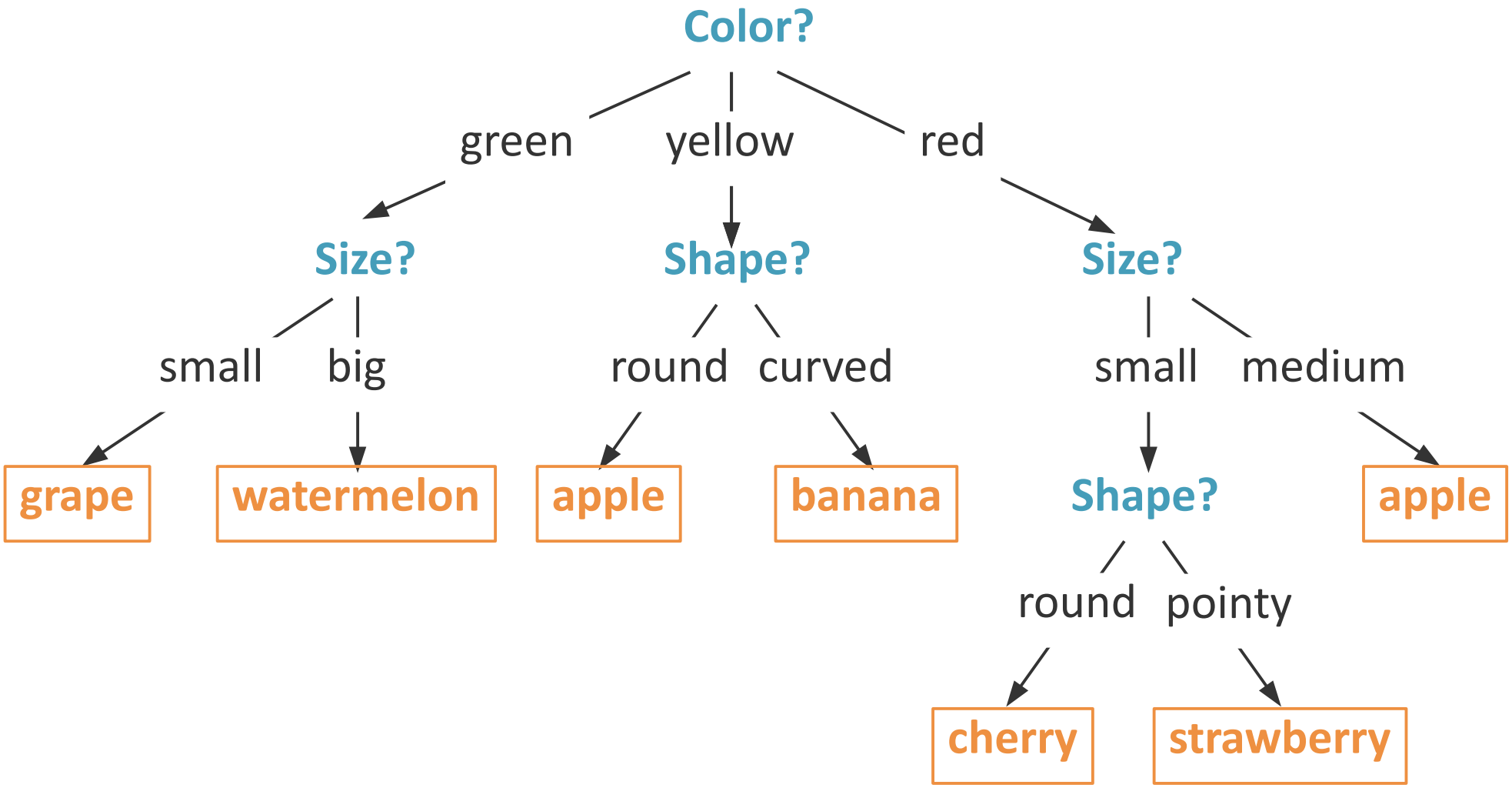
# Nominal data

- Attributes that are
  - **Discrete**
  - Without any natural notion of **similarity/ordering**

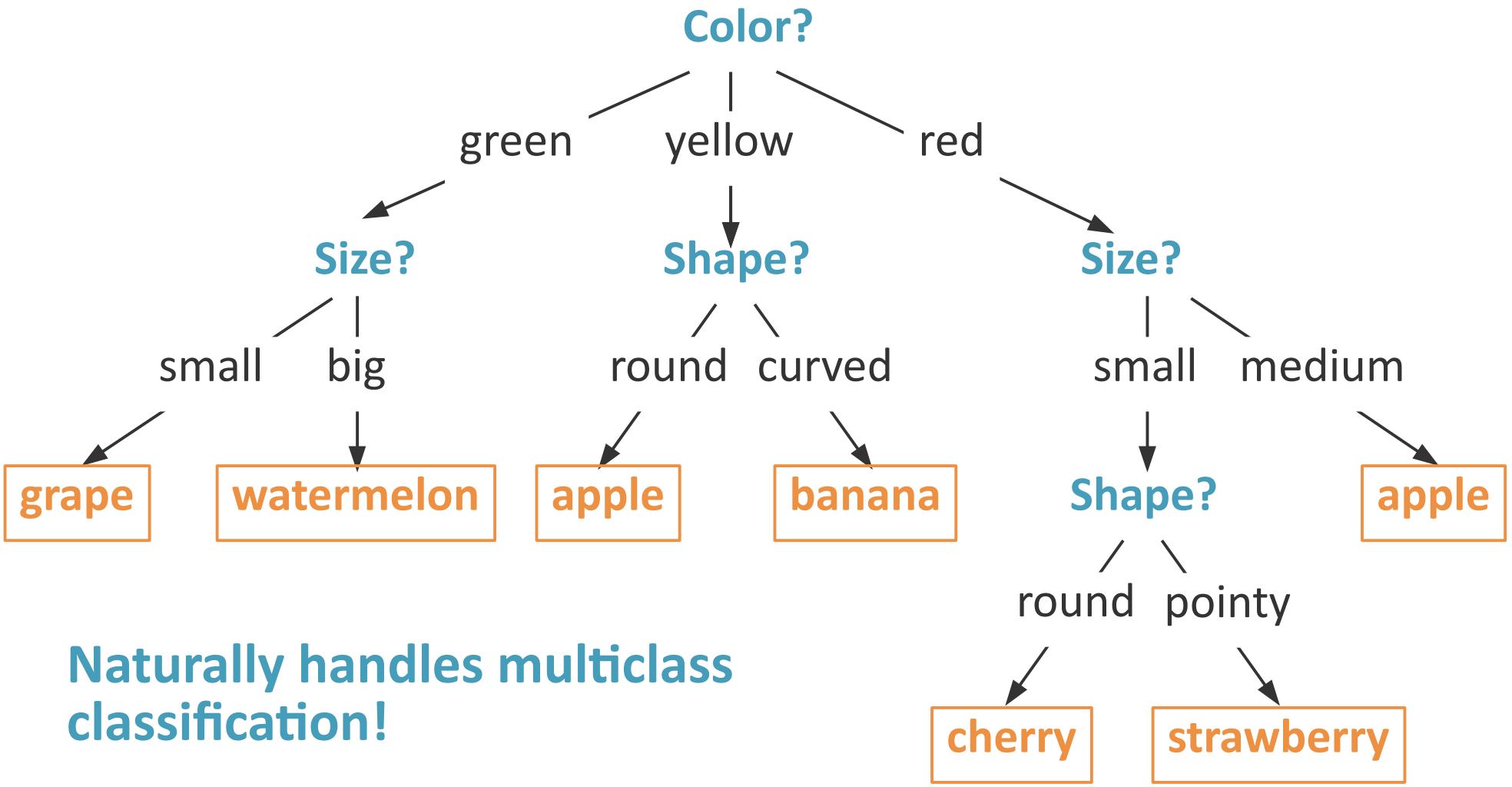
    → **Non-metric learning.**
- Example:

  Classify fruit from {color, shape, texture, size}.

# Decision trees: The 20Q game

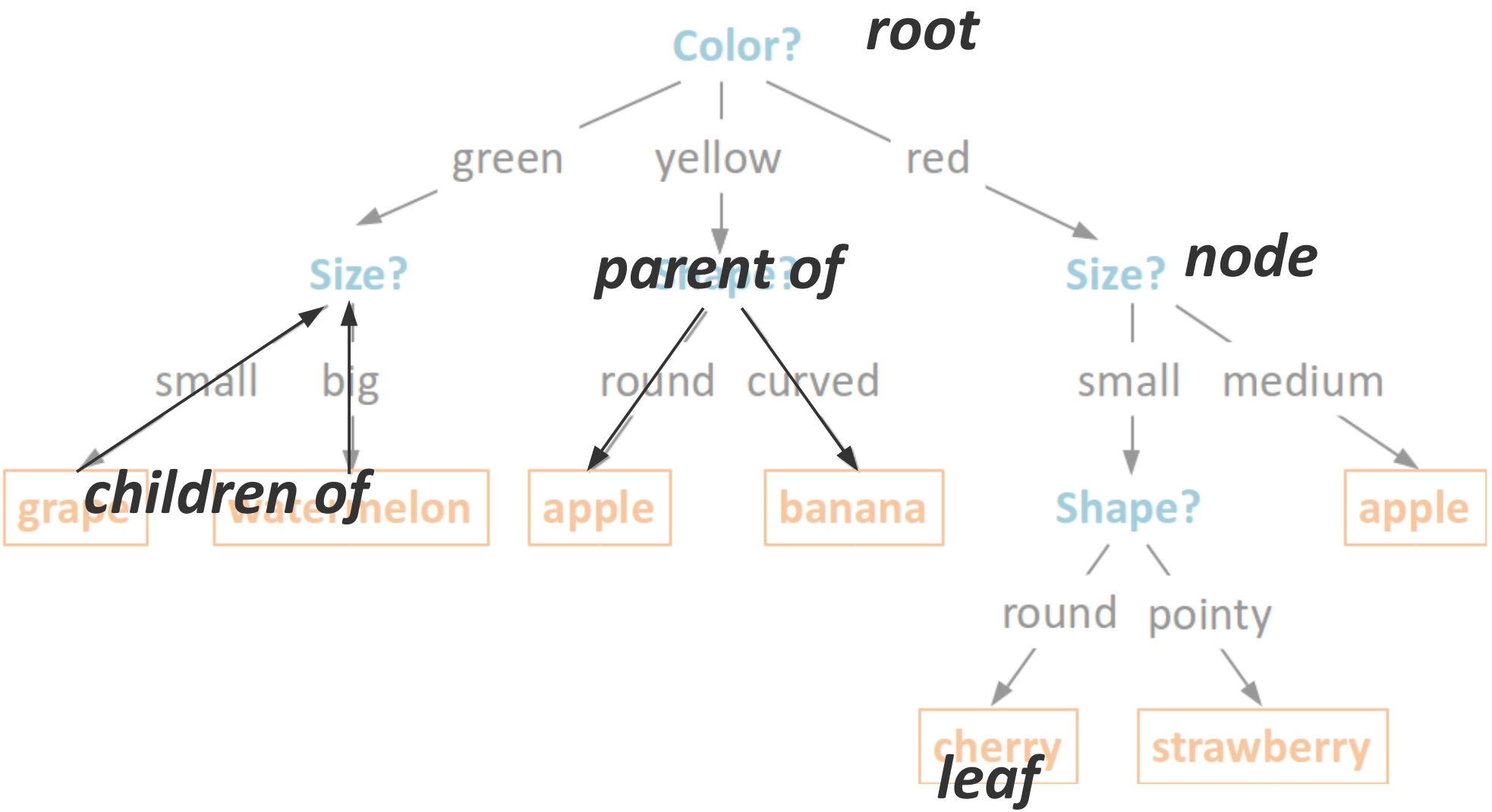


# Decision trees: The 20Q game




**Naturally handles multiclass classification!**

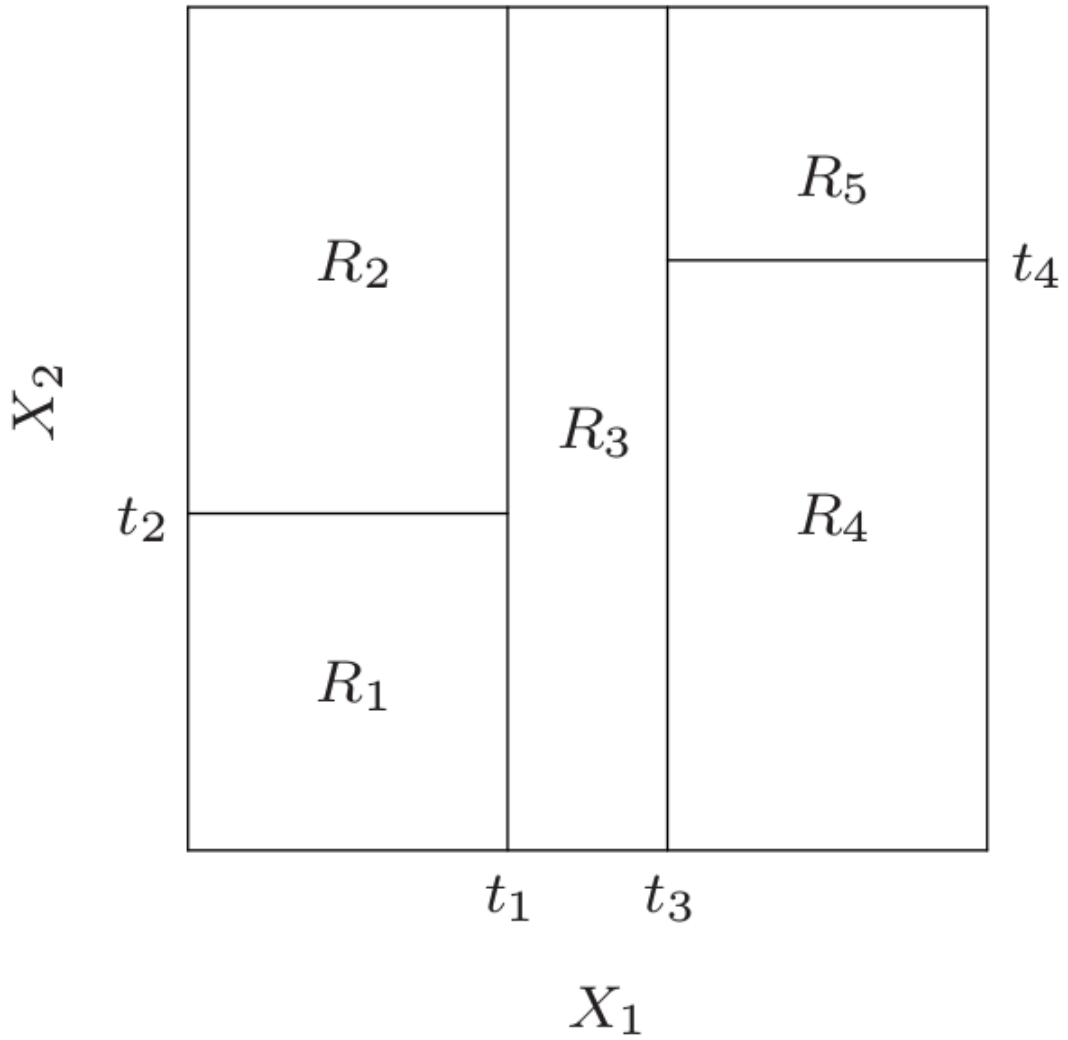
# Decision trees: The 20Q game



# Partition of the feature space

$$f(x) = \sum_{m=1}^{M} c_m I(x \in R_m)$$

indicator




- **Classification?**

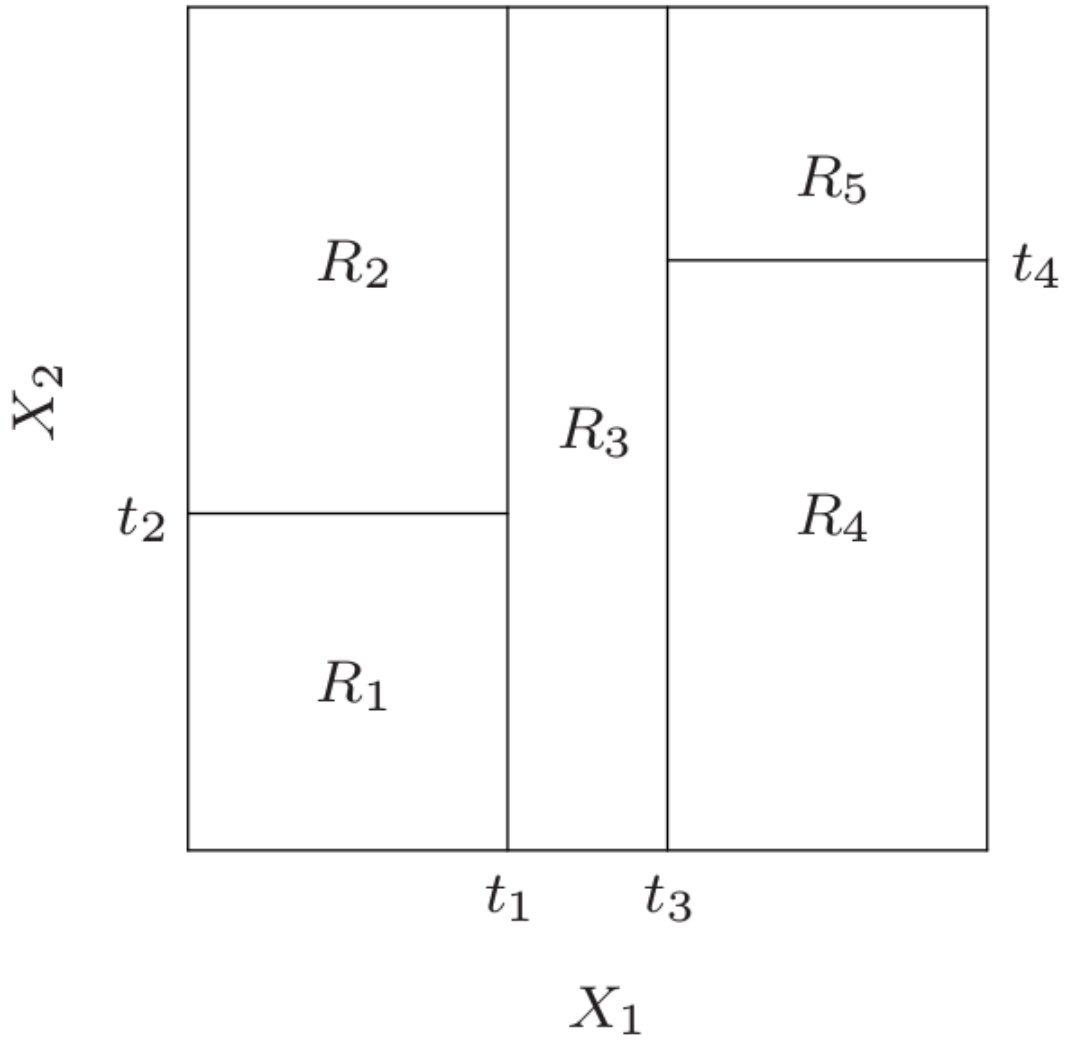
  $c_m$ =

- **Regression?**

  $c_m$ =

# Partition of the feature space

$$f(x) = \sum_{m=1}^{M} c_m I(x \in R_m)$$




- **Classification:**

  $c_m$ = majority vote in region.

- **Regression:**

  $c_m$ = average value in region.

# CART: Design choices

- **CART: Classification And Regression Trees**

  Generic name for a recursive procedure to split a training set and organize it into a tree.

- **Binary** or **multi-way** splits?

  A tree with arbitrary branching factor can always be transformed into a binary tree

- Which **feature** to use at each node?

  How to grow a tree?

- When to **stop** growing a tree?

# How to grow a tree

# How to grow a tree

- **Splitting variable** (j) and **splitting point** (s) define 2 regions:

$$R_l = \{x : x_j \leq s\} \text{ and } R_r = \{x : x_j > s\}$$

# How to grow a tree

- **Splitting variable** (j) and **splitting point** (s) define 2 regions:

$$R_l = \{x : x_j \leq s\} \text{ and } R_r = \{x : x_j > s\}$$

- **Regression tree:** choose j and s to **minimize SE:**

$$\min_{j,s} \left( \sum_{i:x_i \in R_l(j,s)} (y_i - c_l)^2 + \sum_{i:x_i \in R_r(j,s)} (y_i - c_r)^2 \right)$$

# How to grow a tree

- **Splitting variable** (j) and **splitting point** (s) define 2 regions:

$$R_l = \{x : x_j \leq s\} \text{ and } R_r = \{x : x_j > s\}$$

- **Classification tree:** choose j and s to **minimize impurity.**

$$\min_{j,s} \left( \frac{|R_l(j,s)|}{n_{tot}} \times \text{Imp}(R_l(j,s)) + \frac{|R_r(j,s)|}{n_{tot}} \times \text{Imp}(R_r(j,s)) \right)$$

- Greedy algorithm / local optimization.

# How to grow a tree

- **Splitting variable** (j) and **splitting point** (s) define 2 regions:

$$R_l = \{x : x_j \leq s\} \text{ and } R_r = \{x : x_j > s\}$$

- **Classification tree:** choose j and s to **minimize impurity.**
- **Impurity:**
  - Classification error
  - Entropy
  - Gini impurity.

# Impurity: Classification error

- Minimum probability that a training point will be misclassified at node (s,j)

$$\mathrm{Imp}(R_m) = 1 - \max_k \hat{p}_{mk}$$

proportion of training instances from class k in Rm

- If all examples from one class belong to Rm, then **Imp(Rm) = ?**
- If we have 2 balanced classes, and instances are randomly split at (s, j), then **Imp(Rm) = ?**

# Impurity: Classification error

- Minimum probability that a training point will be misclassified at node (s,j)

$$\mathrm{Imp}(R_m) = 1 - \max_k \hat{p}_{mk}$$

proportion of training instances from class k in Rm

- If all examples from one class belong to Rm, then **Imp(Rm) = 0**
- If we have 2 balanced classes, and instances are randomly split at (s, j), then **Imp(Rm) = 0.5**

# Impurity: Entropy

- Information theory: Shannon's entropy

proportion of training instances from class k in Rm

$$\mathrm{Imp}(R_m) = -\sum_k \hat{p}_{mk} \log_2 \hat{p}_{mk}$$

- If all examples from one class belong to Rm, then **Imp(Rm) = ?**
- If we have 2 balanced classes, and instances are randomly split at (s, j), then **Imp(Rm) = ?**

# Impurity: Entropy

- Information theory: Shannon's entropy

proportion of training instances from class k in Rm

$$\mathrm{Imp}(R_m) = -\sum_k \hat{p}_{mk} \log_2 \hat{p}_{mk}$$

- If all examples from one class belong to Rm, then **Imp(Rm) = 0**
- If we have 2 balanced classes, and instances are randomly split at (s, j), then **Imp(Rm) = 1**

# Gini impurity

$$\mathrm{Imp}(R_m) = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk})$$

proportion of training instances from class k in Rm

- If all examples from one class belong to Rm, then **Imp(Rm) = ?**
- If we have 2 balanced classes, and instances are randomly split at (s, j), then **Imp(Rm) = ?**

# Gini impurity

$$\mathrm{Imp}(R_m) = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk})$$

proportion of training instances from class k in Rm

- If all examples from one class belong to Rm, then **Imp(Rm) = 0**
- If we have 2 balanced classes, and instances are randomly split at (s, j), then **Imp(Rm) = 0.5**

# Gini impuritv

$$\mathrm{Imp}(R_m) = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk})$$

$$GI(j,s) = \frac{|R_l(j,s)|}{n_{tot}} \times \mathrm{Imp}(R_l(j,s)) + \frac{|R_r(j,s)|}{n_{tot}} \times \mathrm{Imp}(R_r(j,s))$$

- If the **split respects the overall distribution:**

$$p_{mk} = \frac{|C_k|}{N} \quad \forall k$$

- All regions are identically distributed and have Gini impurity:

$$GI(R) = \sum_{k=1}^{K} \frac{|C_k|}{N}(1 - \frac{|C_k|}{N}) = 1 - \sum_{k=1}^{K} \frac{|C_k|^2}{N^2}$$

- The Gini impurity of a K-way split is:

$$GI(j,s) = \sum_{l=1}^{K} \frac{1}{K}\mathrm{Imp}(R) = \mathrm{Imp}(R)$$

- If, in addition, |Ck|=N/K, then GI = 1 – 1/K

# Gini impurity

$$\text{Imp}(R_m) = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk})$$

$$GI(j,s) = \frac{|R_l(j,s)|}{n_{tot}} \times \text{Imp}(R_l(j,s)) + \frac{|R_r(j,s)|}{n_{tot}} \times \text{Imp}(R_r(j,s))$$

- If the **split is perfect:**
  - All regions have a proportion of 1 of one class and of 0 of the other, and hence have Gini impurity GI(R)=0.
  - The Gini impurity of a K-way split is also 0.

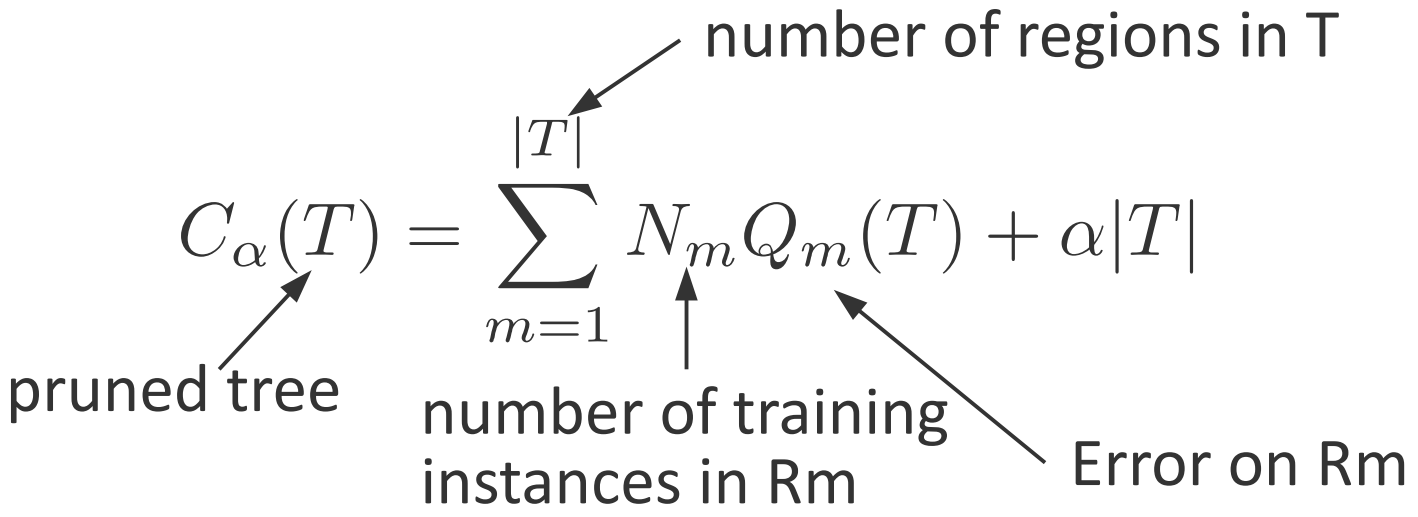
# When to stop growing a tree

# When to stop growing a tree

- **Large tree** might overfit
- **Small tree** might underfit
- Strategy:
  - grow the tree until a **minimum node size** (# training points in the region) is reached;
  - prune the tree: **cost-complexity pruning**.

# When to stop growing a tree

– prune the tree: **cost-complexity pruning**.




α: trade-off between model complexity and goodness of fit.

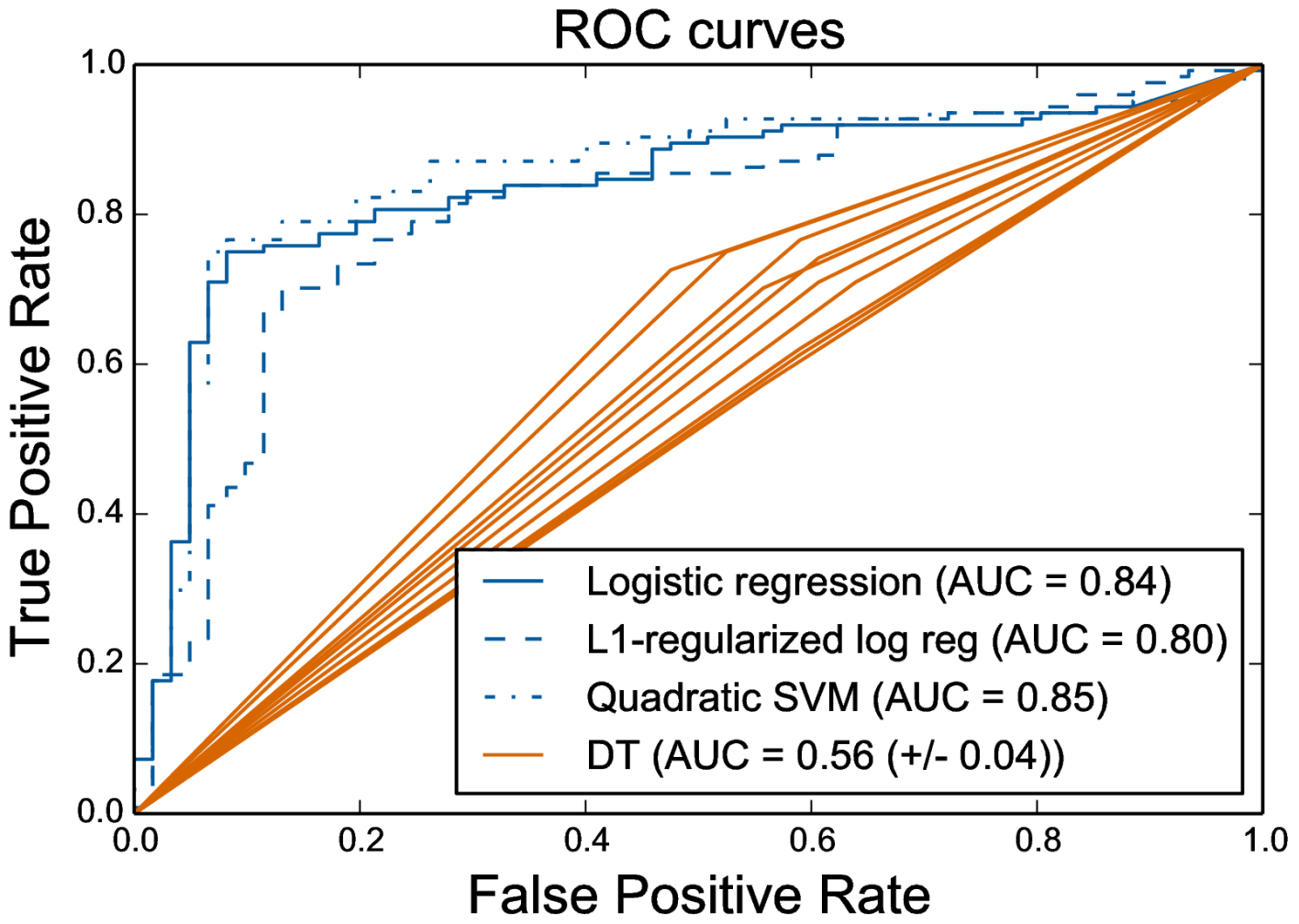
# Advantages & drawbacks of trees

- :-) Trees are **easy to explain.**
- :-) Trees seem to **mirror human-decision making.**
- :-) Trees can be **displayed graphically** and **easily interpreted.**
- :-) Trees can easily handle **quantitative variables.**
- :-) Trees naturally handle **multi-class problems.**

- :-( Trees generally do not have very good **predictive accuracy.**

# Example: Endometrium vs Uterus tumor classification

- 61 endometrium tumor samples
- 124 uterus tumor samples
- 54 675 genes
- 10-fold cross-validation



# Forests

# Building forests

- Idea: **Aggregating many weak learners can substantially increase their performance.**
- **Ensemble learning**
  - **Wisdom of crowds:** Average out the uncorrelated errors of individual classifiers.
  - Idea: Build **different decision trees** from the same data.

# Building ensembles

- **Subsample** the training data
  - **Bagging** [Breiman 1996]: bootstrap resampling
  - **Boosting** [Schapire 1990]: resample based on performance
- Use different **features**
  - Multiple input representations
  - Feature selection
- Use different **parameters** of the learning algorithm

# Combining learners

- **Non-trainable combination:**
  - **Voting** (classification)
  - **Averaging** (regression)
- **Trainable combination:**
  - **Weighted averaging:** based on performance on a validation set.
  - **Meta-learner:** the outputs of the individual learners are features for another learning algorithm.

# Bagging trees

- **Bagging:**
  - Take repeated samples from the training data (bootstrap)
  - Build one predictor from each of these samples
  - **Final prediction:** average (regression) or majority vote (classification)

# Random forests [Breiman 2001]

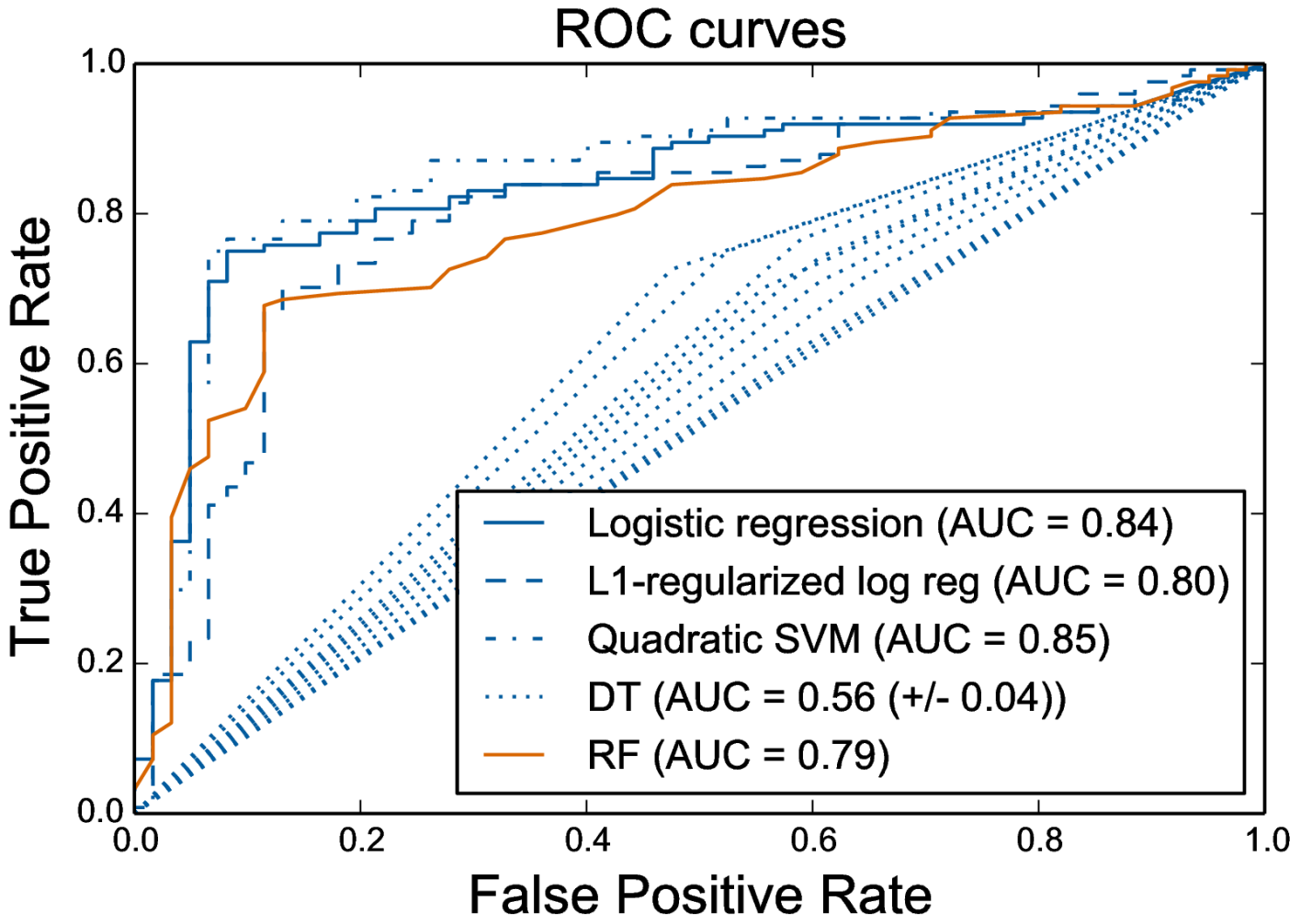
- Similar to bagging trees
- One trick to **decorrelate** the trees:

  Before splitting, first **randomly sample q** (out of p) **variables** among which the one over which to split must be chosen.

  Typically $q = \sqrt{p}$.
- Very good predictive power in practice!

# Example: Endometrium vs Uterus tumor classification

- 61 endometrium tumor samples
- 124 uterus tumor samples
- 54 675 genes
- 10-fold cross-validation



# Summary

- Decision trees are **easy to interpret.**
- Decision trees elegantly deal with
  - **Quantitative variables**
  - **Multiple classes**
  - **Multimodal distributions.**
- Decision trees have **limited predictive power,** but this can be addressed thanks to **ensemble methods**
  - **Bagging**
  - **Random forests.**