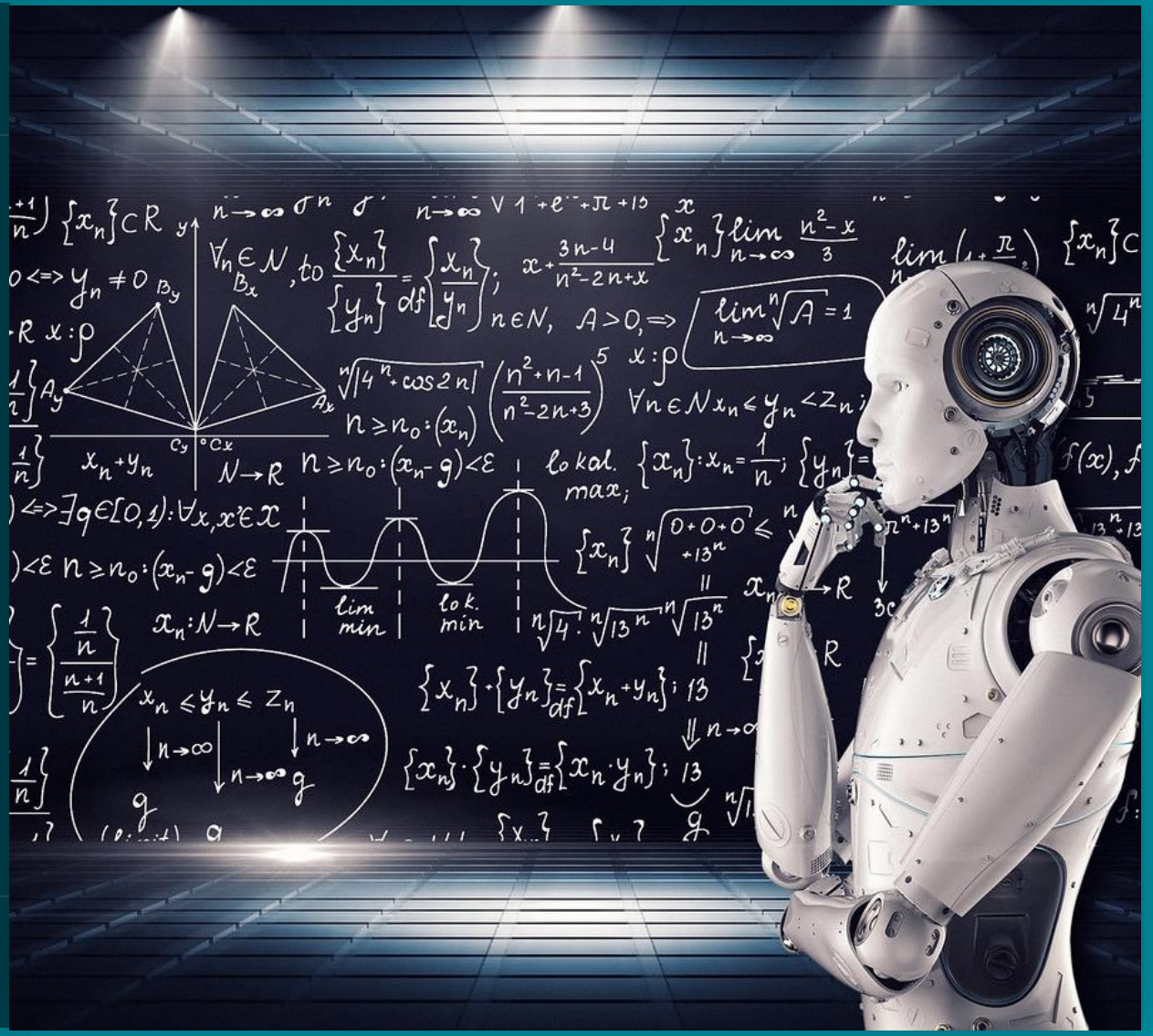
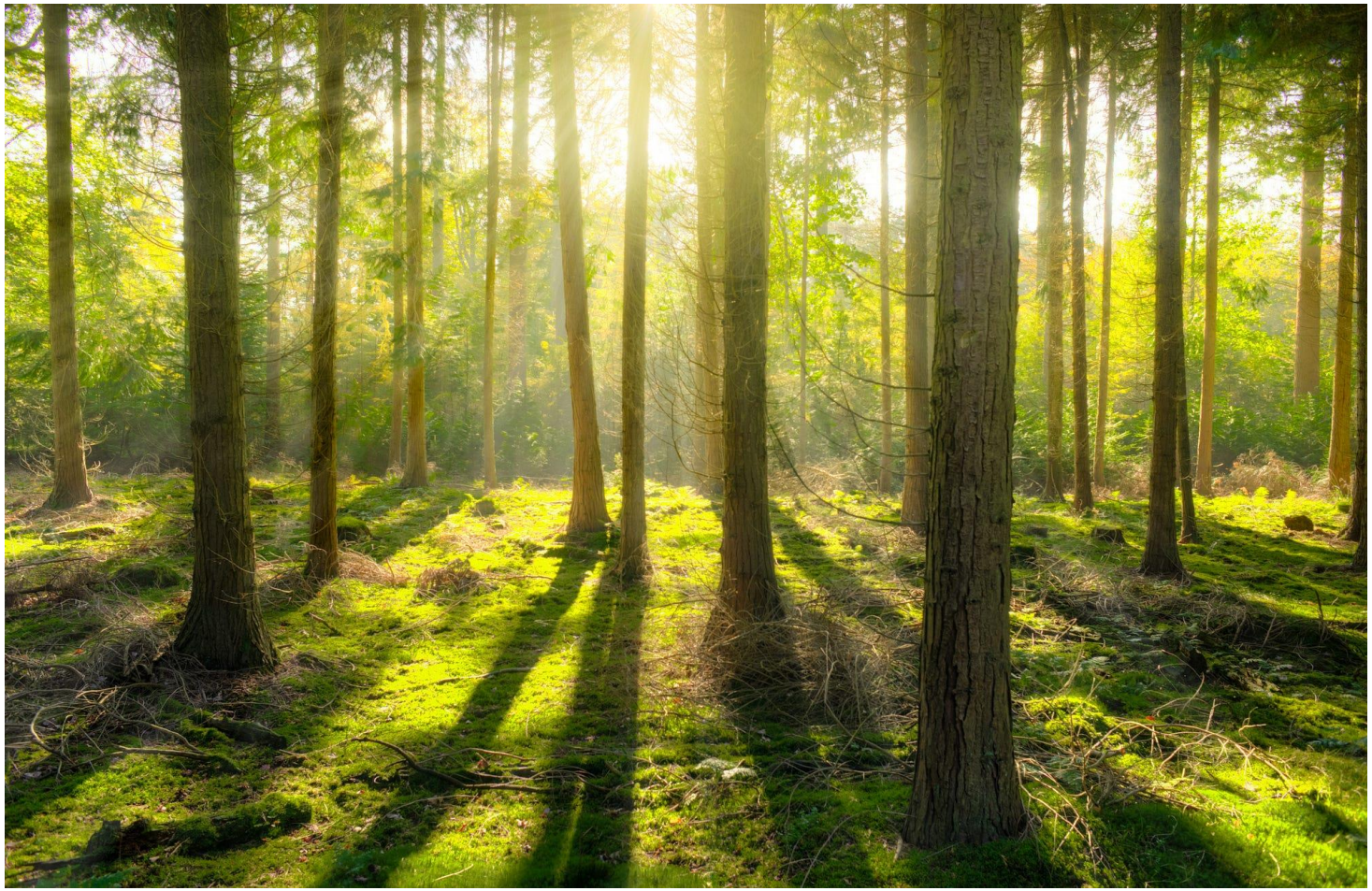
Introduction to Machine Learning

Tree-based methods

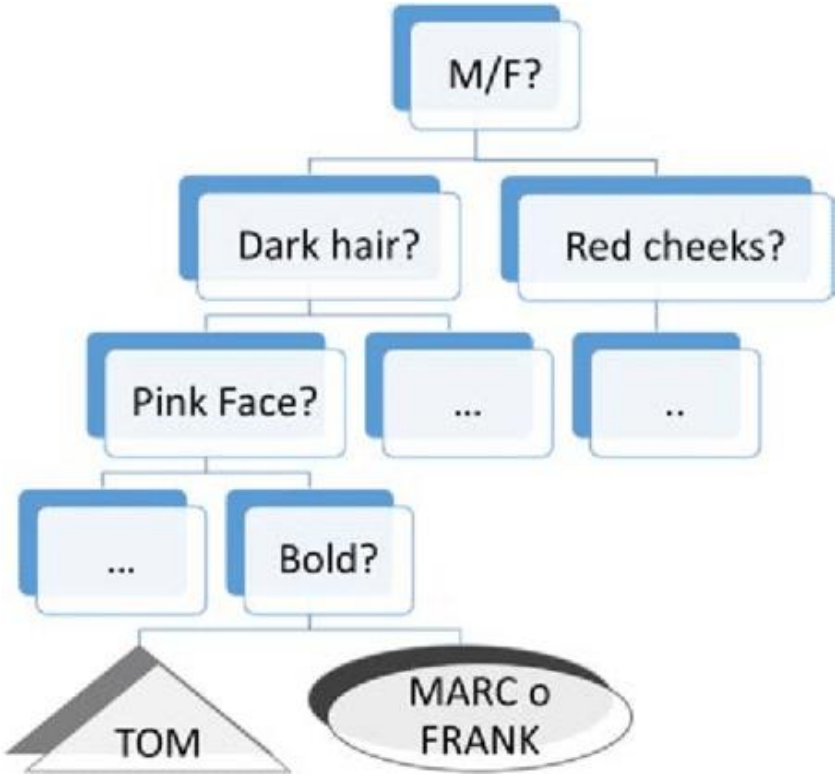


HOSPITAL DA LUZ LEARNING HEALTH
TRAINING, RESEARCH & INNOVATION CENTER

Talking about trees



Decision trees



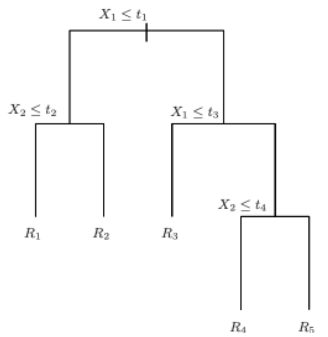
How do we find the number of predictor spaces?

- Goal: to minimize RSS or MSE (in regression)

$$\sum_{j=1}^{J} \sum_{i \in R_j} (y_i - \hat{y}_{R_j})^2$$

- Steps:
  1. Start with all features and find *j* and *s* that minimize RSS
  2. Repeat the process within the result regions
  3. Continue until a stopping criteria is reached

$$\sum_{i:\, x_i \in R_1(j,s)} (y_i - \hat{y}_{R_1})^2 + \sum_{i:\, x_i \in R_2(j,s)} (y_i - \hat{y}_{R_2})^2, \ldots$$



Avoiding overfitting: pruning the tree

- Finding subtrees
  - Experiment different subtrees and choose the one with the less error in CV
  - Cost complexity pruning:
    - Consider a sequence of trees to minimize error based on α terminal nodes

$$\sum_{m=1}^{|T|} \sum_{i:\ x_i \in R_m} (y_i - \hat{y}_{R_m})^2 + \alpha |T|$$

What about classification with decision trees?

- Same principle: instead of predicting the mean value we predict the **most commonly occurring class**

- Instead of minimizing RSS we use classification error rate

  - Gini Index

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

  - Entropy

$$D = -\sum_{k=1}^{K} \hat{p}_{mk} \log \hat{p}_{mk}$$

Lower values indicate "pure" nodes

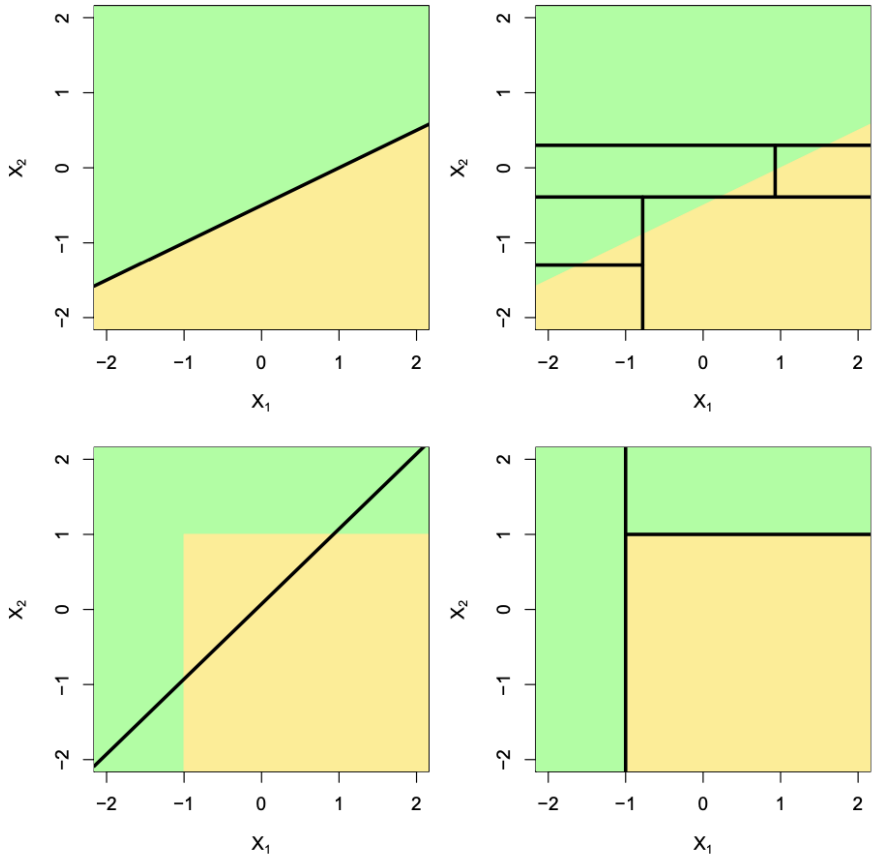
Linear models vs decision trees

- Linear regression

$$f(X) = \beta_0 + \sum_{j=1}^{p} X_j \beta_j,$$

- Decision tree

$$f(X) = \sum_{m=1}^{M} c_m \cdot 1_{(X \in R_m)}$$



Decision trees pros and cons

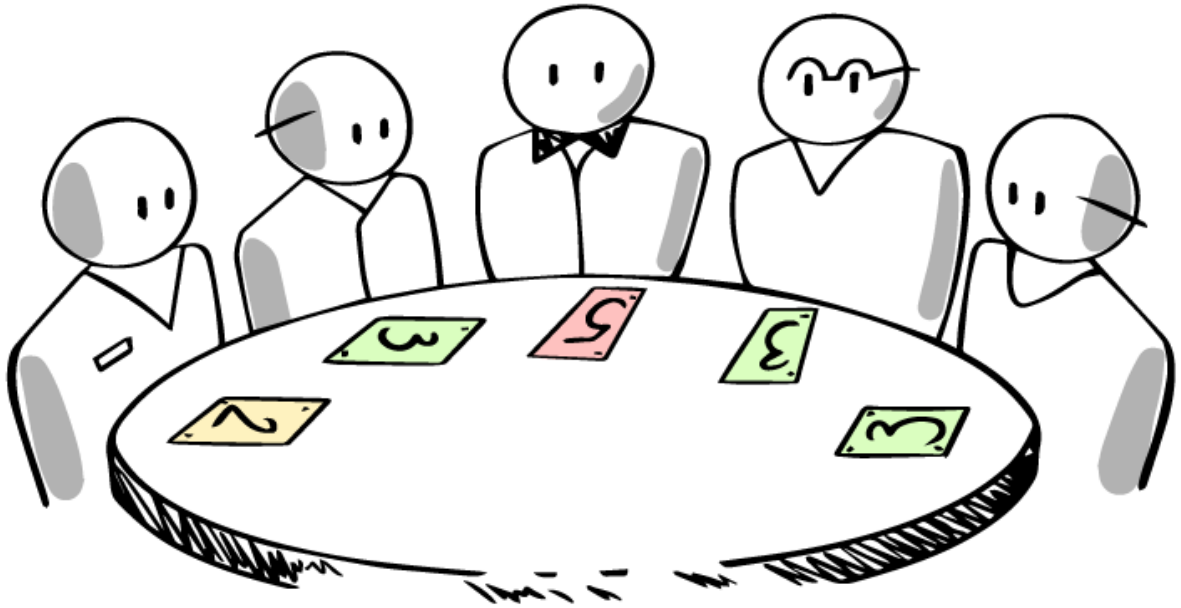
Pros

- Very easy to understand
- Easy to handle qualitative predictors
- No need to standardize data
- Easily displayed graphically

Cons

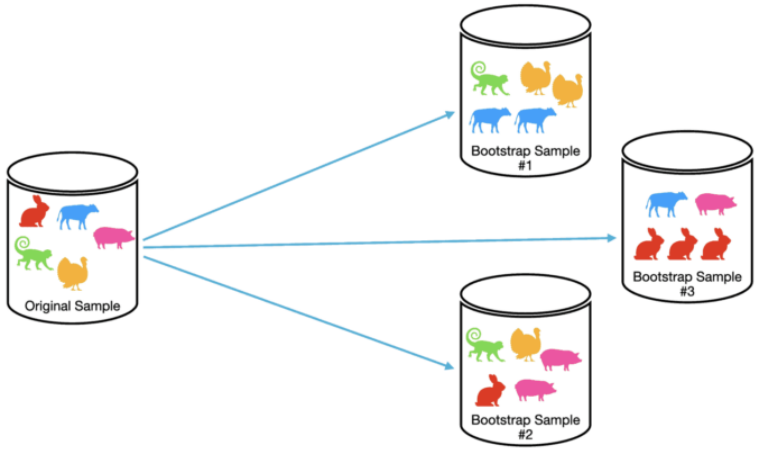
- Often not so accurate as other models
- Non-robust: a small change in data leads to different estimations
- Overfitting!!

From trees to forests



Bagging or Bootstrap aggregation

- Combining prediction of multiple models trained on different subsets of training data
- How it works:
  - Generate multiple sampled from original dataset
  - Each sample will have the same size
  - A model will be trained on each sample
  - Final model (ensemble) will take average predictions (regression) or majority vote (classification) from base learners
- Tradeoff for interpretability!



Regression

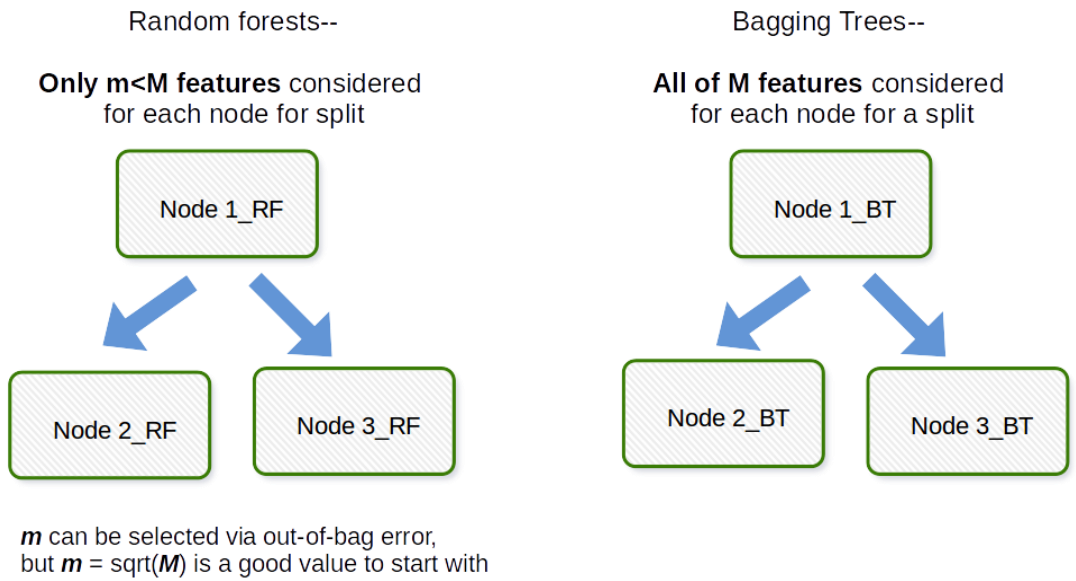
$$\hat{y} = \frac{1}{B}\sum_{i=1}^{B} M_i(x)$$

Classification

$$\hat{y} = \text{mode}(M_1(x), M_2(x), \ldots, M_B(x))$$

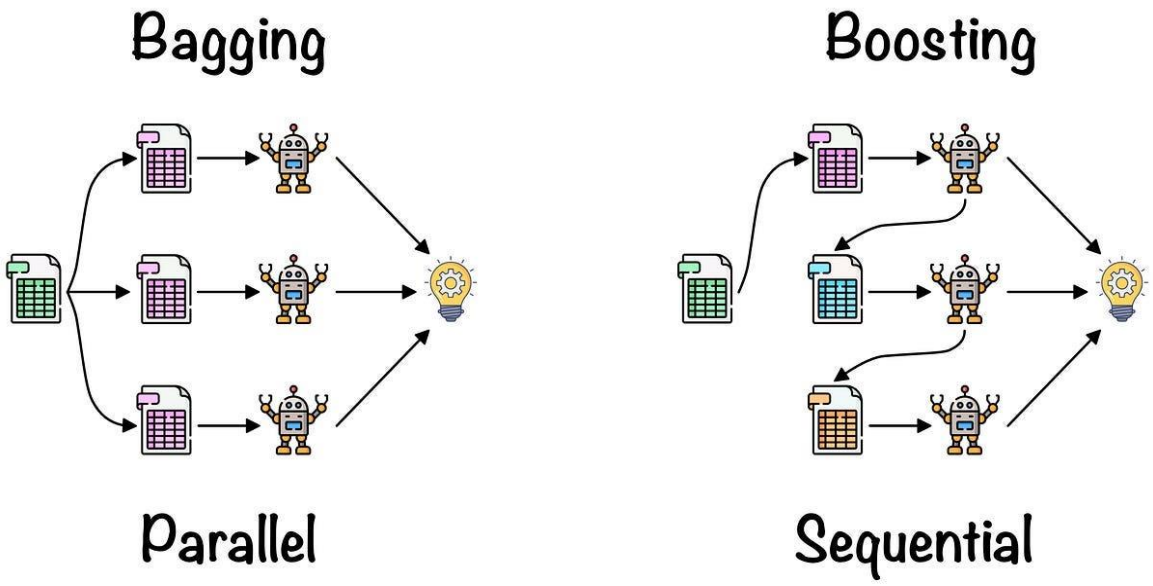
Random forests

- Also generate multiple bootstrap sampled from original training data but in addition, at each split only a **random subset of features** is considered for splitting
  - Typically square root number of total features



Boosting

- Like bagging but trees are grown **sequentially**
  - The input of each model is the residual error of the previous model
  - The goal is to improve where the first model did worse
- Each tree is fit on a modified version of the original dataset
- Leads to models that are good in different "parts" of the training data



Ensemble trees pros and cons

Pros

- Improved accuracy
- Reduced overfitting
- More robust to noise
- Deal well with mixed data types

Cons

- Computationally intensive
- Less interpretable
- Can have complex hyperparameter tuning (boosting machines)