Modelli basati su alberi e - loro interpretazione -

Anna Gottard

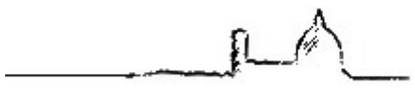
29 maggio 2019

Outline

- Decision trees
- Regression and classification trees
- Bagging and Random forest
- Conditional inference trees and forest
- Bart
- Interpreting and understanding

Remember: as these algorithms are not set according to theoretical assumptions, you need to use

- training/testing sets
- cross-validation



General setting

- A vector of explanatory variables X = (X_1, … X_p) and a response Y are observed on a sample of iid statistical units

- We want to predict Y assuming that

$$Y = f(X) + \epsilon \qquad\qquad f(X) = E[Y \mid X]$$

- The best predictor is the function that minimises among all the possible functions g(X) a loss function, such as for instance the mean squared error (if Y continuous)

$$\text{Mean}\left[\left(Y - g(X)\right)^2 \mid X = x\right]$$

- Let's call such function $\widehat{f(X)}$

Example : Multiple linear regression

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \epsilon$$

- Given estimates $\hat{\beta}_0, \hat{\beta}_1, \ldots \hat{\beta}_p$, we can make predictions using the formula

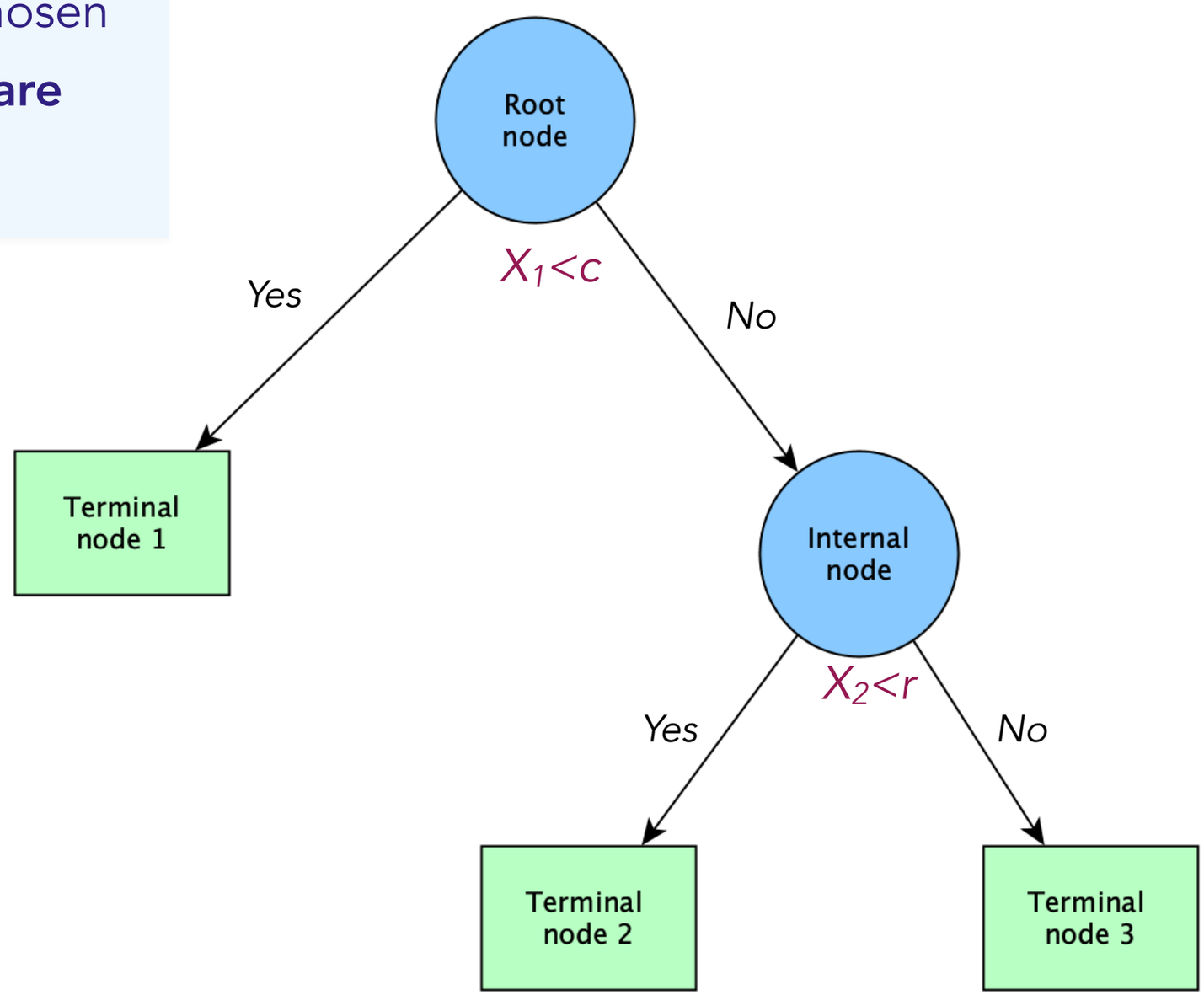
$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \cdots + \hat{\beta}_p x_p.$$

$$\begin{aligned} \text{RSS} &= \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \\ &= \sum_{i=1}^{n} (y_i - \hat{\beta}_0 - \hat{\beta}_1 x_{i1} - \hat{\beta}_2 x_{i2} - \cdots - \hat{\beta}_p x_{ip})^2 \end{aligned}$$

$$MSE = \frac{RSS}{n}$$

Decision trees

→ IDEA: To find a piecewise-constant approximation of f(X) to predict Y, chosen in a way that mimics how **decisions are actually taken**



→ EXAMPLE:

Y = treatment

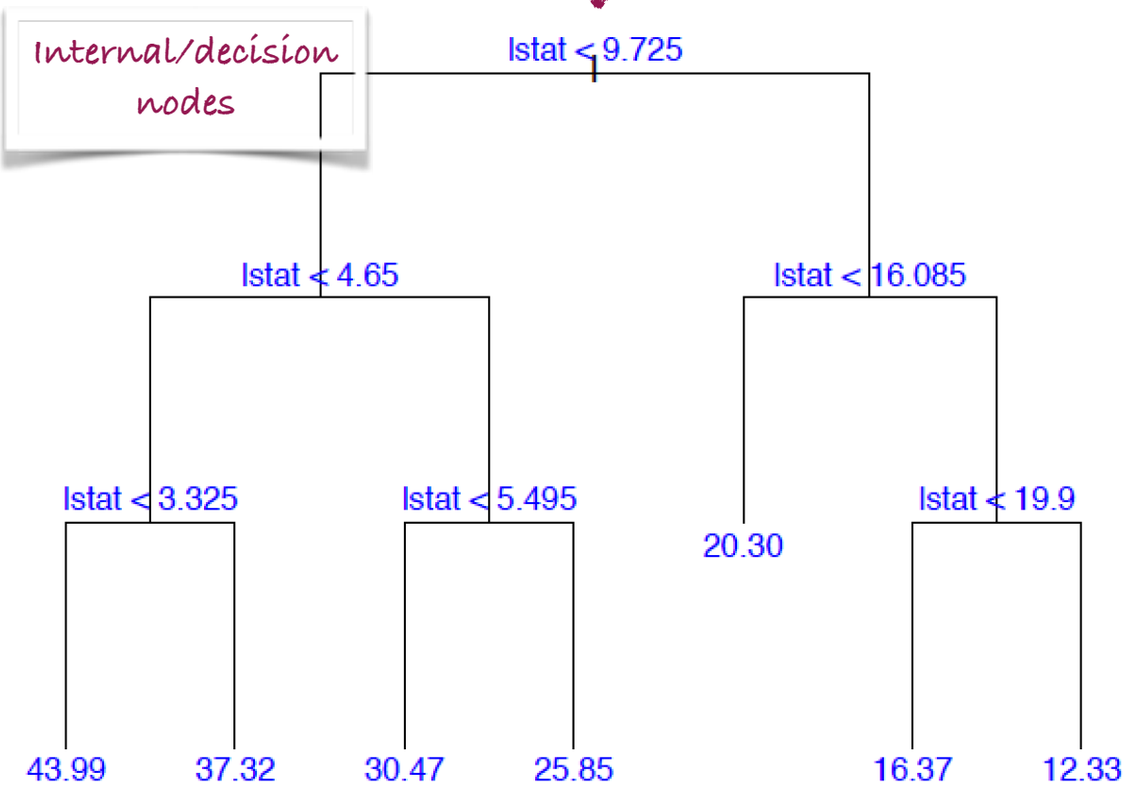
X_1 = Fever

X_2 = Pain

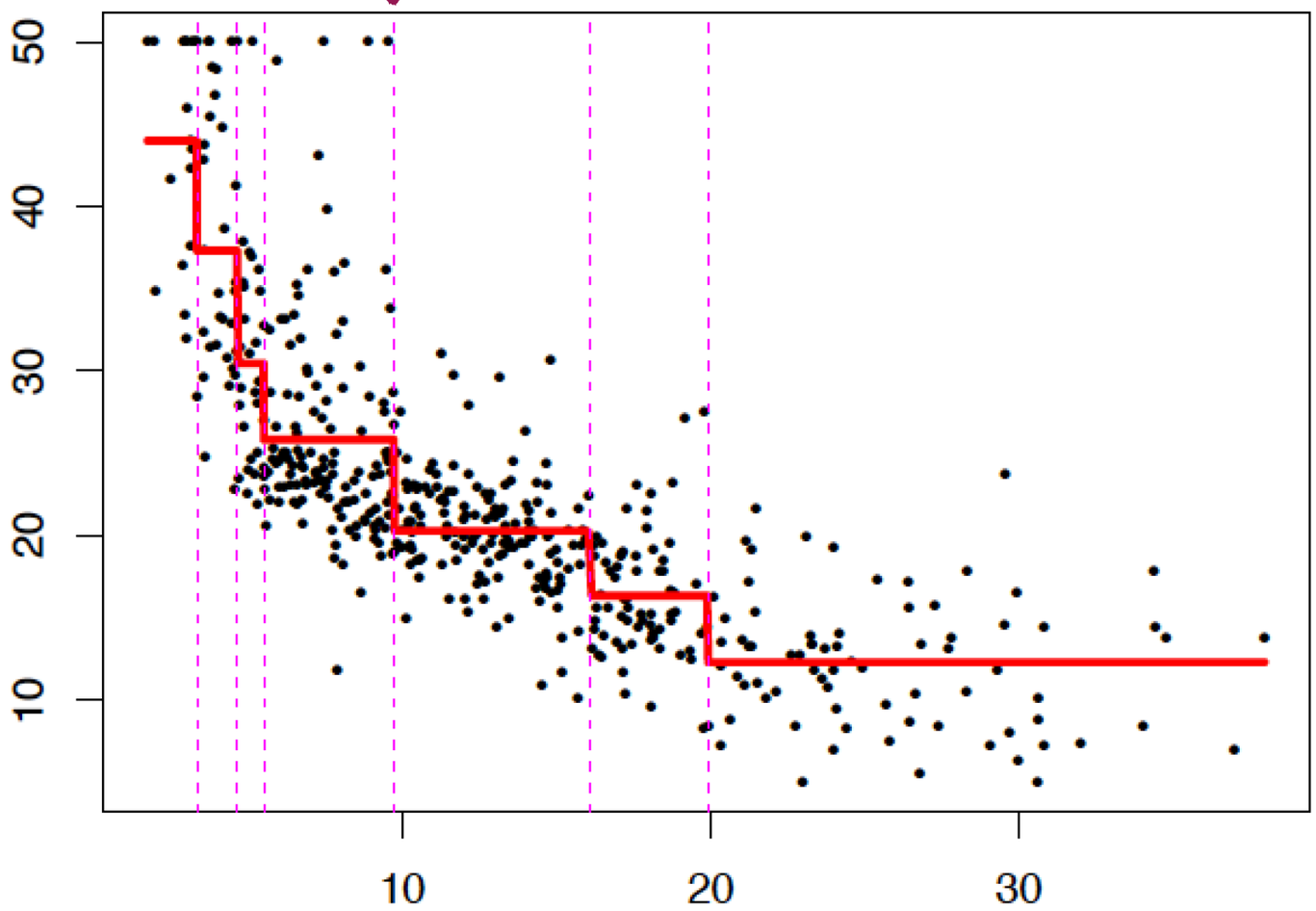
Example with only 1 covariate

First split

Internal/decision nodes



Terminal nodes / leaves

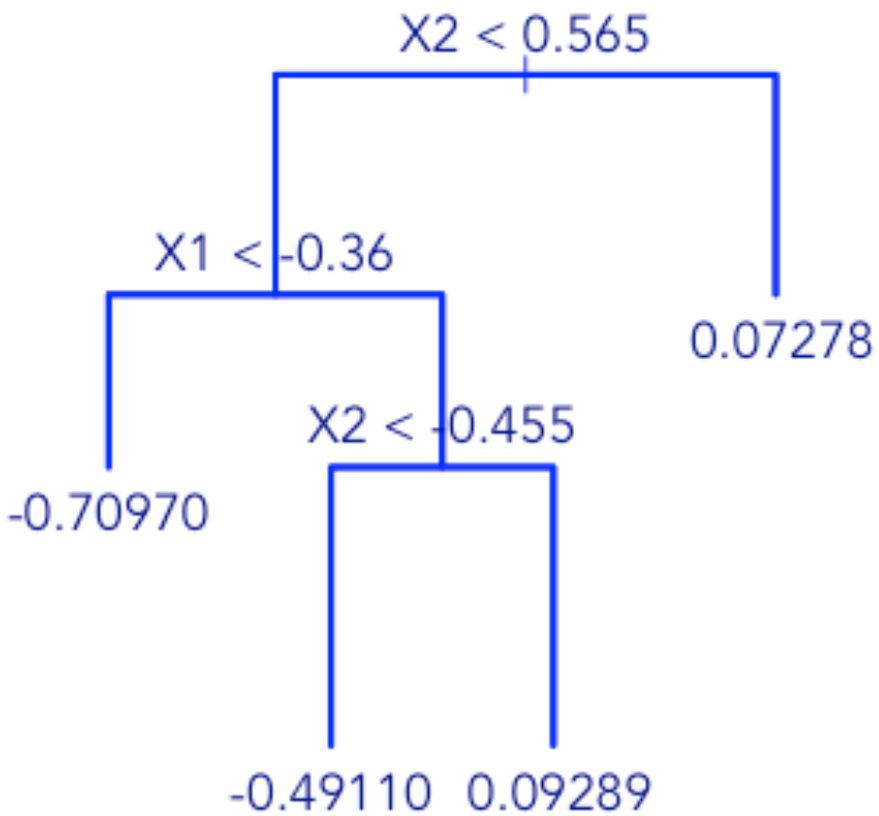


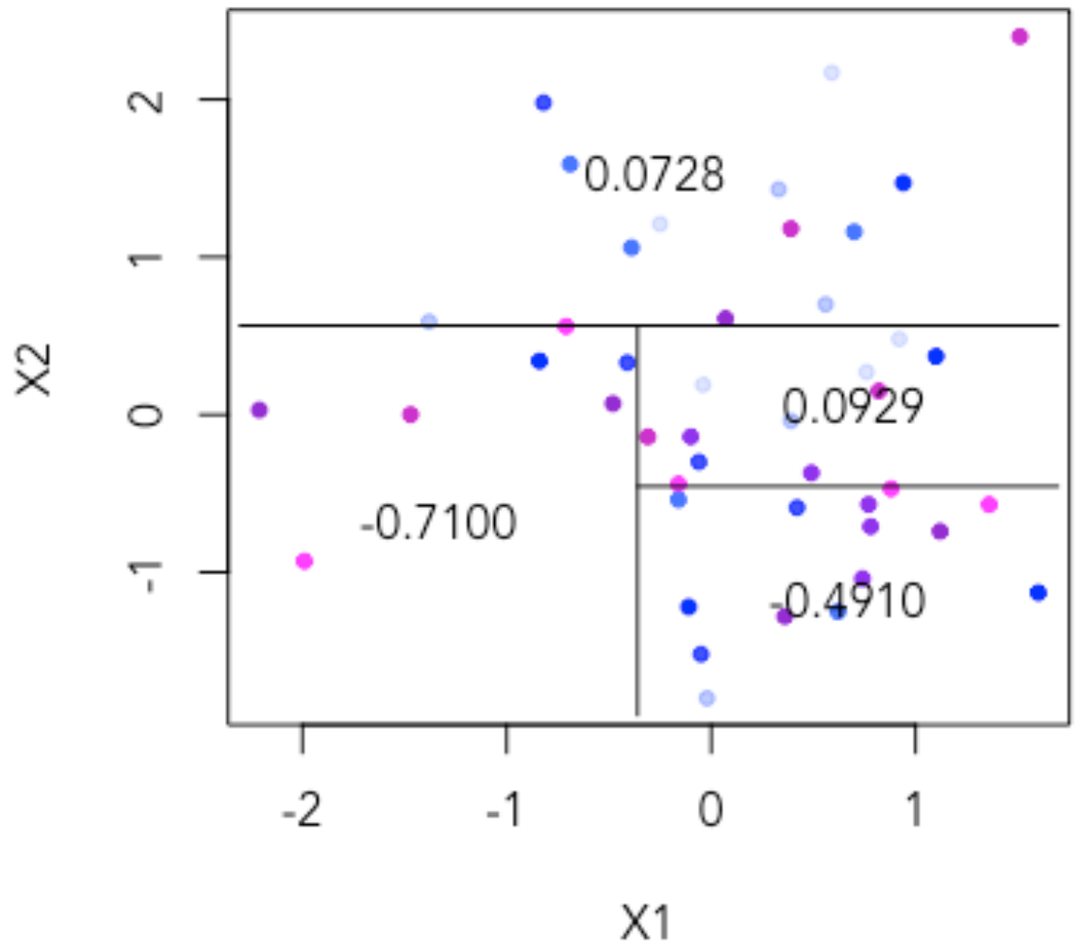
Induced partitioning

Example : simulated data, 2 covariates

$$Y = \sin(X_1) \cdot \sin(X_2) + \epsilon, \qquad \epsilon \sim N(0,1)$$
$$X_1, X_2 \sim N(0,1)$$

- A decision tree is a structure organised hierarchically
- The tree structure is equivalent to partition the joint space of the explanatory variables into M (= n. leaves) subspaces



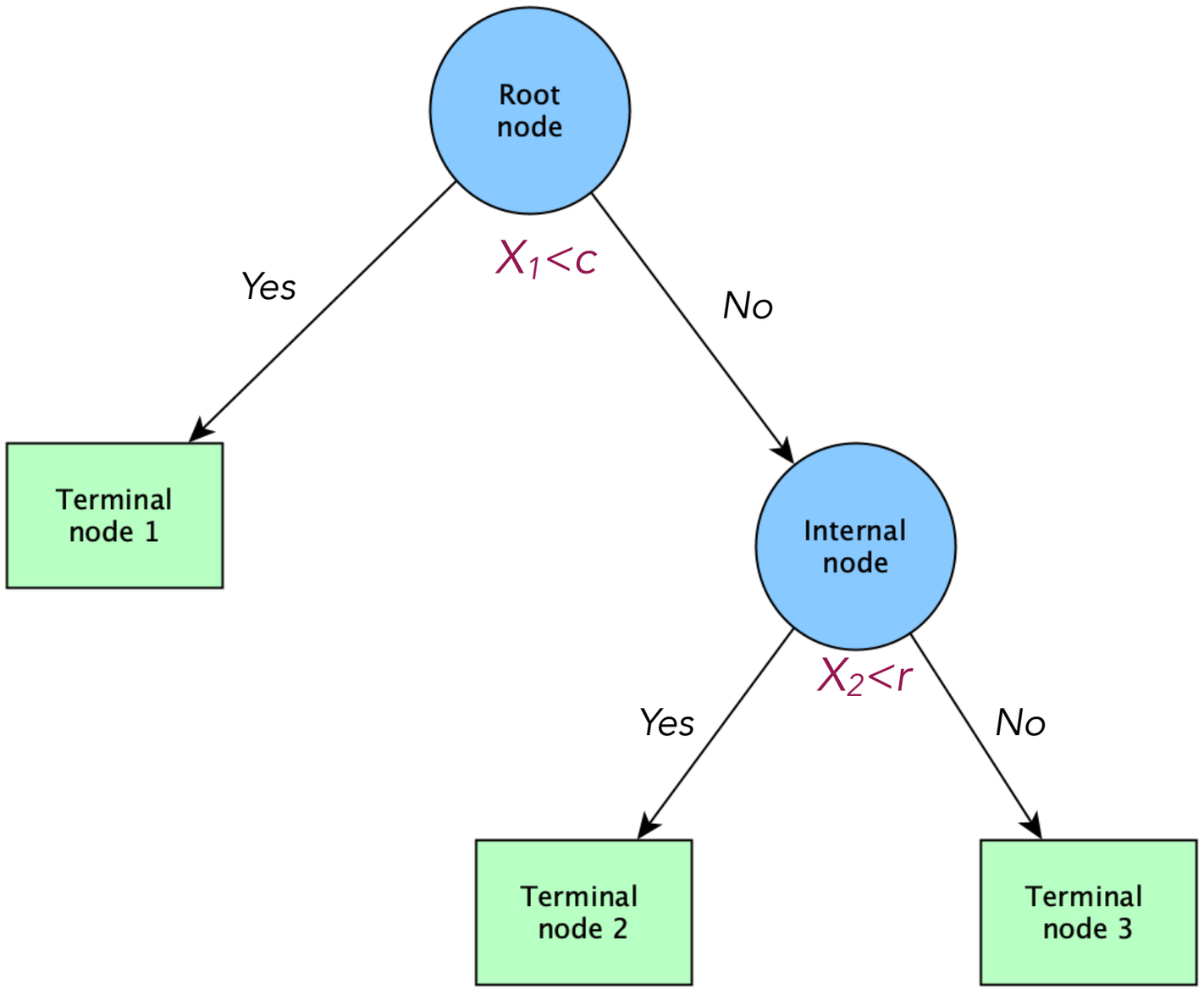


The number in each leaf is the mean of the response for the observations that fall there.

Regression trees

→ We need to learn:

- structure of the tree
- which variable splits and where
- predictions in each leaf/region



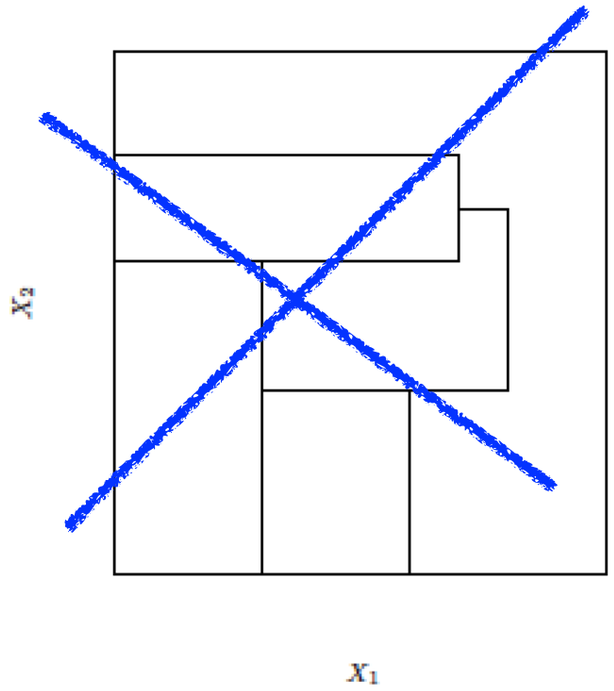
Recursive binary splitting: a top-down, greedy approach

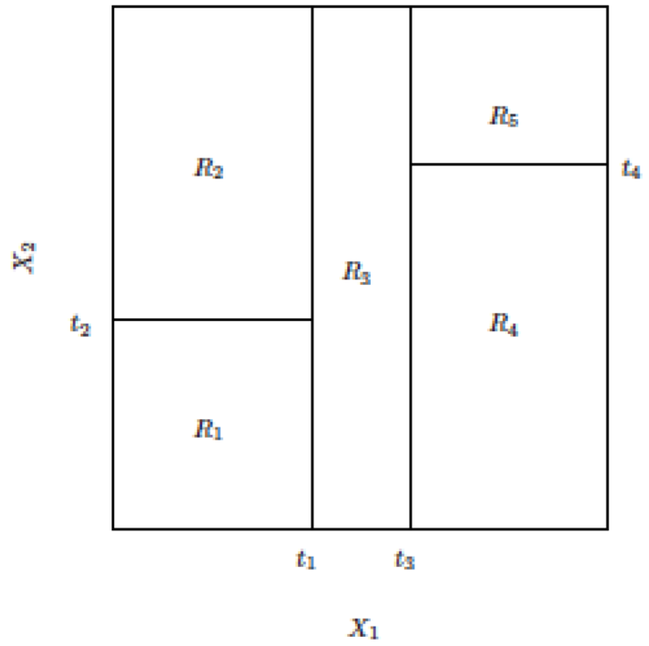
- The approach is **top-down** because it begins at the top of the tree (at which point all observations belong to a single region) and then successively splits the covariate space

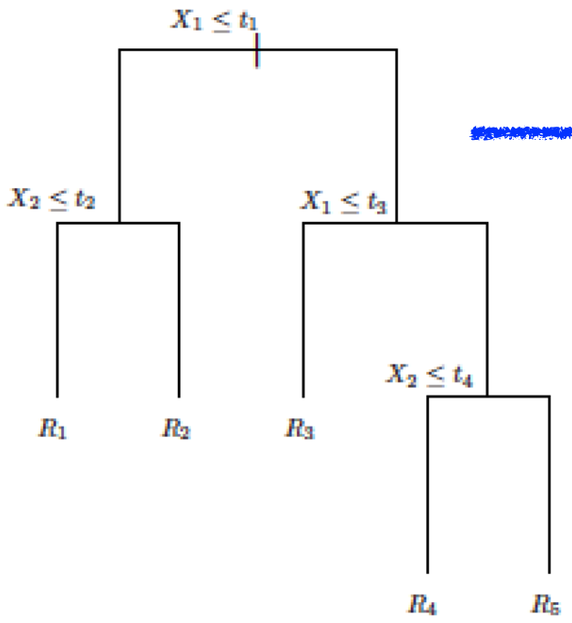
- **Binary**: Each split is indicated via **two new branches** further down on the tree

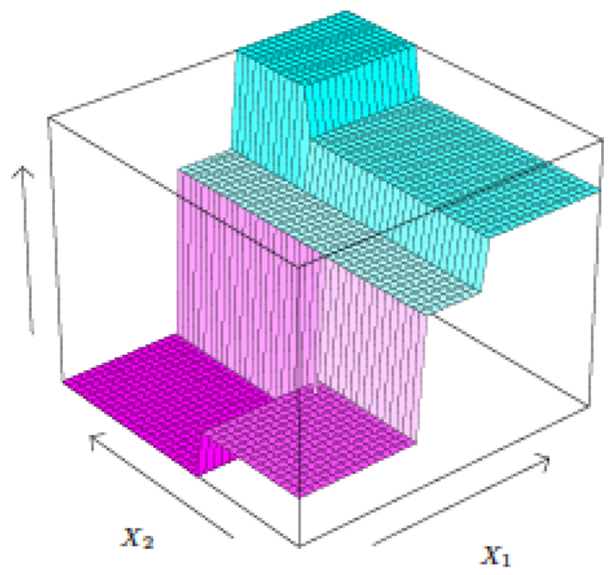
- It is **greedy** because at each step of the tree-building process, the best split is made at that particular step, rather than looking ahead and picking a split that will lead to a better tree in some future step.

The tree-building process examples









Regression trees from a regression perspective

(1) Start with $M = 1, \qquad R_1 = \mathbb{R}^p$

(2) Search for the first split:

j=1, ..., p

$$(j, s_1) \; : \; R_1 = \{(X_1, \ldots, X_p) \in \mathcal{X} : X_j \leq s_1\}, \quad R_2 = \{(X_1, \ldots, X_p) \in \mathcal{X} : X_j > s_1\}$$

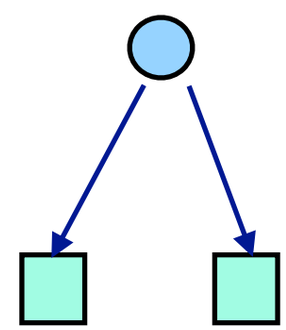
$$\min_{j, s_1} \left[\min_{\mu_1} \sum_{i : x_{ij} \in R_1} (y_i - \mu_1)^2 + \min_{\mu_2} \sum_{i : x_{ij} \in R_2} (y_i - \mu_2)^2 \right]$$

This corresponds to finding j and s_1 that minimise the MSE of the one-factor regression model

$$Y_i = \mu_1 \, \mathbb{I}_{\{X_{ij} \leq s_j\}} + \mu_2 \, \mathbb{I}_{\{X_{ij} > s_j\}} + \varepsilon_i$$

$$\Rightarrow \quad \widehat{\mu}_m = \bar{y}_{R_m} \quad m = 1, 2$$

Estimate/Prediction



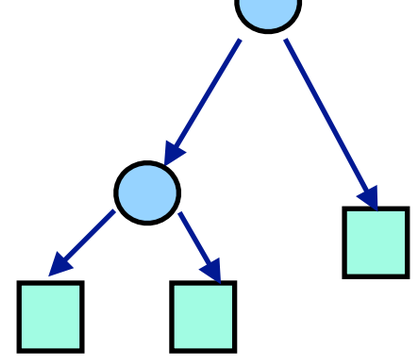
Regression trees from a regression perspective

(3) Search for the second split: repeat the procedure (2) within R_1 or R_2

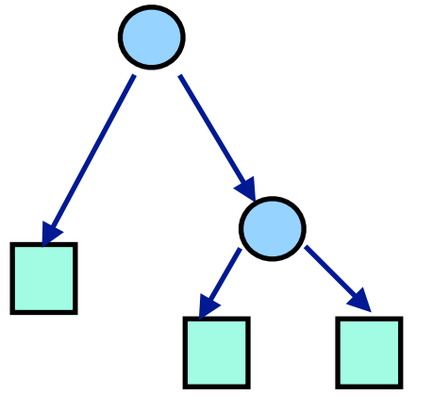
R_1 , R_2 → R_1 , R_2 , R_3 minimising the loss function (MSE)

This corresponds to finding k and s_2 that minimise the MSE in one of the one-factor regression models

$$Y_i = \mu_1 \mathbb{I}_{\{X_{ij} \leq s_1\}} \mathbb{I}_{\{X_{ik} \leq s_2\}} + \mu_2 \mathbb{I}_{\{X_{ij} \leq s_1\}} \mathbb{I}_{\{X_{ik} > s_2\}} + \mu_3 \mathbb{I}_{\{X_{ij} > s_1\}} + \varepsilon_i,$$



$$Y_i = \mu_1 \mathbb{I}_{\{X_{ij} \leq s_1\}} + \mu_2 \mathbb{I}_{\{X_{ij} > s_1\}} \mathbb{I}_{\{X_{ik} \leq s_2\}} + \mu_3 \mathbb{I}_{\{X_{ij} > s_1\}} \mathbb{I}_{\{X_{ik} > s_2\}} + \varepsilon_i$$

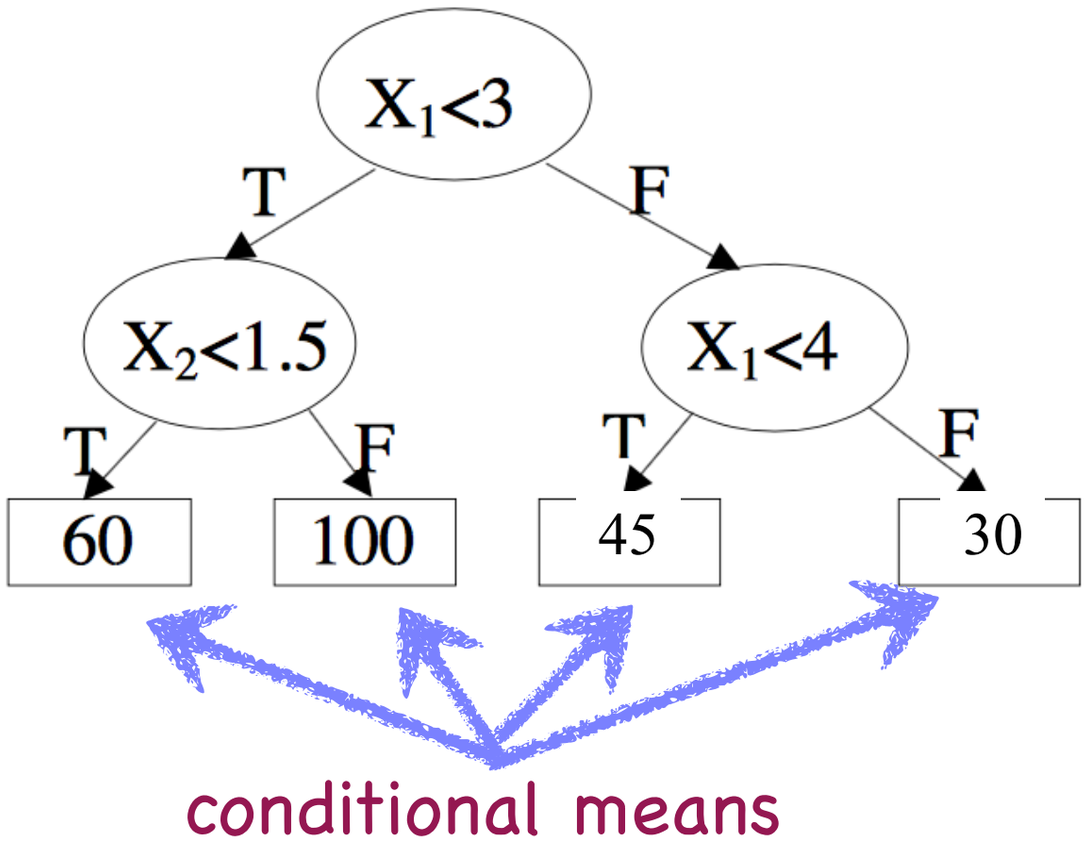


$$\Rightarrow \quad \widehat{\mu}_m = \bar{y}_{R_m} \quad m = 1, \ldots, 3$$

Estimate/Prediction

The tree model : assigning a value to each terminal node

$$\mathbb{E}[Y \mid X = x] = \sum_{m=1}^{M} \mu_m \mathbb{I}_{\{x \in R_m\}}$$



$$R_1 : \; x_1 < 3 \wedge x_2 < 1.5 \Rightarrow \mu_1 = 60$$

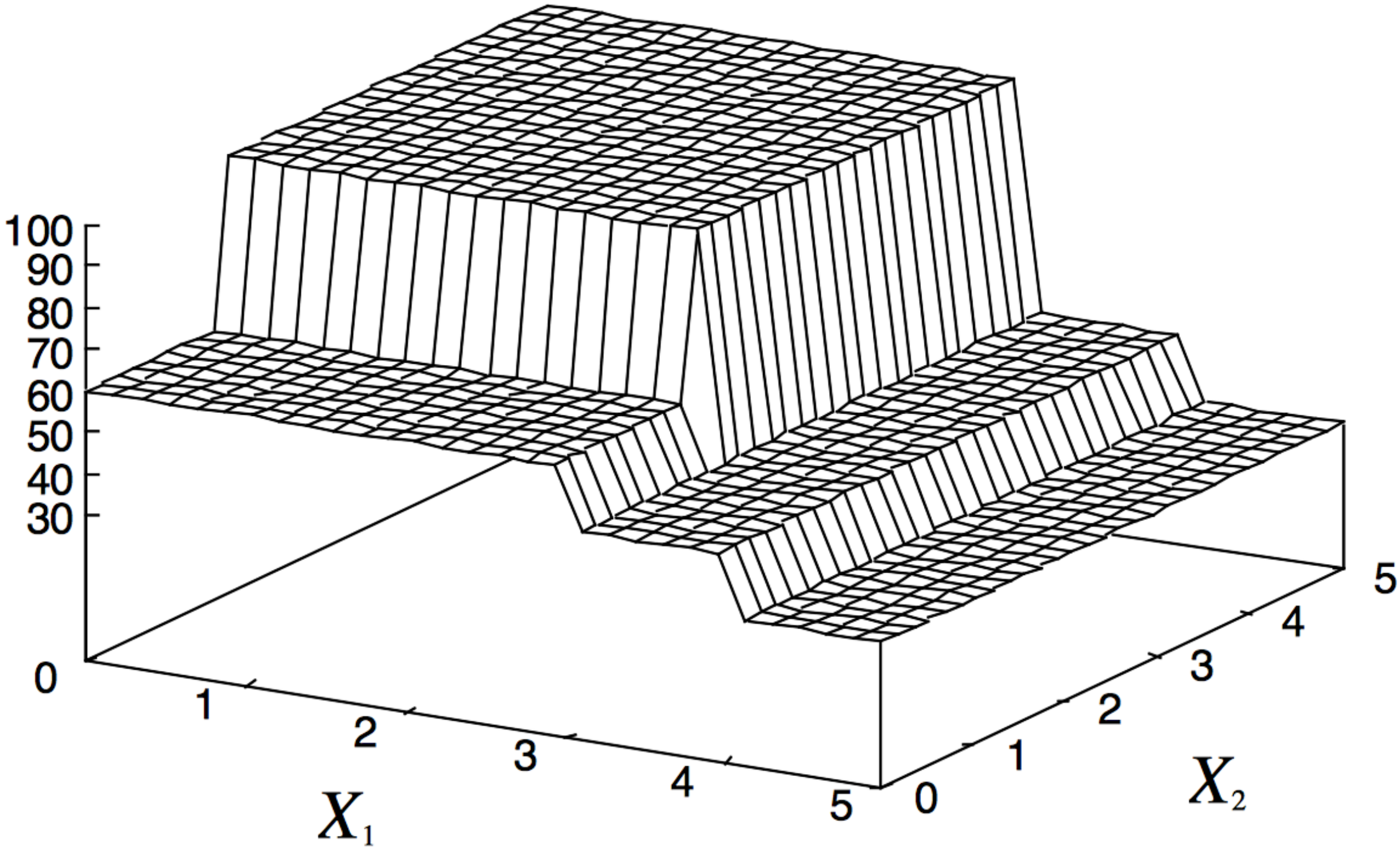
$$R_2 : \; x_1 < 3 \wedge x_2 \geq 1.5 \Rightarrow \mu_2 = 100$$

$$R_3 : \; x_1 \geq 3 \wedge x_1 < 4 \Rightarrow \mu_3 = 45$$

$$R_4 : \; x_1 \geq 3 \wedge x_1 \geq 4 \Rightarrow \mu_4 = 30$$

The tree model

$$\mathbb{E}[Y \mid X = x] = \sum_{m=1}^{M} \mu_m \mathbb{I}_{\{x \in R_m\}}$$



The tree-building algorithm: stopping rule

STOPPING RULE

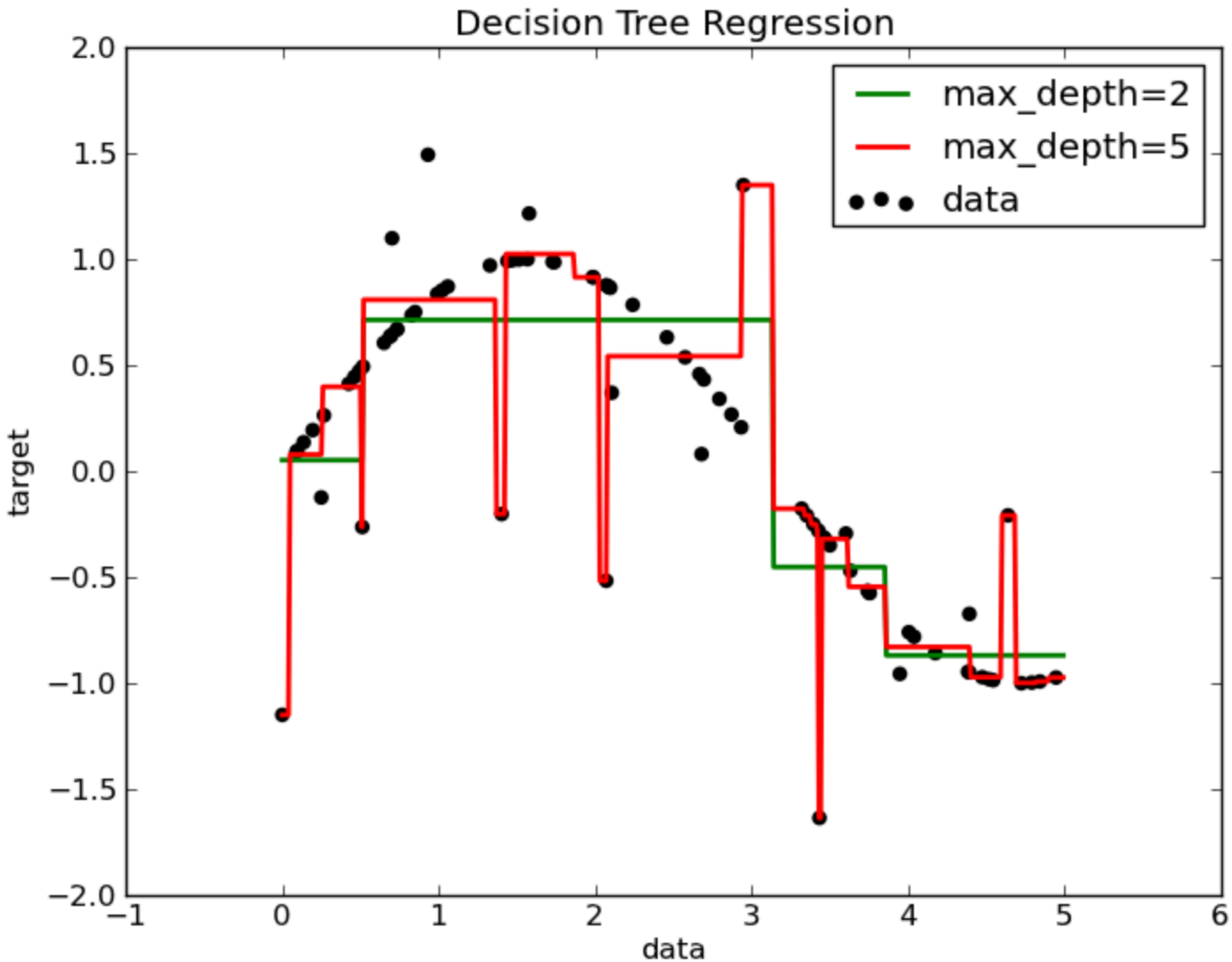
On the training data:

To avoid to have leaves with only one unit - perfect (over)fitting

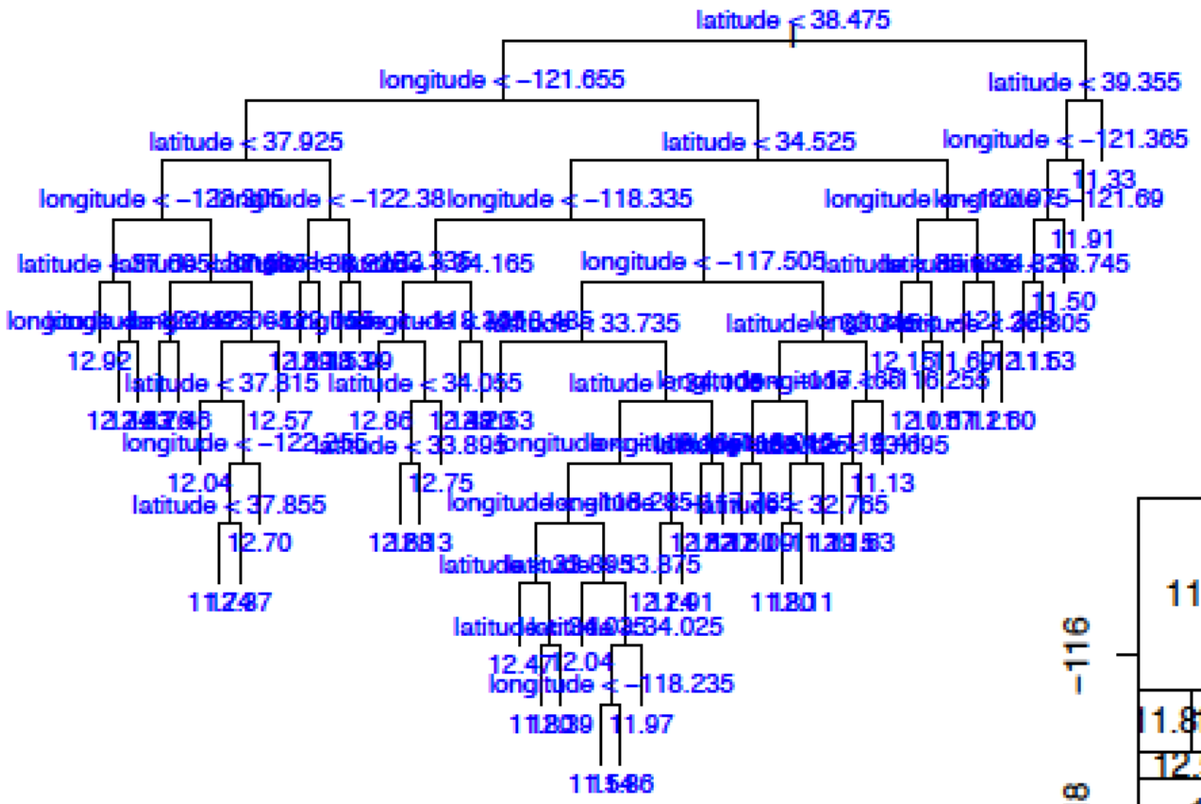
- stop if the node contains less than a pre-specified minimum node size (usually 5, but depends on n)

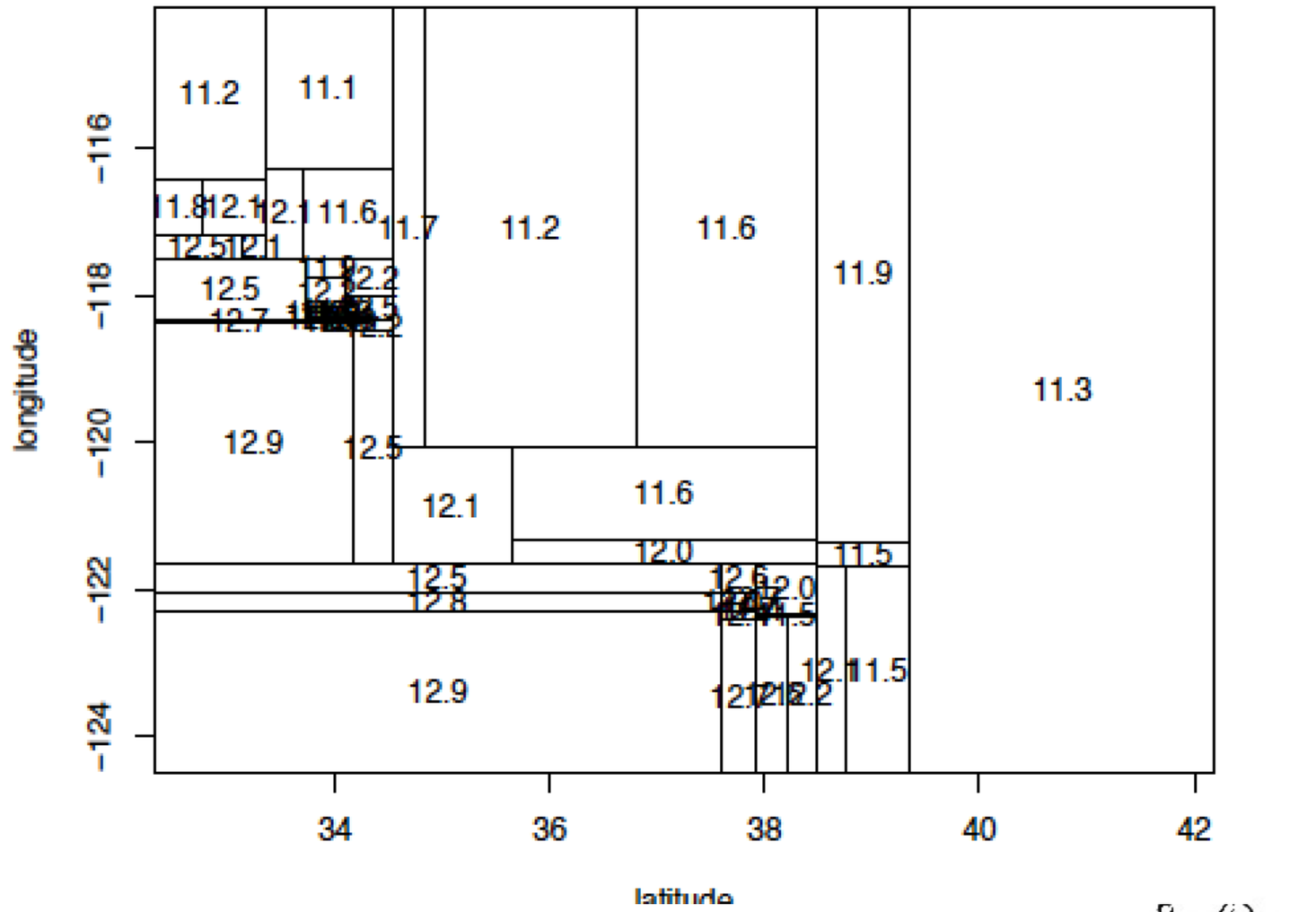
- stop if the pre-specified maximum tree depth limit is reached

Stopping rule : Example with only 1 covariate



Stopping rule : Example with 2 covariate2





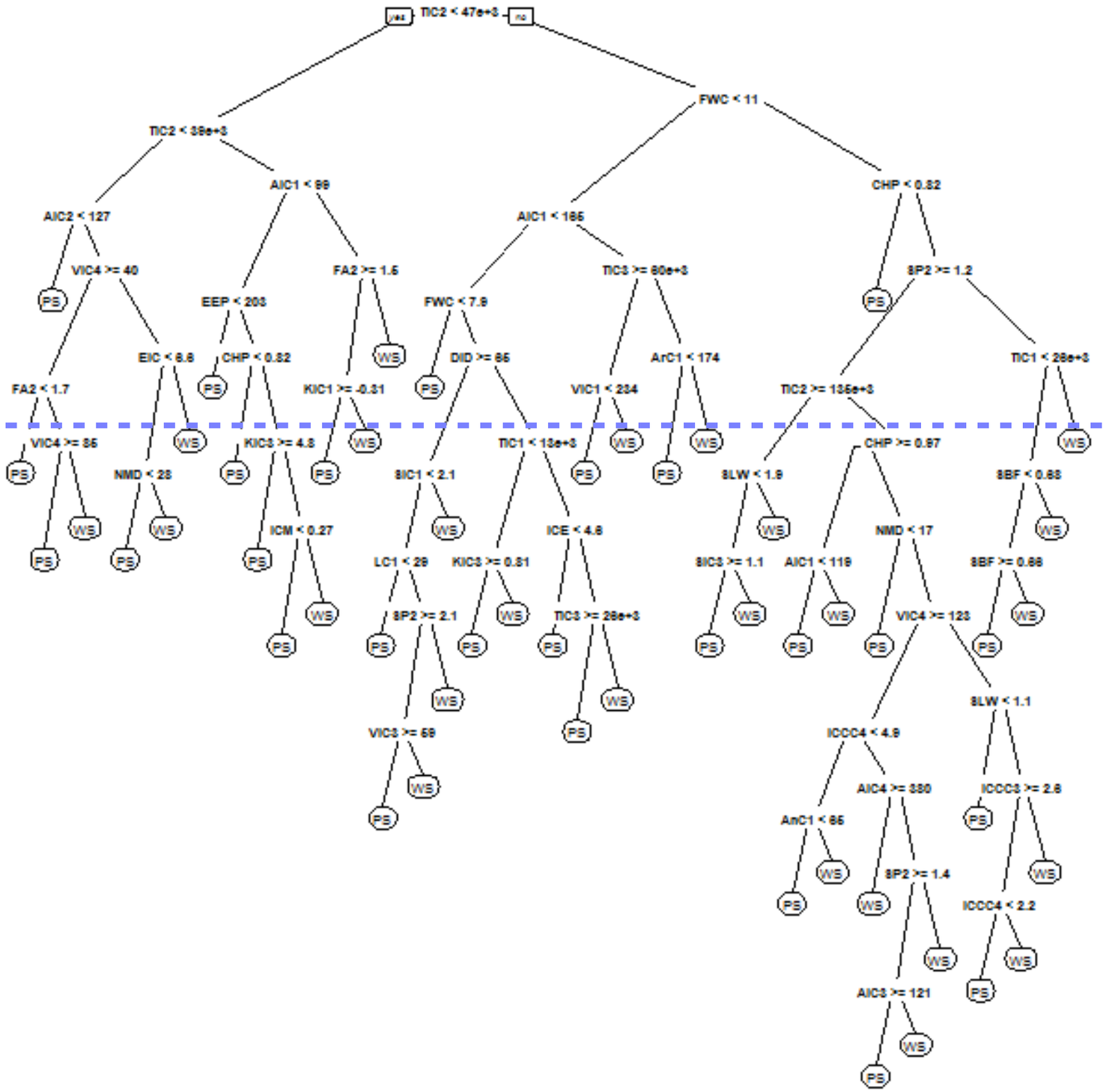
Pruning a tree

- This process may produce good predictions on the training set, but is likely to **overfit** the data, leading to poor test set performance.

- A smaller tree with fewer splits (fewer regions R_1 … R_J) might lead to lower variance and better interpretation at the cost of a little bias.

- **Good strategy**: grow a very large tree then **prune** it back

Pruning a tree



Pruning a tree : cost-complexity measure

Cost-complexity pruning :

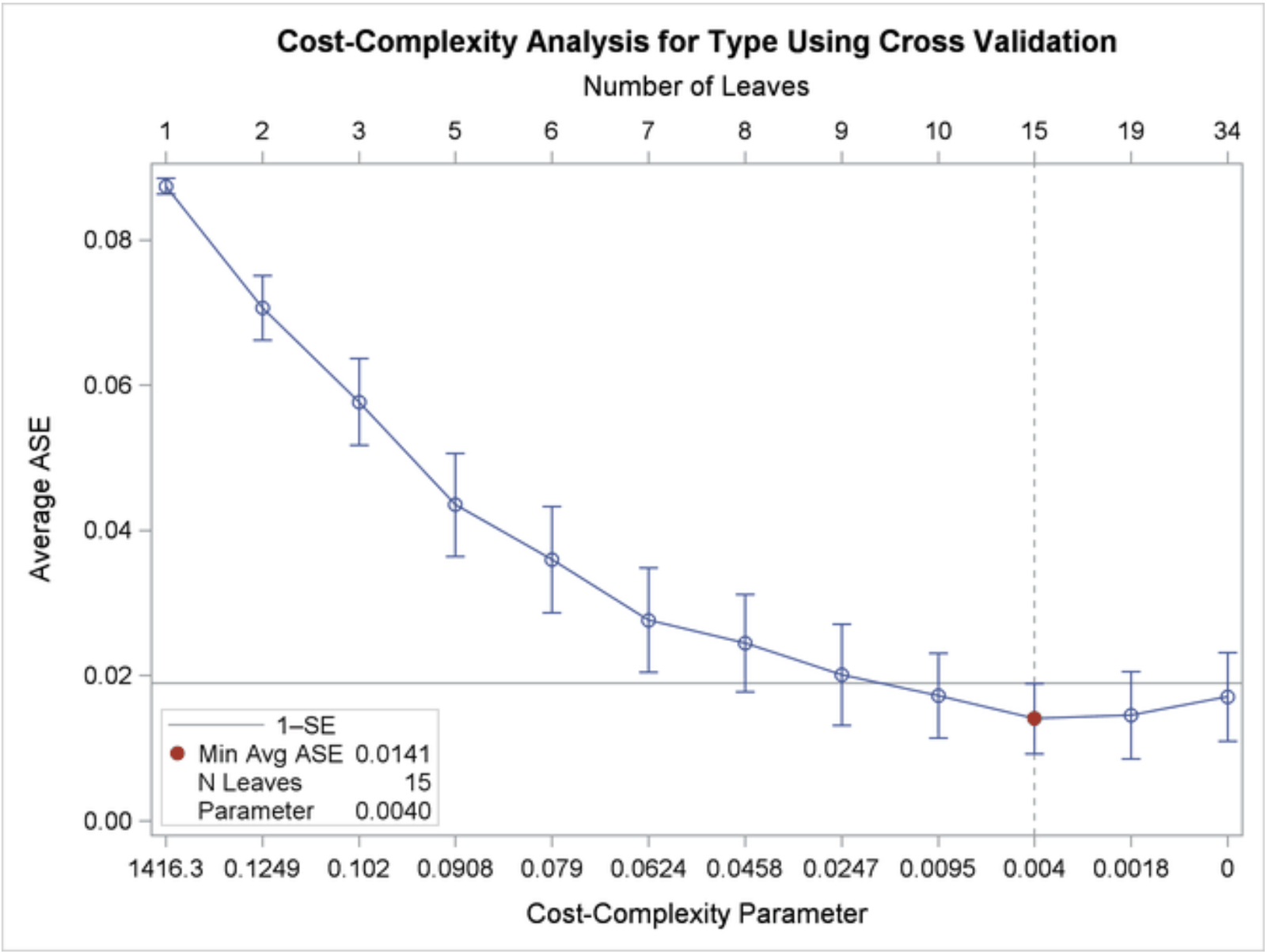
Construct a sequence of sub-trees, pruned at different depth d, having numbers of nodes varying from 1 to $|T_d|$

Compute the cost complexity measure for the tree, which is based on

$$CP(d) = \sum_{m=1}^{|T_d|} \sum_{i:x_i \in R_m} (y_i - \mu_m)^2 + \alpha |T_d|$$

where α is a non-negative regularization parameter controlling the trade-off between the tree complexity and its fitting

Pruning a tree : Choosing the best subtree



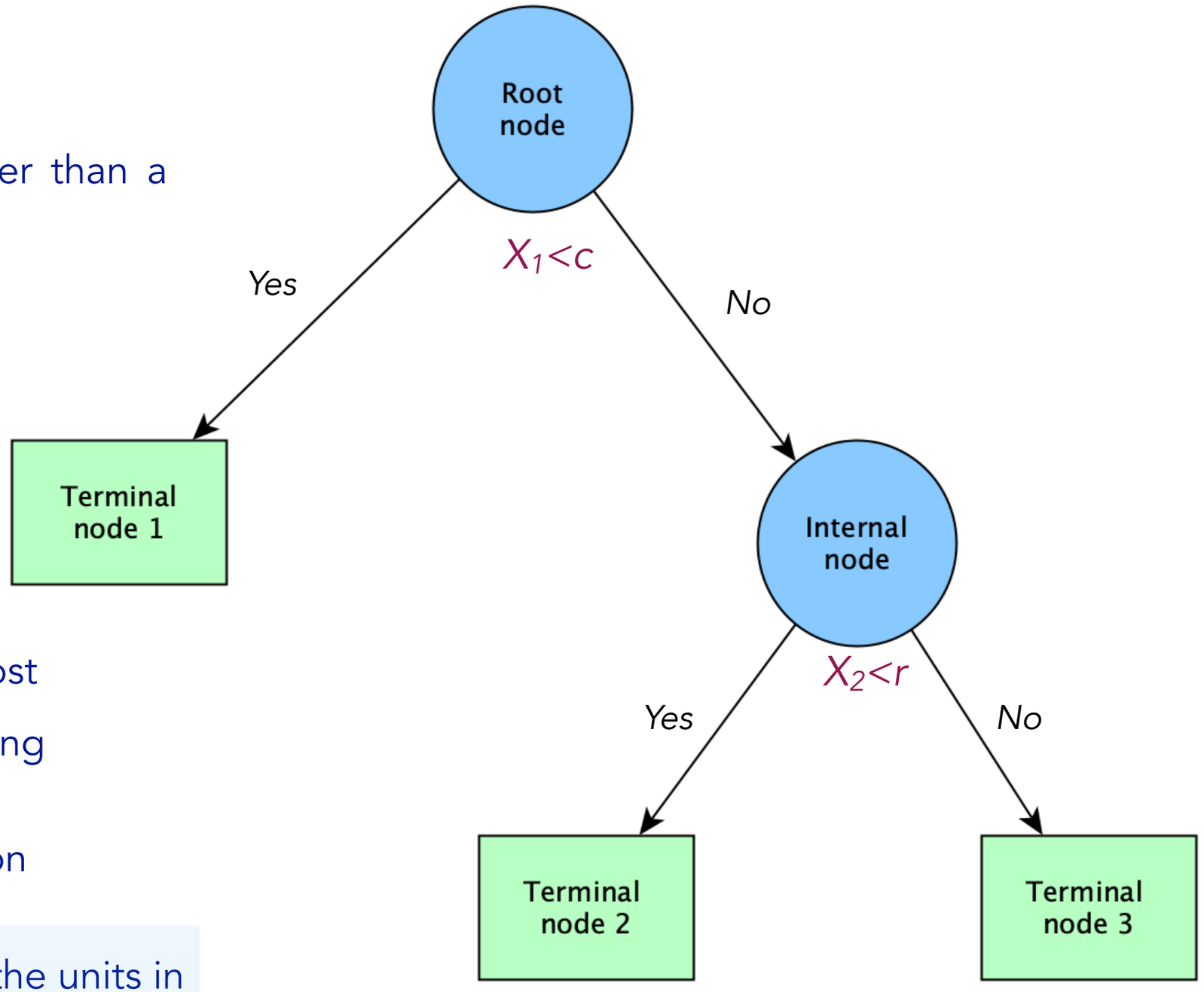
Choose α:

- α controls the trade-off between complexity and fitting
- optimal value chosen using cross-validation
- Then fit the tree on full data using the chosen optimal value

Classification trees

Classification

- Very similar to a regression tree,
- For qualitative responses rather than a quantitative one
- Response classes: 1, …, K



- Prediction at each node: the most common class in the corresponding
- Need to change the loss function
- IDEA: the more homogeneous the units in the leaves the better

Splitting Rule : purity/ impurity measures

-> Proportion of units in node m having Y = k

$$\hat{p}_{mk} \qquad m = 1, \ldots M \qquad k = 1, \ldots K$$

-> **Gini index** for node m $G_m = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk})$

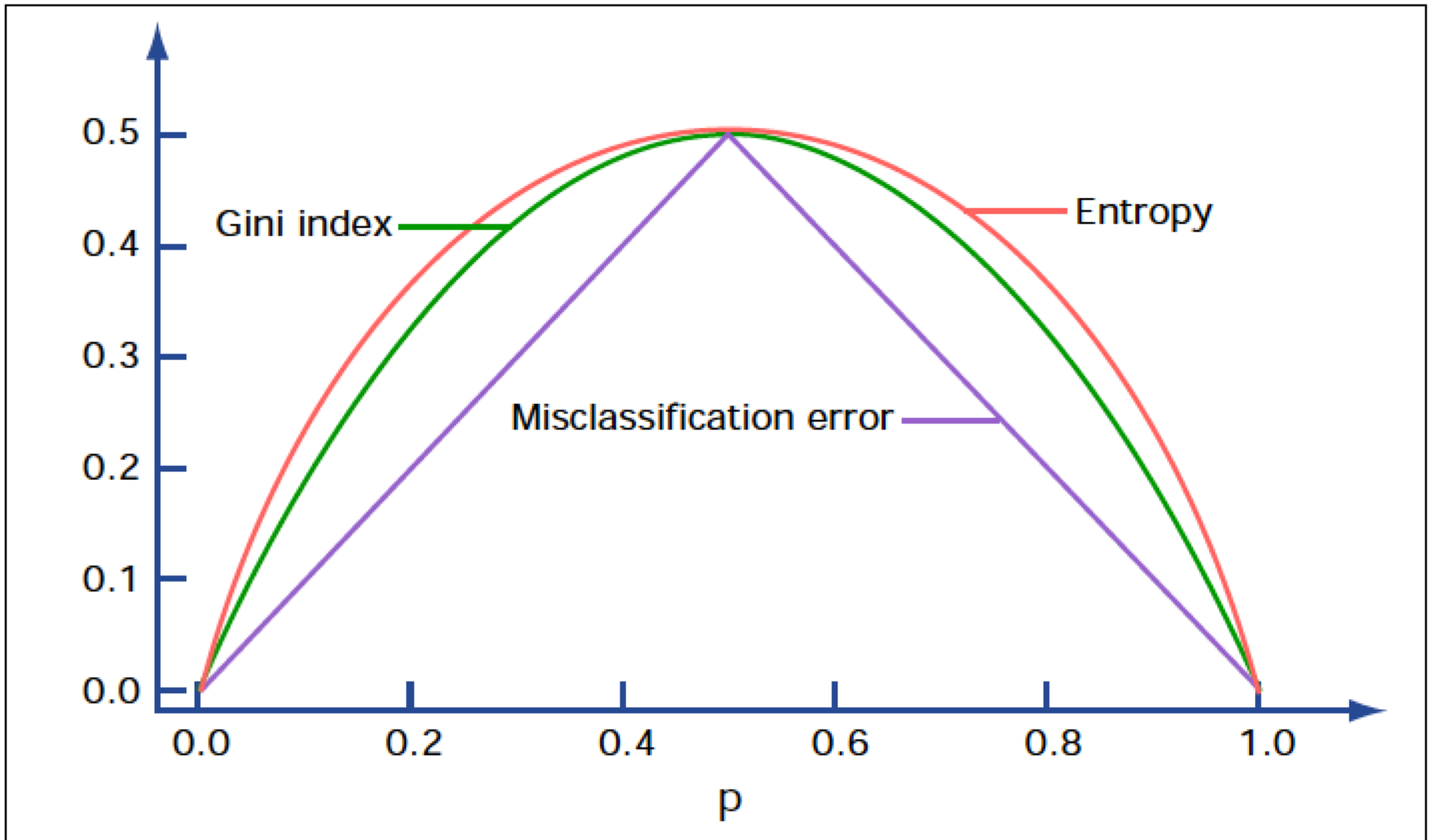
Variance of a Bernoulli distribution

Total variance

-> **Cross entropy** for node m $D_m = -\sum_{k=1}^{K} \hat{p}_{mk} \log \hat{p}_{mk}$

-> **Misclassification error** for node m $E_m = 1 - \max_k \hat{p}_{mk}$

Splitting Rules : impurity measures



Node impurity measures for two-class classification, as a function of the proportion p in class 2. Cross-entropy has been scaled to pass through (0.5, 0.5).

Image by MIT OpenCourseWare, adapted from Hastie et al., *The Elements of Statistical Learning*, Springer, 2009.

Splitting and stopping rules

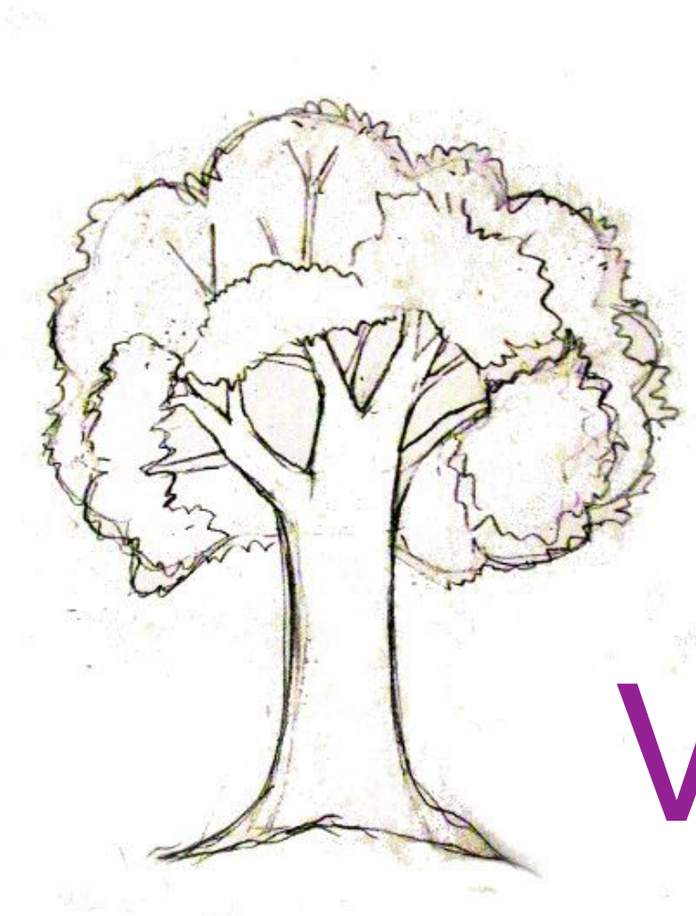
- Gini index and cross-entropy are quite similar numerically

- Cross-entropy and the Gini index are more sensitive to changes in the node probabilities than the misclassification rate

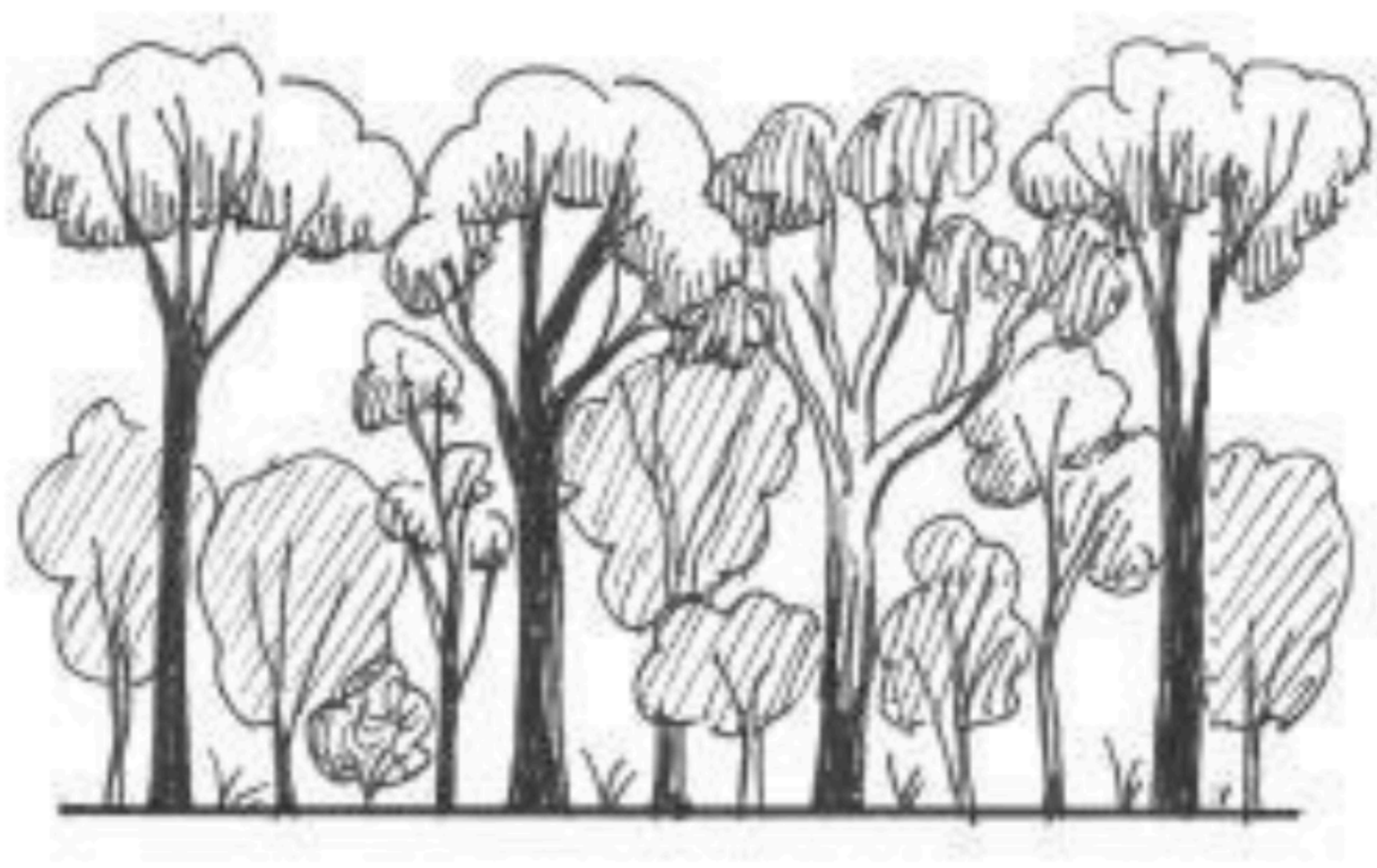
- Gini index or the cross-entropy are typically used to evaluate the quality of a particular split

- Any of these three approaches might be used when pruning the tree but Classification error rate is preferable for comparing the prediction accuracy of the final pruned tree

Ensamble methods



VS



Many beats one

- Trees are someway easier to explain to people than other predictive procedures : the **tree plot** makes them easy to understand

- Trees can naturally deal with interactions and non-linearities, continuous and continuous predictors and responses

- But they cannot boast a great predictive performance

- Can we use more trees?

Bagging
(Bootstrap aggregation)

Bagging

- BAGGING = Bootstrap AGGregation

- To have more trees we need to introduce some variability among the trees: **grow each tree on a different bootstrap sample**

- Averaging many trees reduces the variability of the prediction

- Bagging grows B trees, taking advantage of resampling techniques

Many beat one

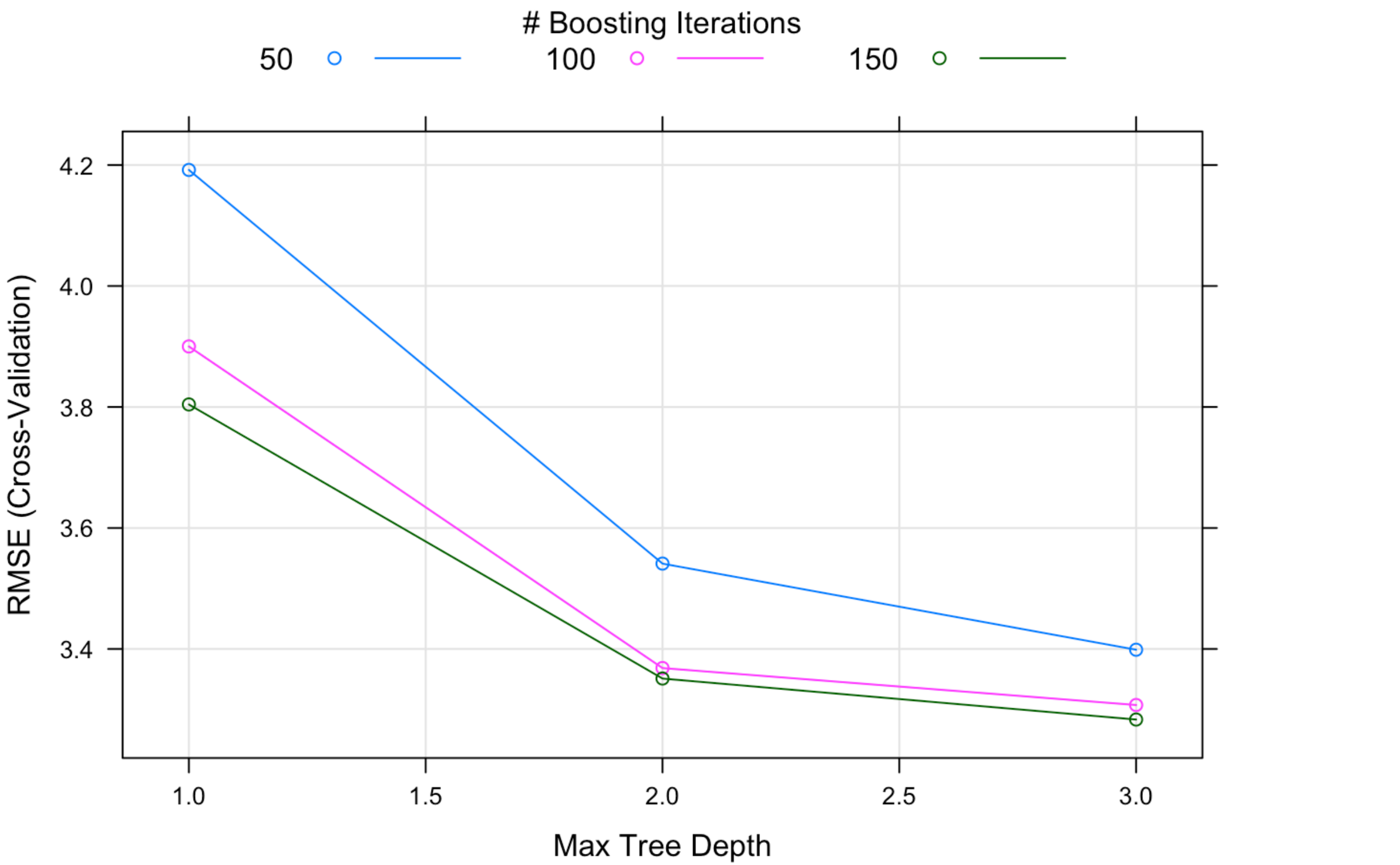
- Trees are someway easier to explain to people than other predictive procedures : The **tree plot** makes them easy to interprete

- They can naturally deal with interactions and non-linearities, continuous and continuous predictors and responses

- But they cannot boast a great predictive performance

- Can we have more trees?

Choice of B


Boosting Iterations
50
100
150
RMSE (Cross-Validation)
4.2
4.0
3.8
3.6
3.4
1.0
1.5
2.0
2.5
3.0
Max Tree Depth

Out-of-Bag prediction

- On average, each bagged tree uses of around 2/3 of the observations
- The remaining 1/3 of the units, not used to fit a bagged tree, are called the **out-of-bag** (OOB) sample
- We can predict the Y_i using each of the trees in which unit i was OOB
- This yields around B/3 predictions for the i-th unit
- To obtain a single prediction for the ith observation we **average** those predicted responses (quantitative), or take a **majority** vote (qualitative)

OOB error estimate

- Since an OOB prediction can be computed for all n units, we can compute **an overall OOB MSE** or **classification error rate**

- It can be shown that **with B sufficiently large**, OOB error is virtually equivalent to leave-one-out cross-validation error

- **Price to pay** for bagging : interpretation

Random forests

From bagging to Random Forest

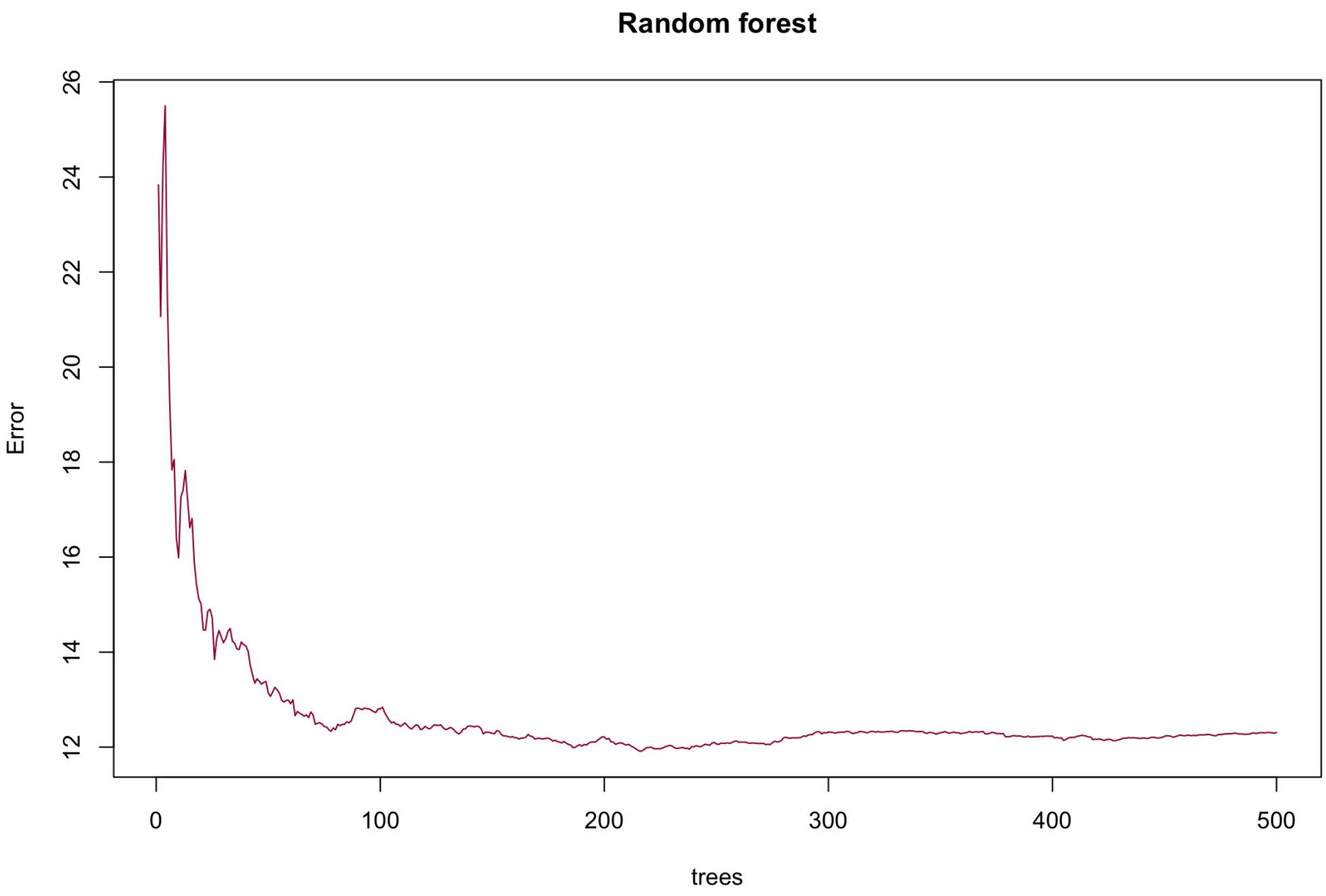
- The bagged trees based on the bootstrapped samples often look quite similar to each other. They are therefore often highly correlated

- Averaging uncorrelated trees can lead to a larger reduction in variance

- To de-correlate the trees, random forests build a number of trees on bootstrapped data using a **random sample of mtry predictors**

Random Forest

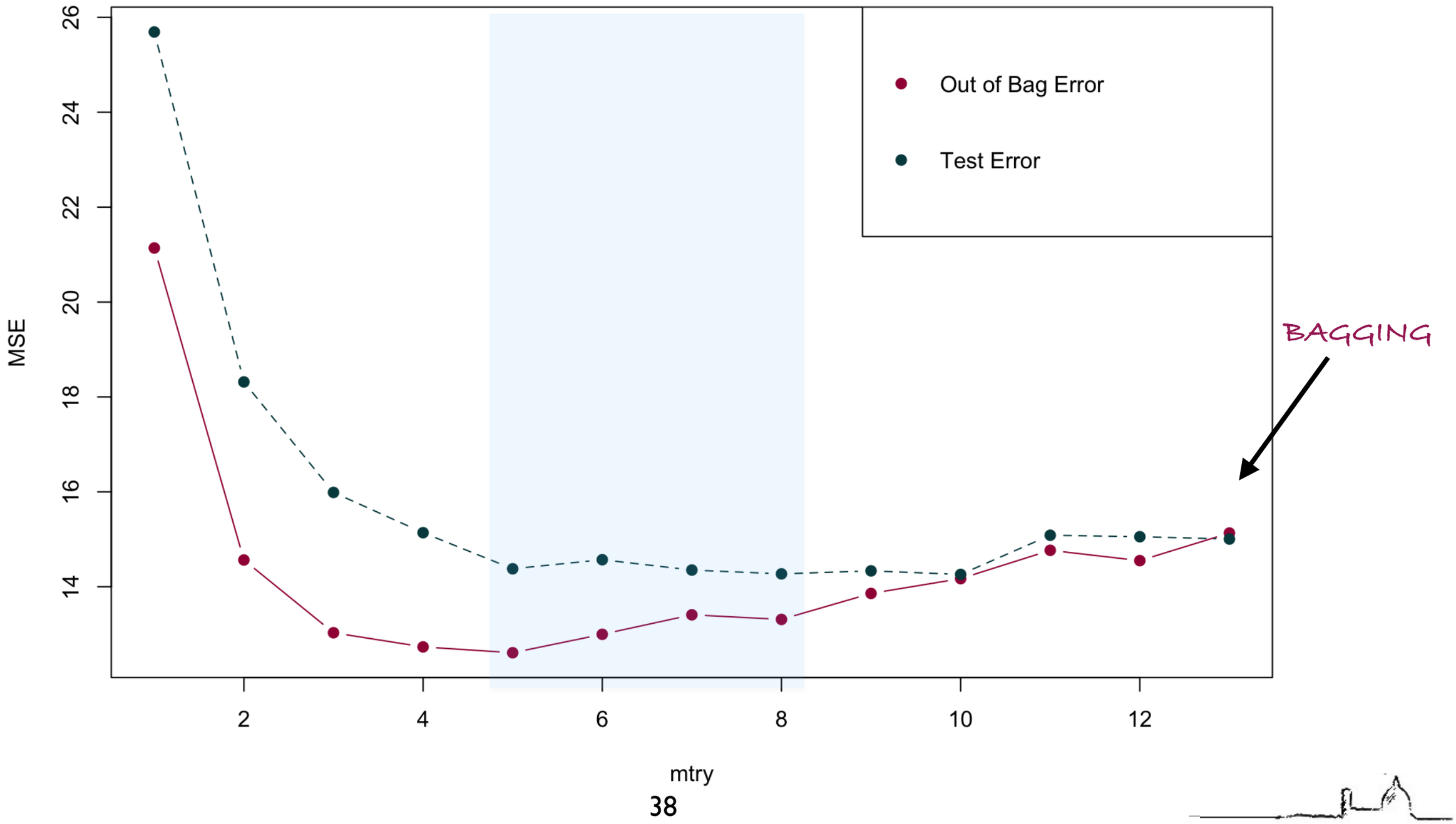
Two parameters to be tuned: (1) B = number of trees



Random Forest

Two parameters to be tuned: (2) mtry = n. variables sampled at each split

NB: It depends on the unknown number of good predictors



Conditional inference trees and forests

Conditional inference trees

- In conditional inference trees (CTREE), we perform a Fisher permutation test for independence between the response and each predictor

- A split is possible only if the p-value (adjusted for multiple comparisons) is smaller than a pre-specified nominal level

- No need to prune the tree!

(1) Perform all the independence tests

(2) Choose the variable with lowest p-value and split maximising the contrast

(3) Stop when no adjusted p-values are below the threshold

BART

BART = Bayesian Additive Regression Trees

$$\mathbb{E}[Y \mid \boldsymbol{X} = x] = \sum_{t=1}^{\tau} T(\boldsymbol{X}; \mathcal{R}_t, \boldsymbol{\gamma}_t) = \sum_{t=1}^{\tau} \sum_{m=1}^{M_t} \mu_{mt} \mathbb{I}_{\{x \in R_{mt}\}}$$

Bayesian “almost" nonparametric

Sum of tree model, no bagging, no variable sampling, no pruning

The trees are grown via MCMC and regularised by ad hoc priors

Each tree is evaluated in its entirety via the leaves parameters

Very good performance

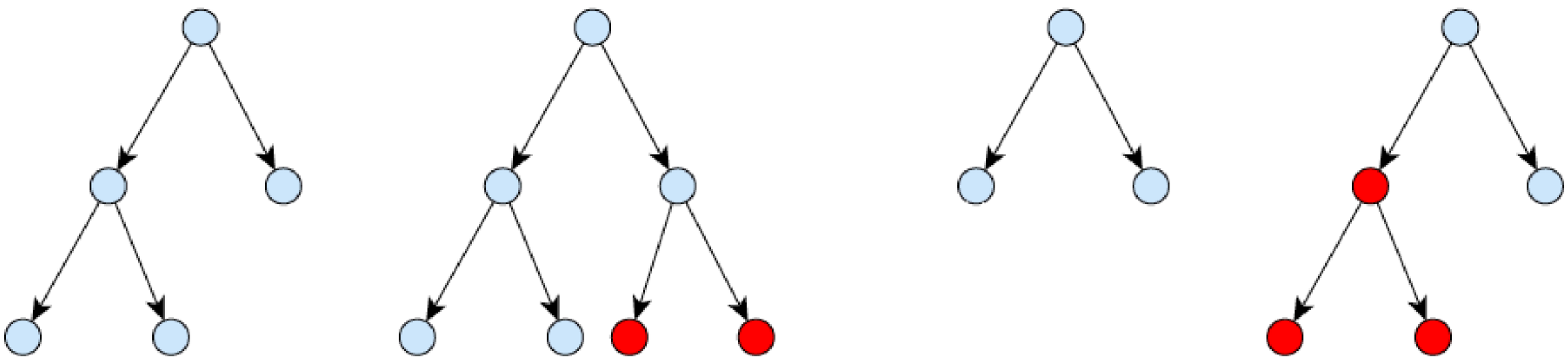
BART = Bayesian Additive Regression Trees

$$\mathbb{E}[Y \mid \boldsymbol{X} = x] = \sum_{t=1}^{\tau} T(\boldsymbol{X}; \mathcal{R}_t, \boldsymbol{\gamma}_t) = \sum_{t=1}^{\tau} \sum_{m=1}^{M_t} \mu_{mt} \mathbb{I}_{\{x \in R_{mt}\}}$$

No greedy search, but Backfitting MCMC algorithm

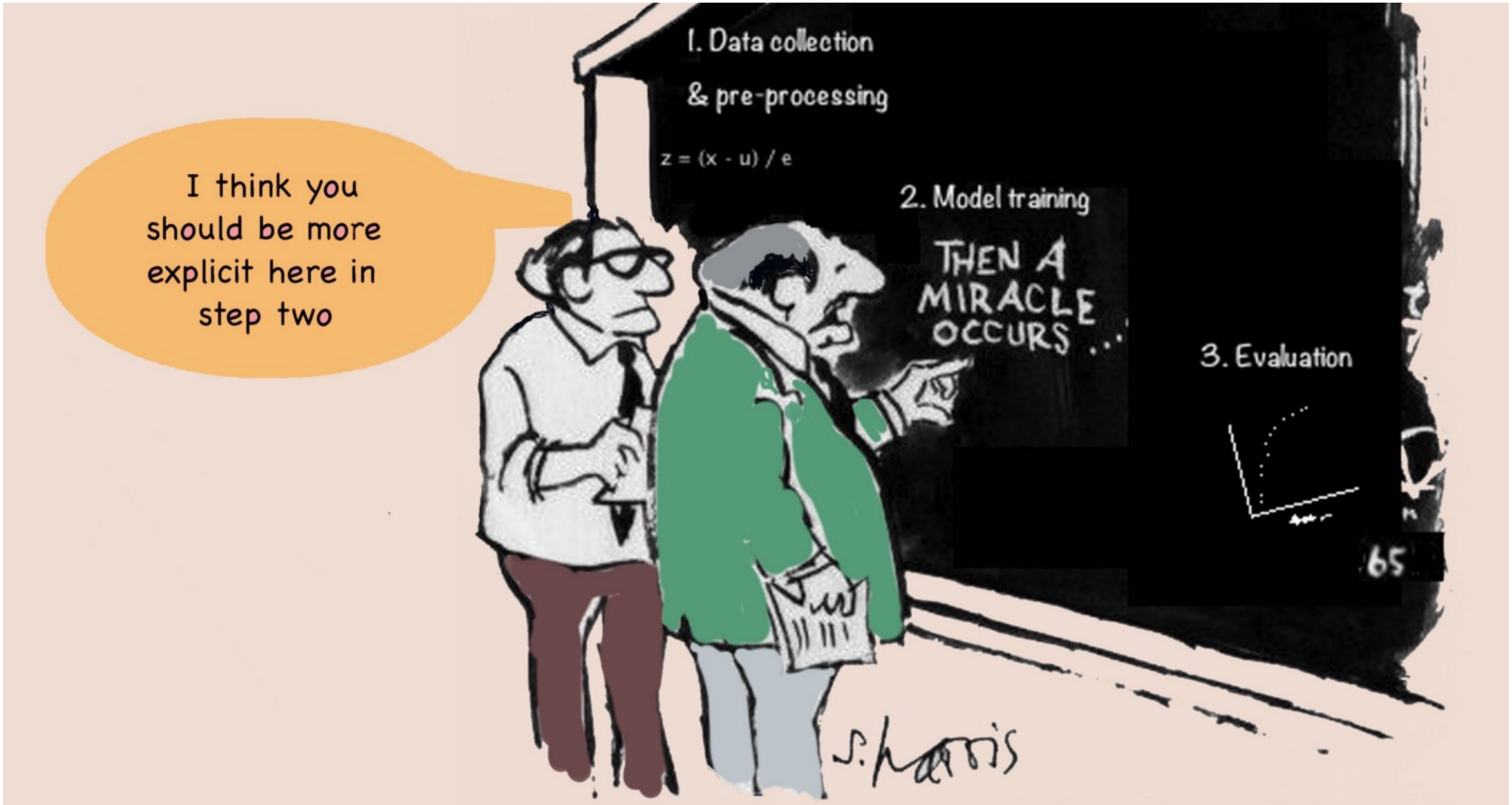
At each step, we sample from the full-conditional using the residuals given the other trees

A move in the tree structure consists of **Growing**, **Pruning**, **Changing**



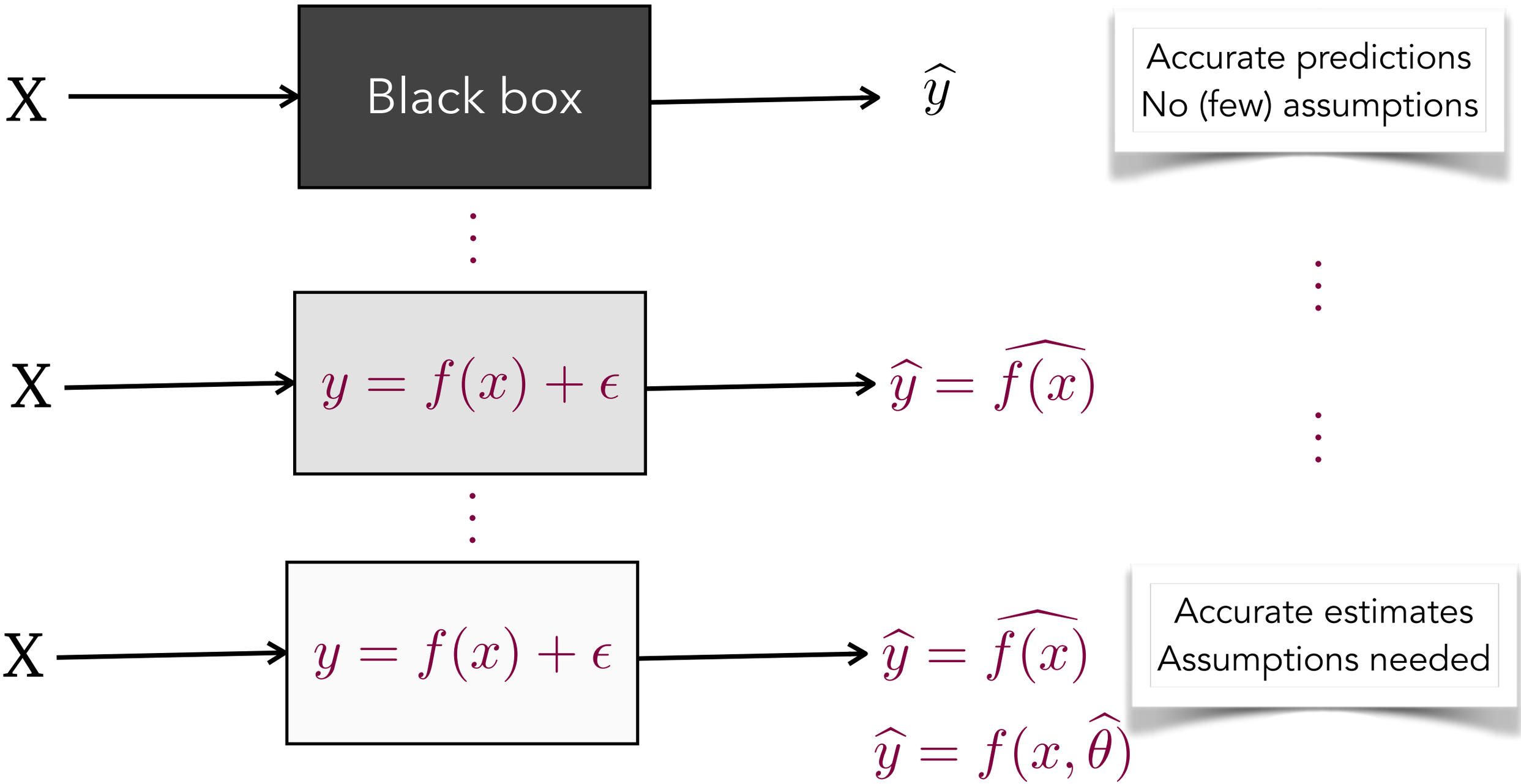
On the interpretation

Interpreting and understanding

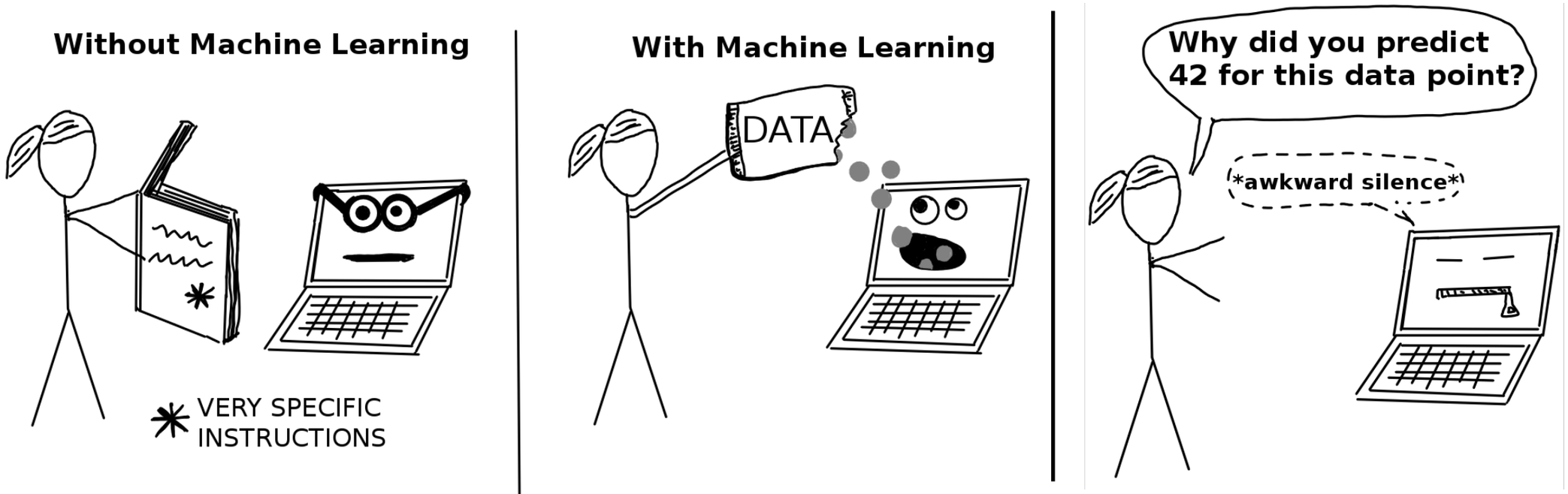


from *What's so Funny about Science?* by Sidney Harris (1977)

Interpreting tree-based models : longing transparency



… Using machine learning for making decisions…



- Out of the Artificial Intelligence/technological framework …

- Sometimes accurate predictions are not enough for making <u>good decisions</u>

- To understand whether the algorithm is working in a sensible way, the black box has to be whitened

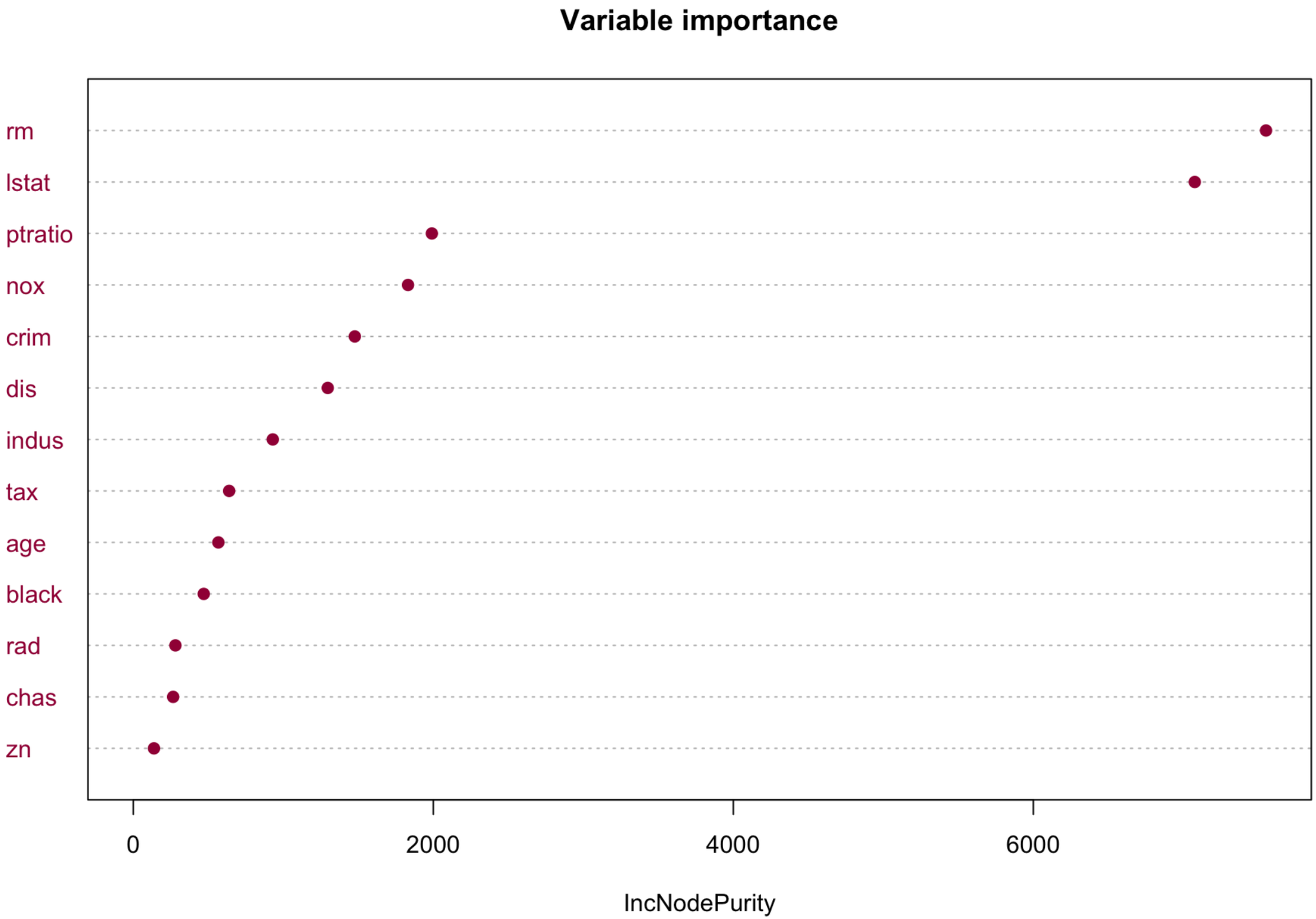
- It is not a matter of exactly understanding every bit and bytes of the model for all data points

- It is a matter of exactly understanding what drives the prediction, which are the discriminative predictors … are they **reasonable**?

How to interpret Trees and Forests?

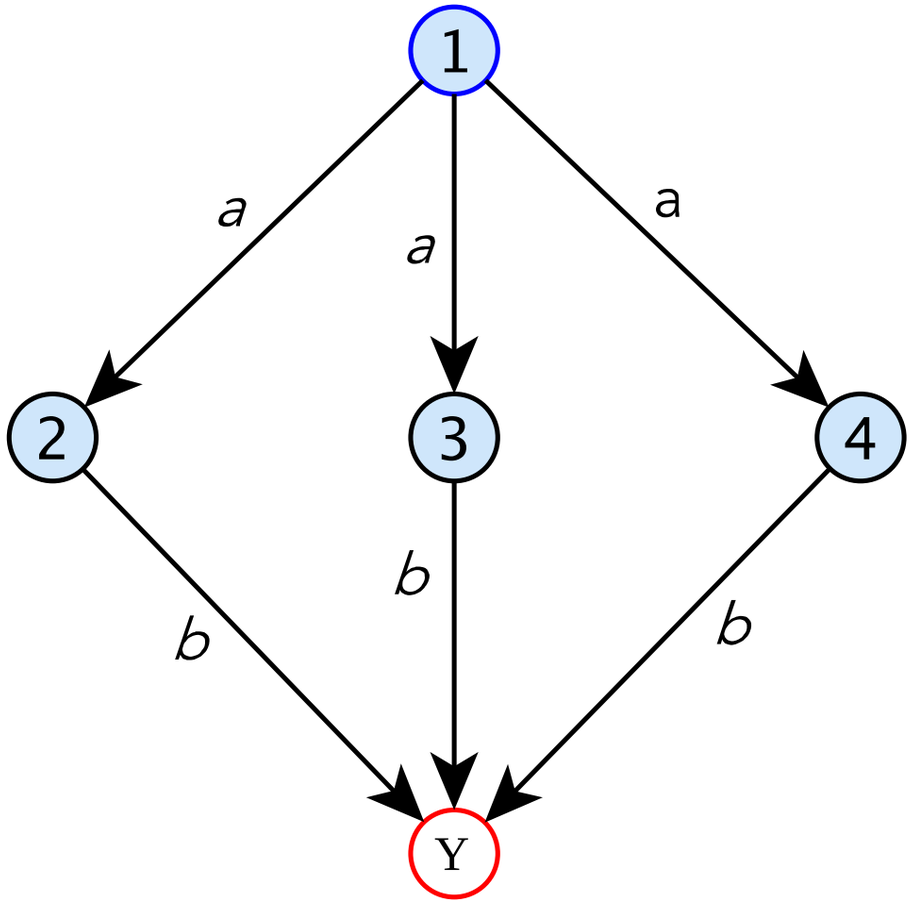
- One tree -> the **tree plot** makes clear how predictions have been made
- Forests are less transparent -> Variable importance measures
- Variable importance is a measure of the **importance of each variable in predicting the response**
- Several way to compute variable importance: the best is based on permutation of the variable -> Gain in prediction
- **Importance in predicting is not importance in explaining or causing**

How to interpret Variable importance?



Tricky example

- Variable importance in predicting sometimes deviates from true causal mechanism/direct association



- Think about Personalised medicine

- Or algorithms for banks to give a loan

- It is not just a matter of “to know how” but also of “Is it **fair**?”

Generative/explanatory vs Predictive models

- Statistical/Machine learning is focused on predicting

- Computer science, text or image processing rely on predictive modelling: the focus is on new/future observation

- Human sciences usually require generative/explanatory modelling: observed data are used to assess causal/explanatory hypotheses

- **Predicting is different from explaining**

- Lack of understanding in many disciplines of this distinction

THE TIMES
SATURDAY JUNE 9 2018

Chinese pigs help predict earthquakes

Calum MacLeod

July 7 2015, 1:01am, The Times

Thanks for your attention!

Some references

- Berk, R. A. (2008). Statistical learning from a regression perspective (2nd edition). New York: Springer.

- Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1984). *Classification and Regression Trees*. Chapman & Hall/CRC.

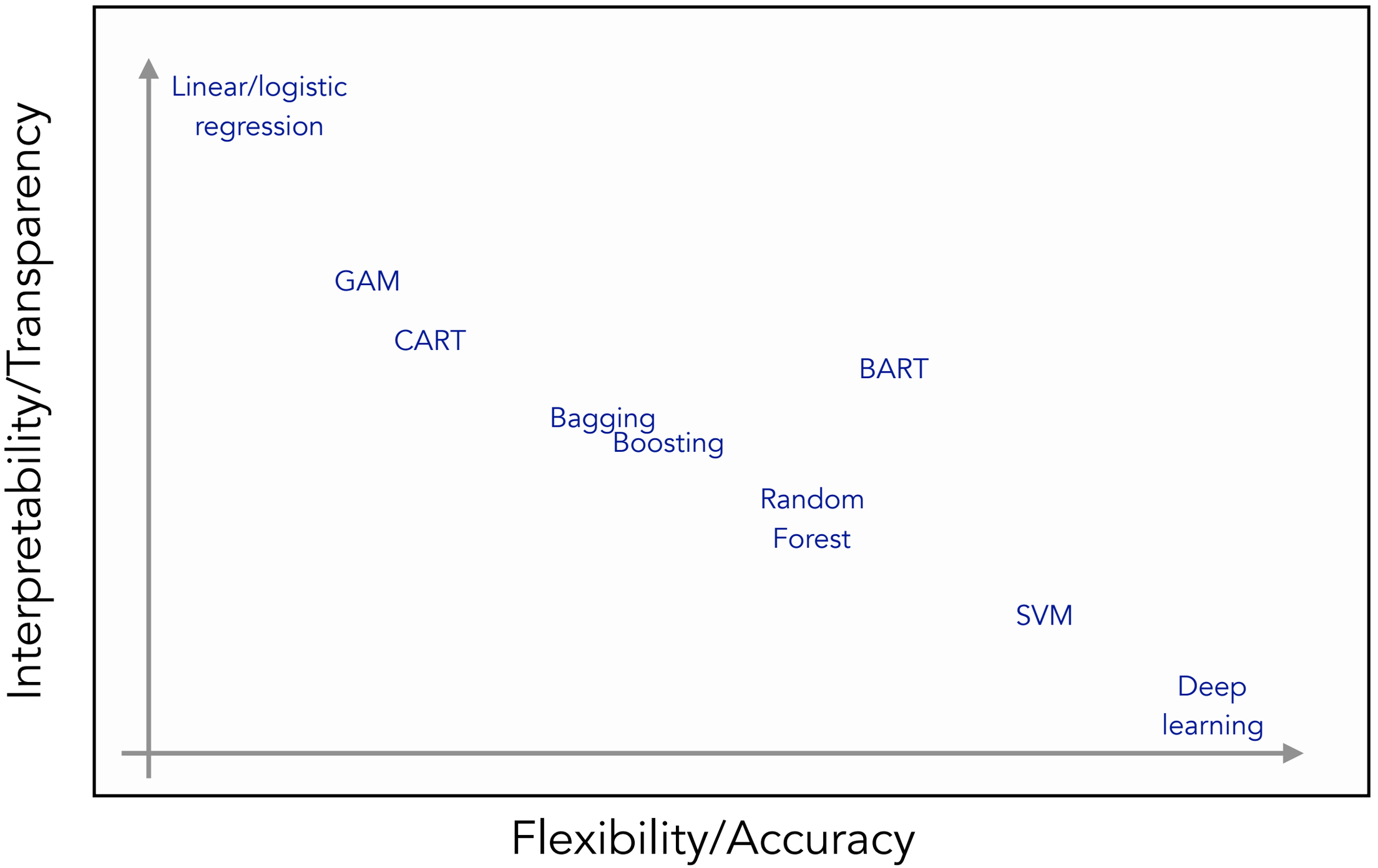
- Breiman, L. (2001). Random forests. Machine Learning, 45:5–32.

- Chipman, H. A., George, E. I., and McCulloch, R. E. (2010). BART: Bayesian additive regression trees. *Annals of Applied Statistics*, 4:266–298.

- Friedman, J., Hastie, T., & Tibshirani, R. (2001). The elements of statistical learning (Vol. 1, No. 10). New York: Springer series in statistics.

- Hothorn, T., Hornik, K., and Zeileis, A. (2006). Unbiased recursive partitioning: a conditional inference framework. Journal of Computational and Graphical Statistics, 15:651–674

Flexibility - interpretability trade-off



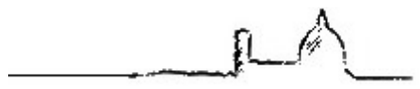
Cross-validation

Training Error versus Test error

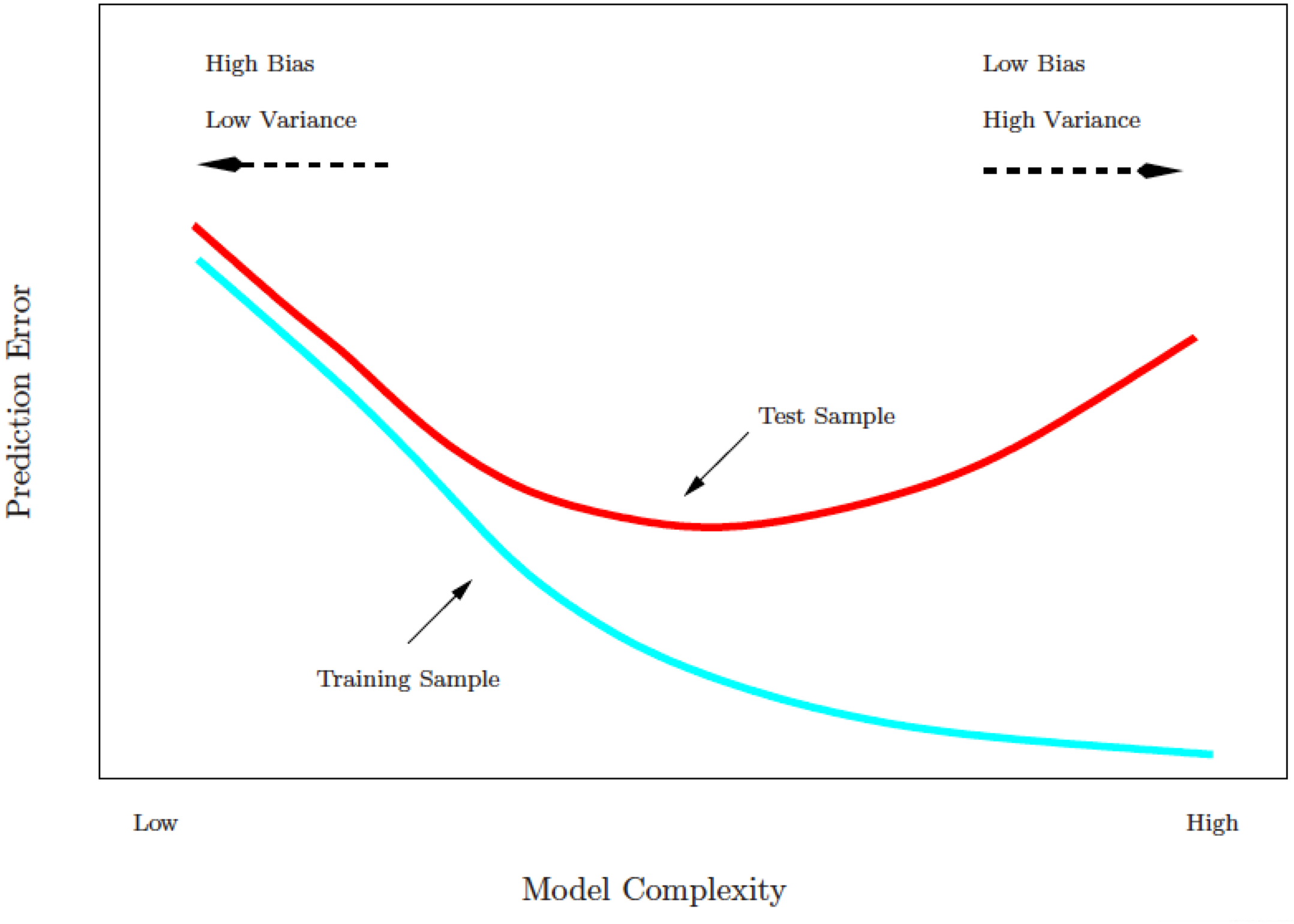
- **Training error** : it is the value of the loss measure computed on the training data

- **Test error** : it is the value of the loss measure computed predicting the statistical learning method on the test data

- The training and the test errors can be quietly different, and can vary a lot among different partitions of the data



Training- versus Test-Set Performance



K-fold Cross-validation

K=5