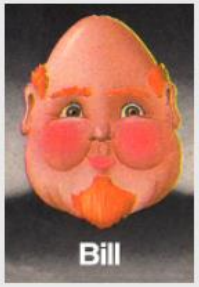
Random Forest

A. Fornaser – alberto.fornaser@unitn.it

Sources

- Lecture 15: decision trees, information theory and random forests, Dr. Richard E. Turner
- Trees and Random Forests, Adele Cutler, Utah State University
- Random Forests for Regression and Classification, Adele Cutler, Utah State University

Guess who?

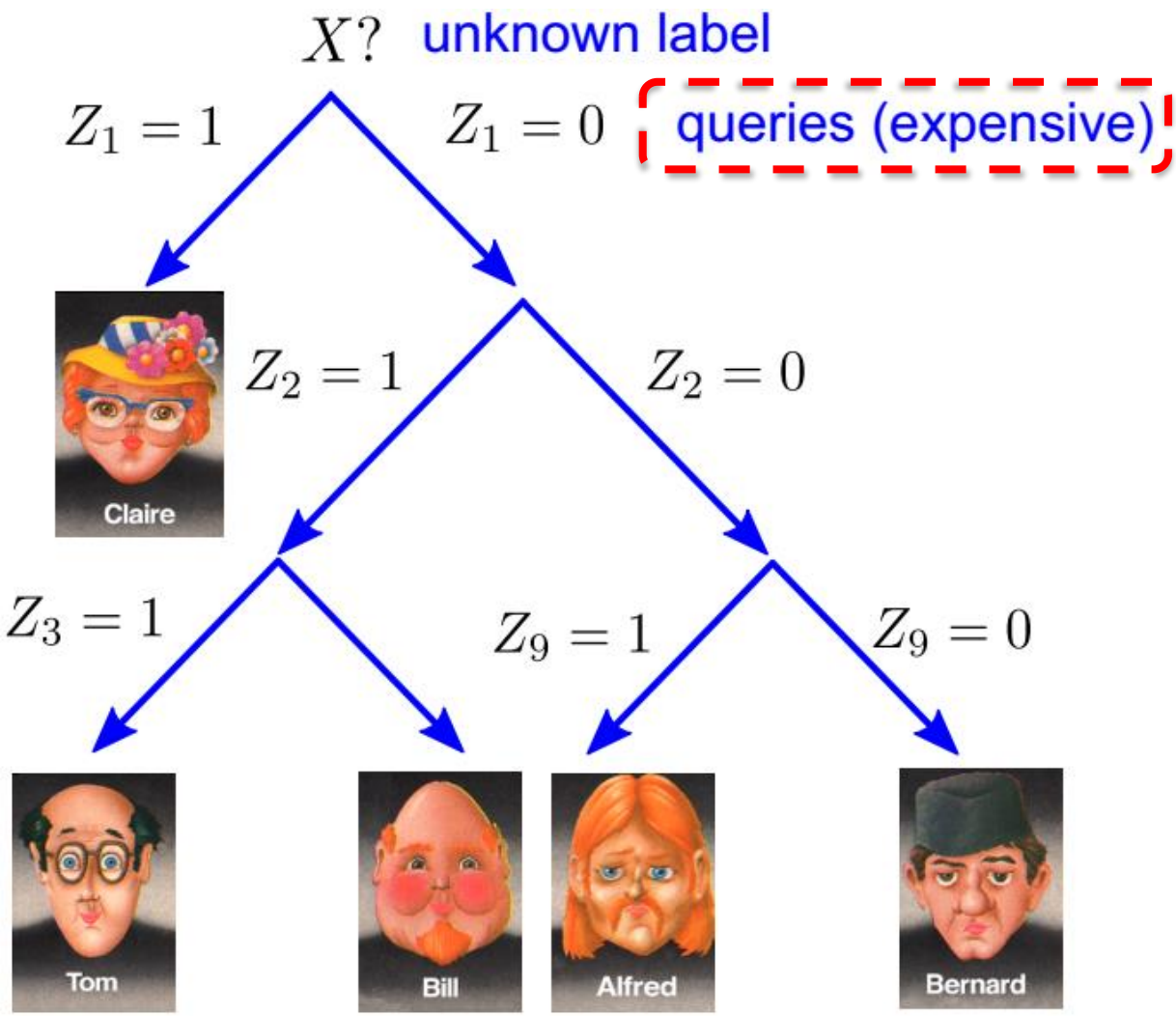


feature vector

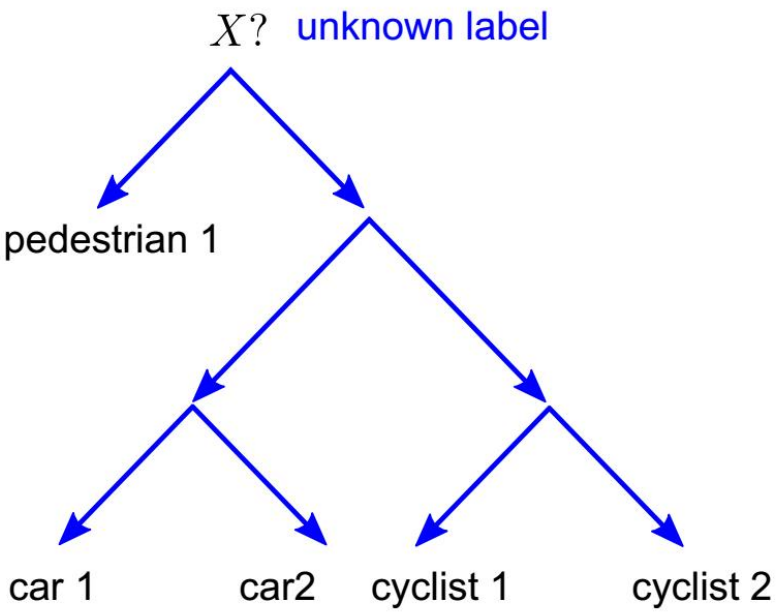
$$Z = \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \end{bmatrix} \begin{array}{l} \leftarrow \text{female} \\ \leftarrow \text{bald} \\ \leftarrow \text{brown eyes} \\ \\ \end{array}$$

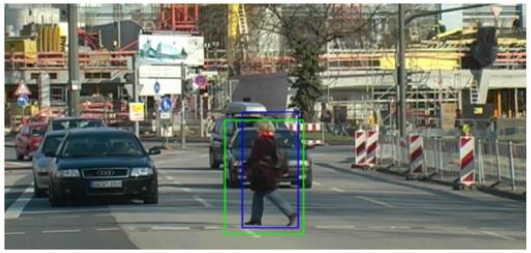
class label

$$X = 12$$


X? unknown label
queries (expensive)
$Z_1 = 1$
$Z_1 = 0$
$Z_2 = 1$
$Z_2 = 0$
$Z_3 = 1$
$Z_9 = 1$
$Z_9 = 0$
Claire
Tom
Bill
Alfred
Bernard

Motivation: object detection





feature vector

$$Z = \begin{bmatrix} 0.9 \\ 0.3 \\ 0.2 \\ \vdots \end{bmatrix}$$

patch of pixels

class label

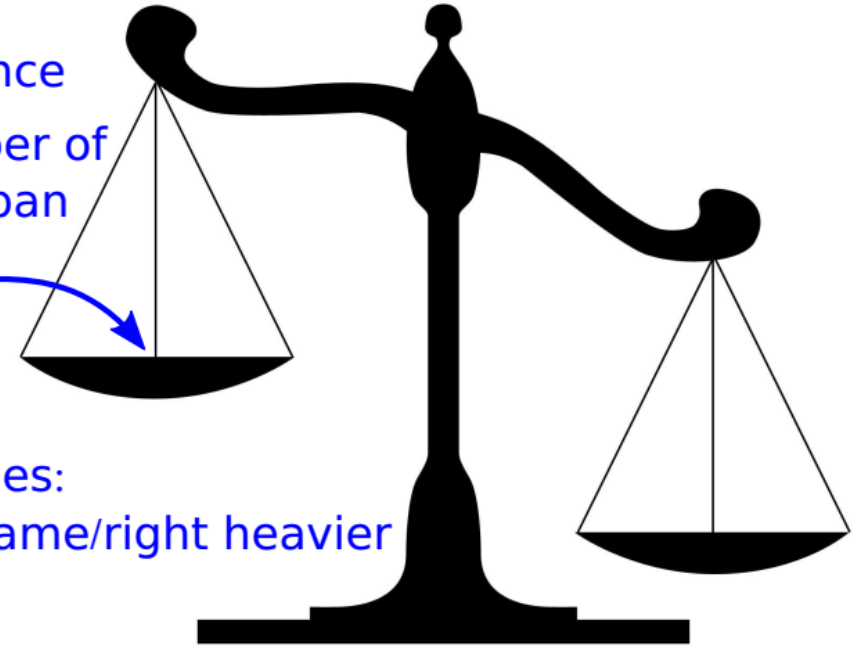
$$X = 12$$

How do we automatically learn optimal queries to make at each node?
How do we minimise the expected number of queries?
How do we make learning/estimation efficient?
How do we handle continuous features/distributional outputs?

TASK: find the odd ball **and** whether it's heavier or lighter in as few a weighings as possible

two pan balance

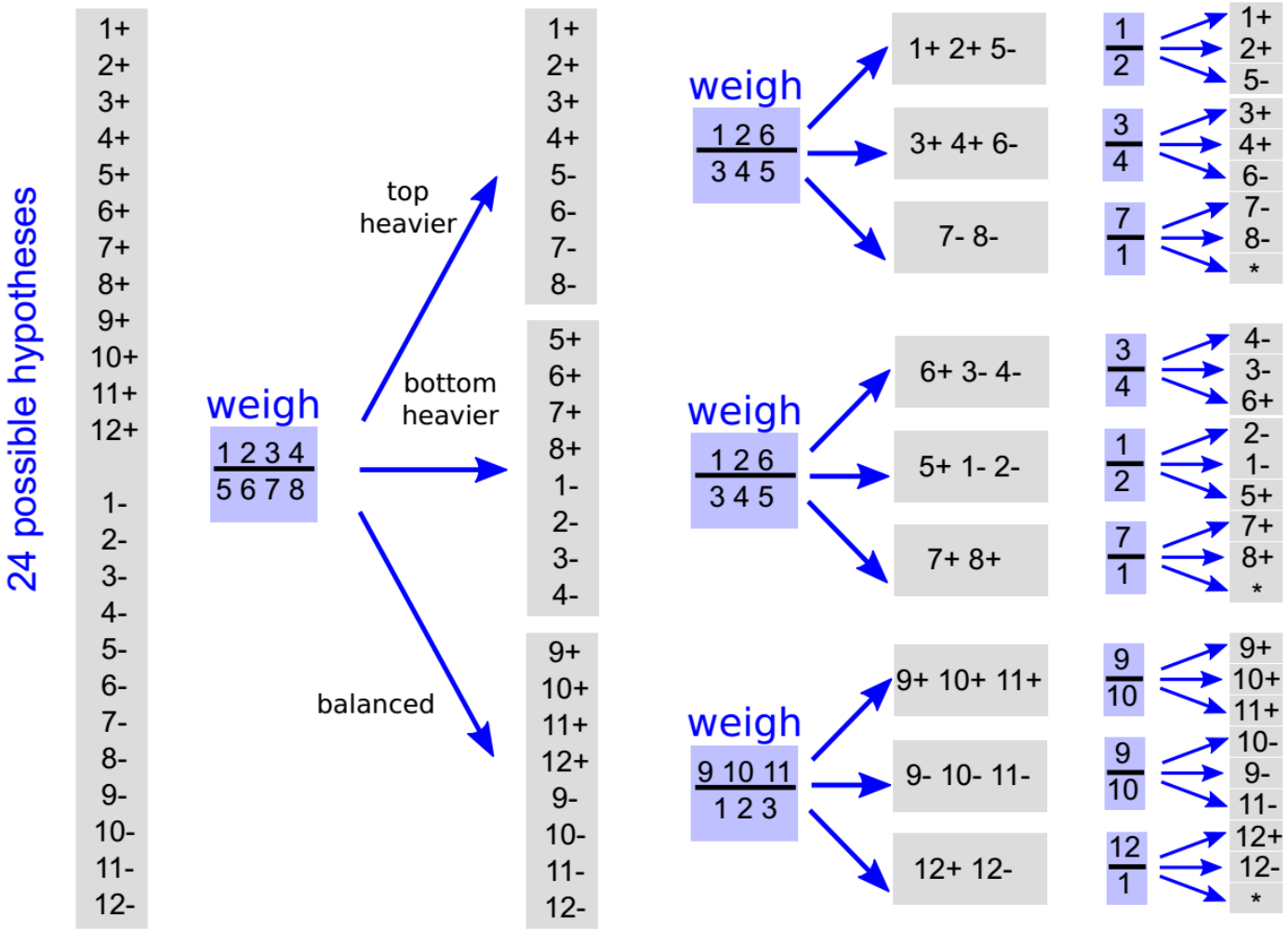
put any number of balls in each pan



three outcomes:
left heavier/same/right heavier

12 balls, all but one same weight, **one is heavier or lighter**

Decision tree



Classification and Regression Trees

Pioneers:

- Morgan and Sonquist (1963).
- Breiman, Friedman, Olshen, Stone (1984).
- Quinlan (1993).

- **Tree-based methods are simple and useful for interpretation.**
- However they typically are not competitive with the best supervised learning approaches in terms of prediction accuracy.
- Hence we also discuss *bagging*, *random forests*, and *boosting*. These methods grow multiple trees which are then combined to yield a single consensus prediction.
- **Combining a large number of trees can often result in dramatic improvements in prediction accuracy, at the expense of some loss interpretation.**

Bagging

Bootstrap aggregation is a general-purpose procedure for **reducing the variance of a statistical learning method**; it is particularly useful and frequently used in the context of decision trees.

Averaging a set of observations reduces variance → this is not practical because we generally do not have access to multiple training sets.

Instead, we can **bootstrap, by taking repeated samples from the (single) training data set**.
We generate B different bootstrapped training data sets, then train our method on the bth bootstrapped training set in order to get f(x), the prediction at a point x.

Then average all the predictions to obtain:

$$\hat{f}_{\text{bag}}(x) = \frac{1}{B}\sum_{b=1}^{B}\hat{f}^{*b}(x).$$

This is called bagging.

Decision trees

“Decision trees are the individual learners that are combined”.

Decision trees, one of most popular learning methods commonly used for data exploration.

One type of decision tree is called:
CART → Classification And Regression Tree *(Breiman 1983)*
CART: **greedy, top-down binary, recursive partitioning** that divides feature space into **sets of disjoint rectangular regions**.

- Regions should be pure with respect to response variable.
- Simple model is fit in each region
  - **constant value for regression**
  - **majority vote for classification**

Regression

Given predictor variables x, and a **continuous** response variable y, build a model for:

- Predicting the value of y for a new value of x
- Understanding the relationship between x and y

e.g. predict a person's systolic blood pressure based on their age, height, weight, etc.

Classification

Given predictor variables x, and a **categorical** response variable y, build a model for:

- Predicting the value of y for a new value of x
- Understanding the relationship between x and y

e.g. predict a person's 5-year-survival (yes/no) based on their age, height, weight, etc.

Regression Methods

- Simple linear regression
- Multiple linear regression
- Nonlinear regression (parametric)
- Nonparametric regression:
  - Kernel smoothing, spline methods, wavelets
  - Trees (1984)
- <u>Machine learning methods:</u>
  - Bagging
  - **Random forests**
  - Boosting

Classification Methods

- Linear discriminant analysis (1930's)
- Logistic regression (1944)
- Nonparametric methods:
  - Nearest neighbor classifiers (1951)
  - Trees (1984)
- <u>Machine learning methods:</u>
  - Bagging
  - **Random forests**
  - Support vector machines

Classification and Regression Trees

- Grow a binary tree.
- At each node, "**split**" the data into two "daughter" nodes.
- Splits are chosen using a **splitting criterion**.
- Bottom nodes are "terminal" nodes.
- For **regression** the predicted value at a node is the ***average*** **response variable for all observations in the node**.
- For **classification** the predicted class is the ***most common*** ***class*** **in the node (majority vote).**
    - For classification trees, can also get estimated **probability of membership in each of the classes**

Splitting criteria

- **Regression**: residual sum of squares

$$\boxed{RSS = \sum_{left} (y_i - y_L^*)^2 + \sum_{right} (y_i - y_R^*)^2}$$

where y_L^* = mean y-value for left node

y_R^* = mean y-value for right node

- **Classification**: Gini criterion

$$\boxed{Gini = N_L \sum_{k=1,\dots,K} p_{kL} (1- p_{kL}) + N_R \sum_{k=1,\dots,K} p_{kR} (1- p_{kR})}$$

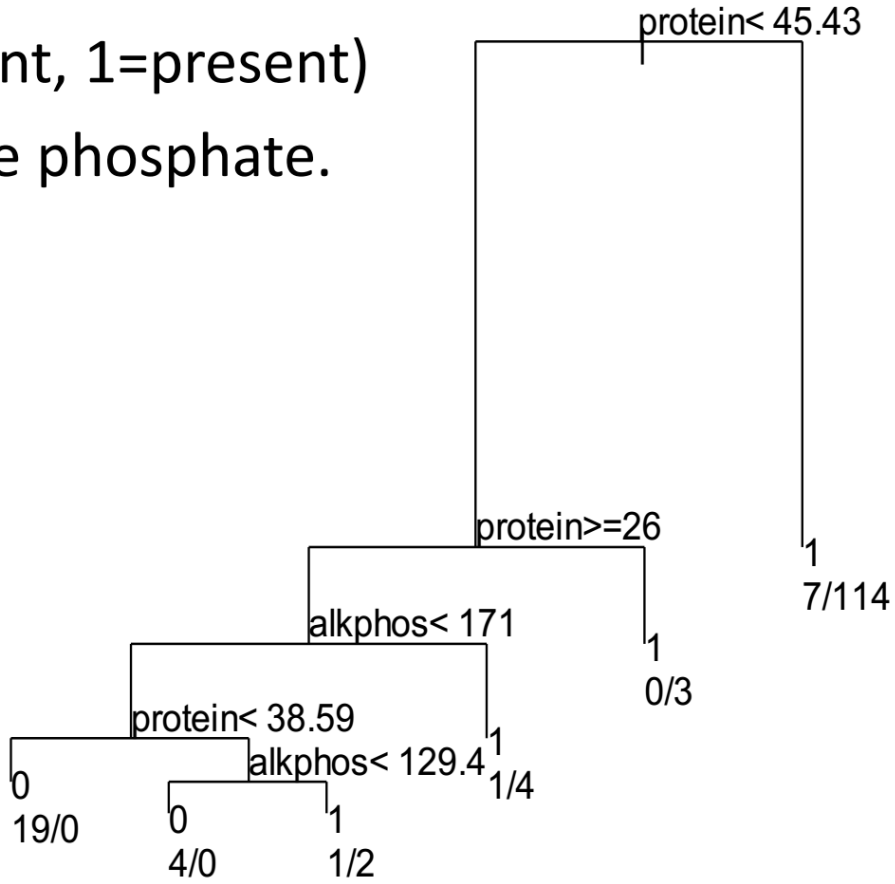
where p_{kL} = proportion of class k in left node

p_{kR} = proportion of class k in right node

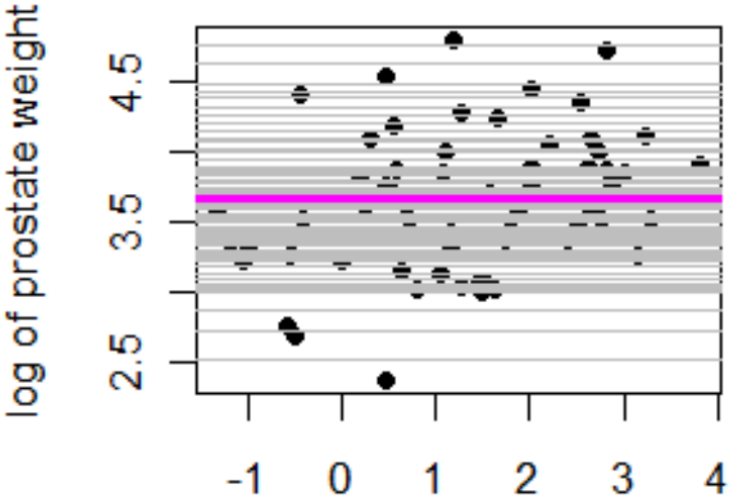
A Classification Tree

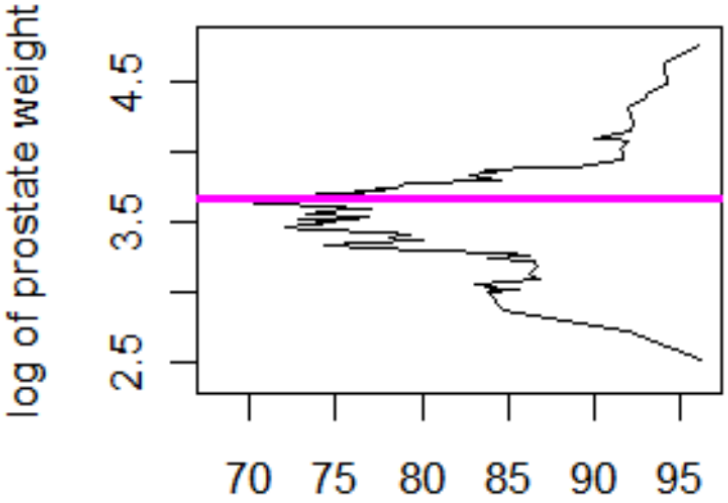
Predict hepatitis (0=absent, 1=present) using protein and alkaline phosphate.

"Yes" goes left.



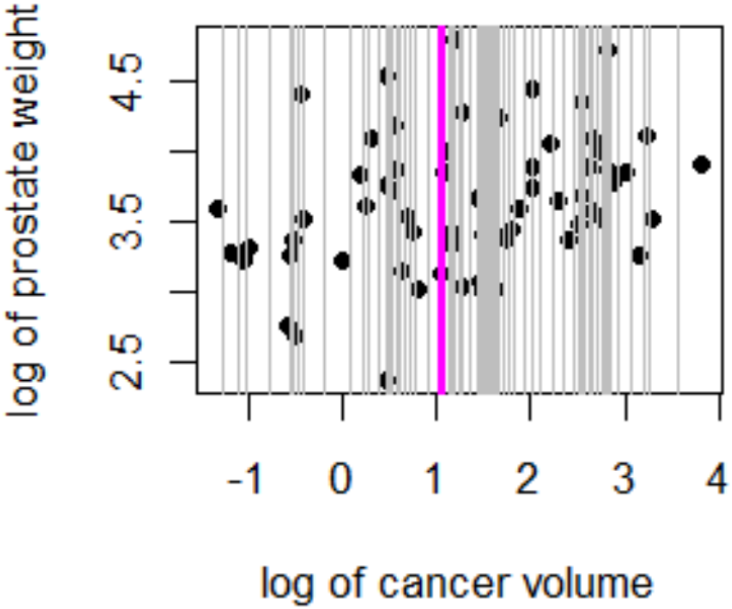
Choosing the best horizontal split

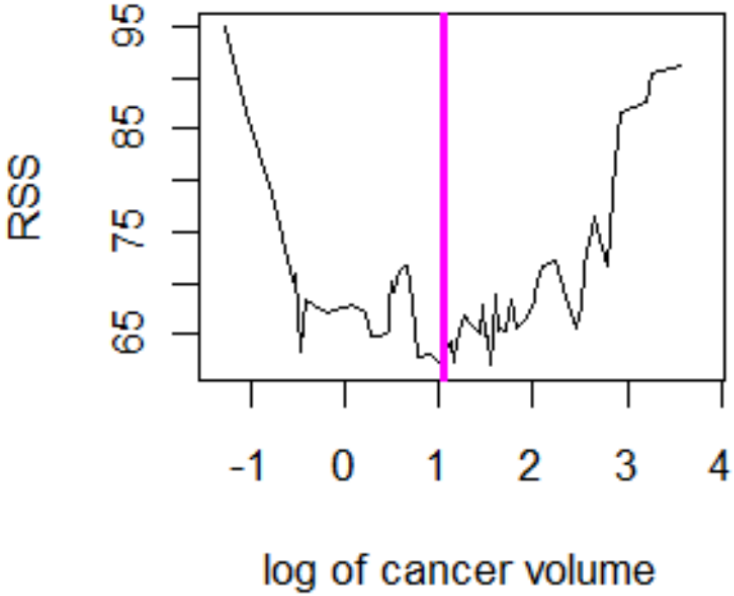




Best horizontal split is at 3.67 with RSS = 68.09.

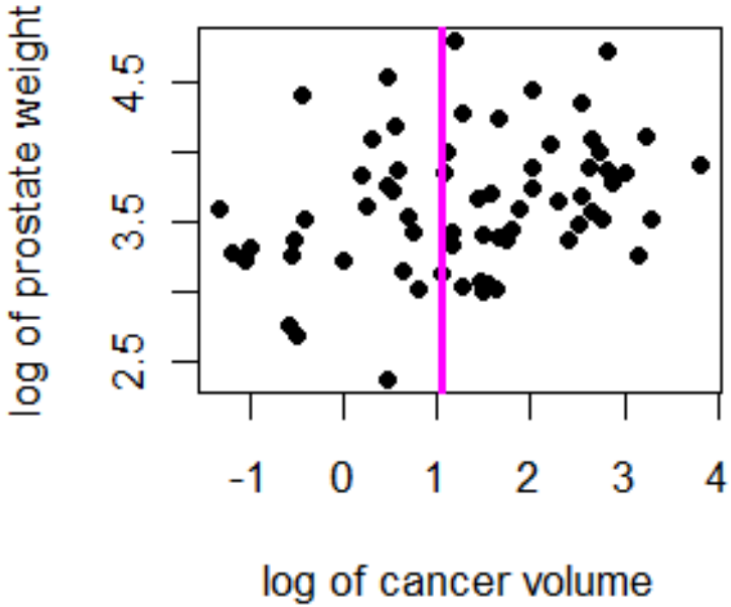
Choosing the best vertical split

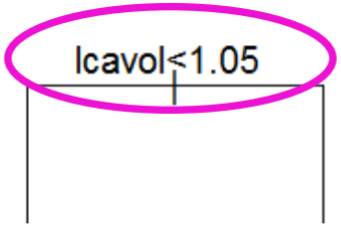




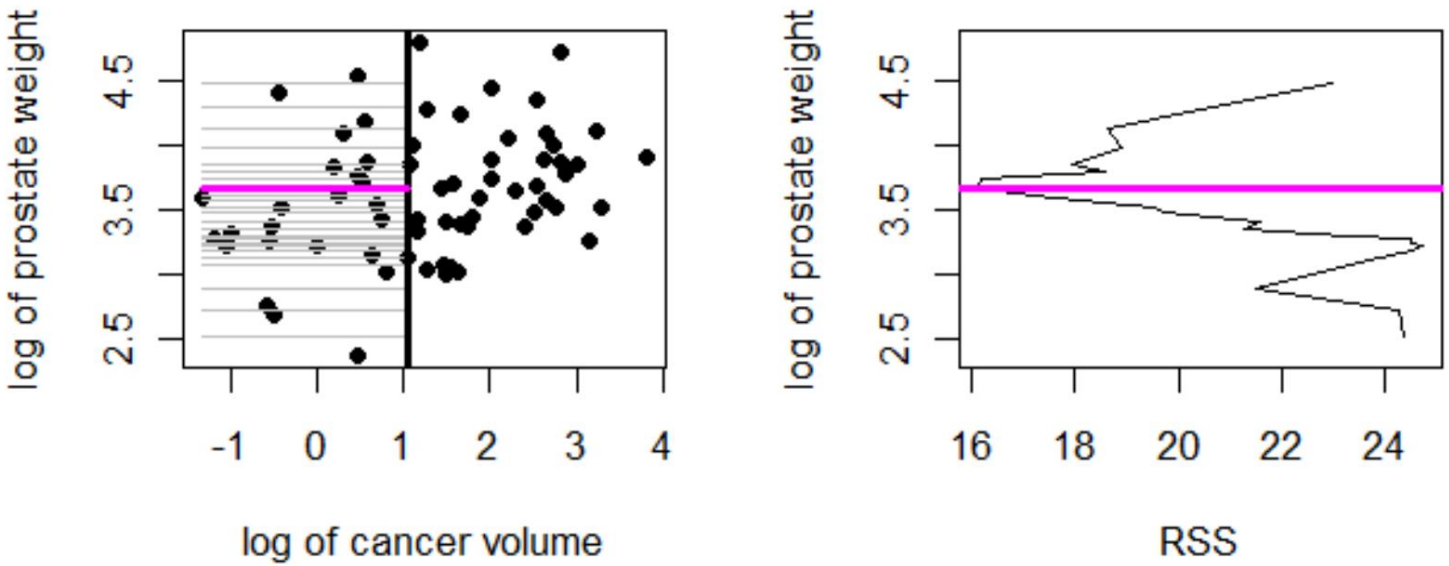
Best vertical split is at 1.05 with RSS = 61.76.

Regression tree (prostate cancer)



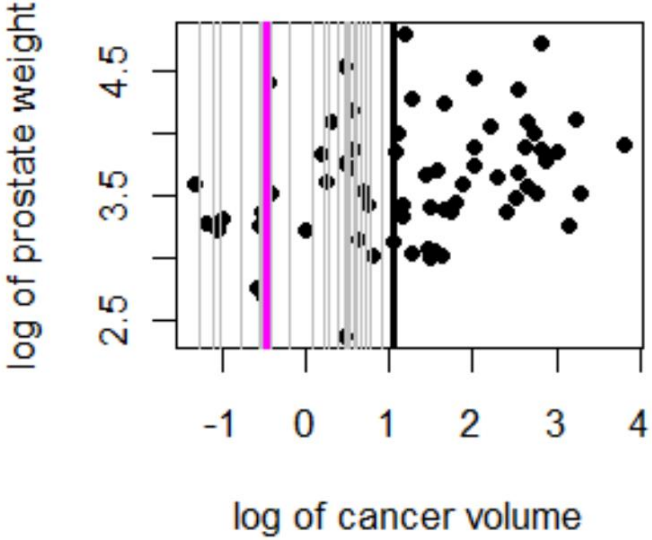


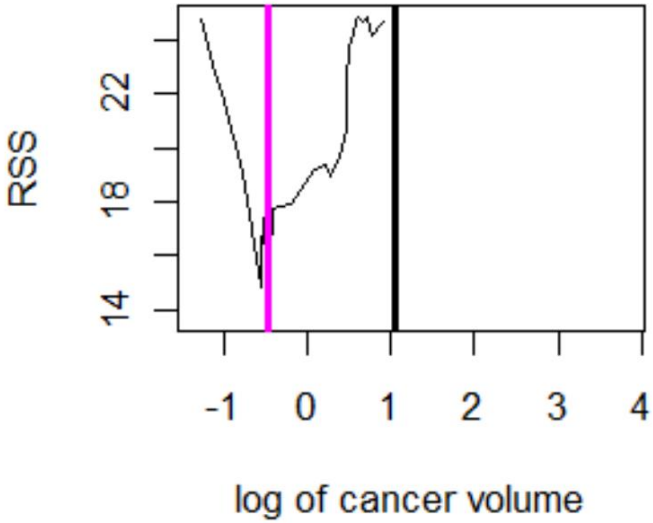
Choosing the best split in the left node



Best horizontal split is at 3.66 with RSS = 16.11.

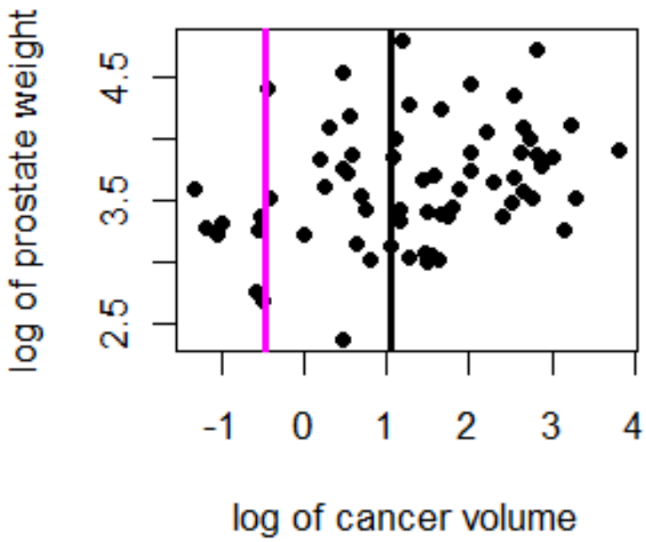
Choosing the best split in the left node

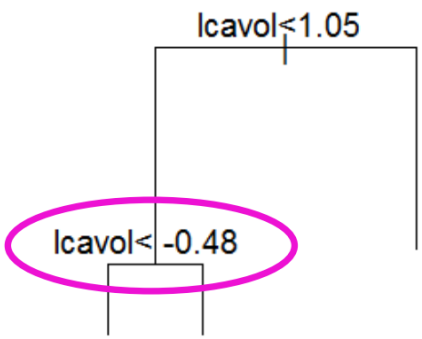




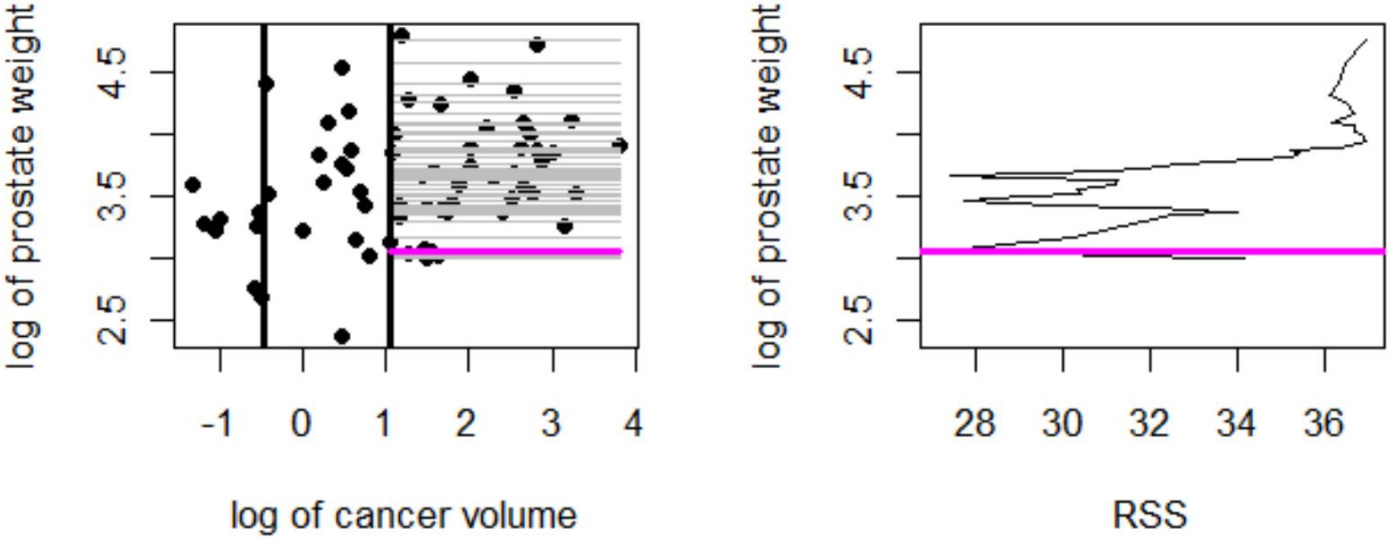
Best vertical split is at -.48 with RSS = 13.61.

Regression tree (prostate cancer)



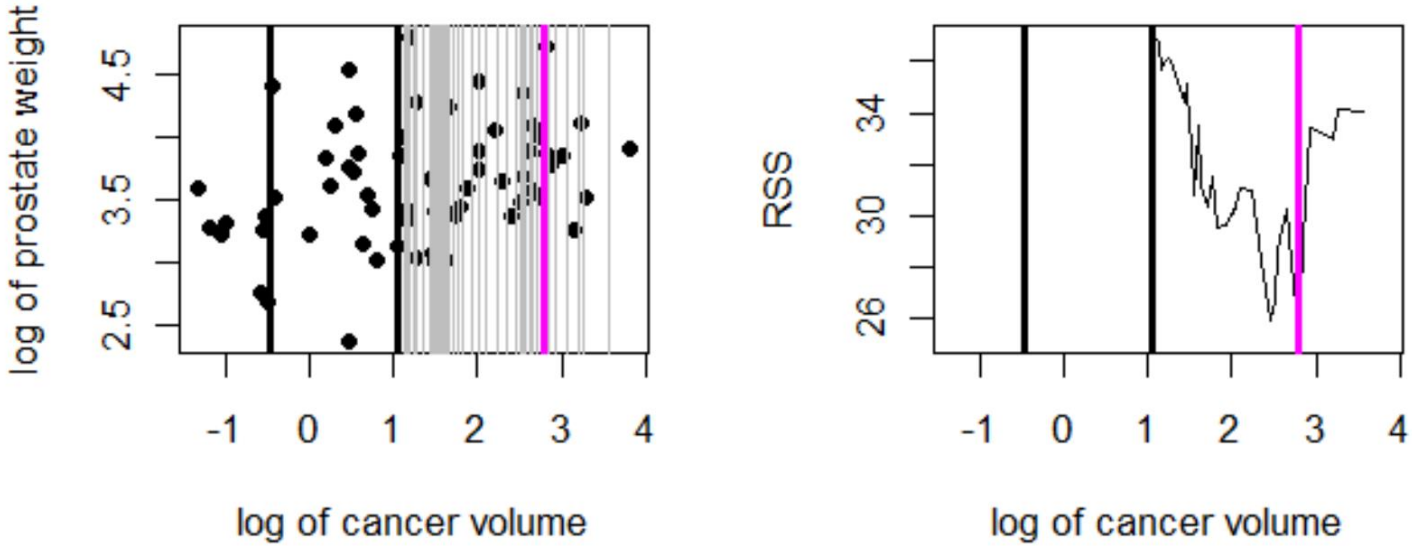


Choosing the best split in the right node



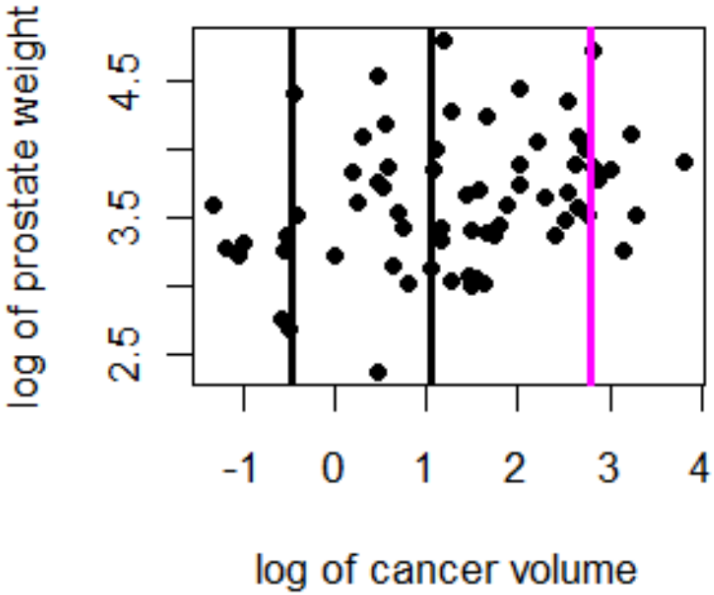
Best horizontal split is at 3.07 with RSS = 27.15.

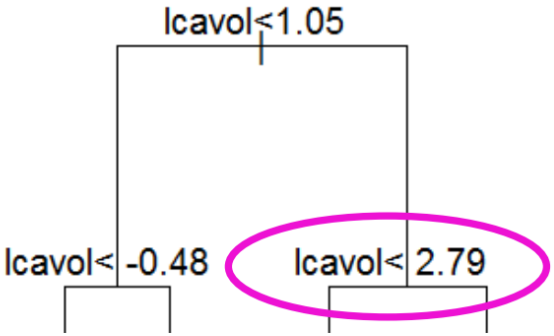
Choosing the best split in the right node



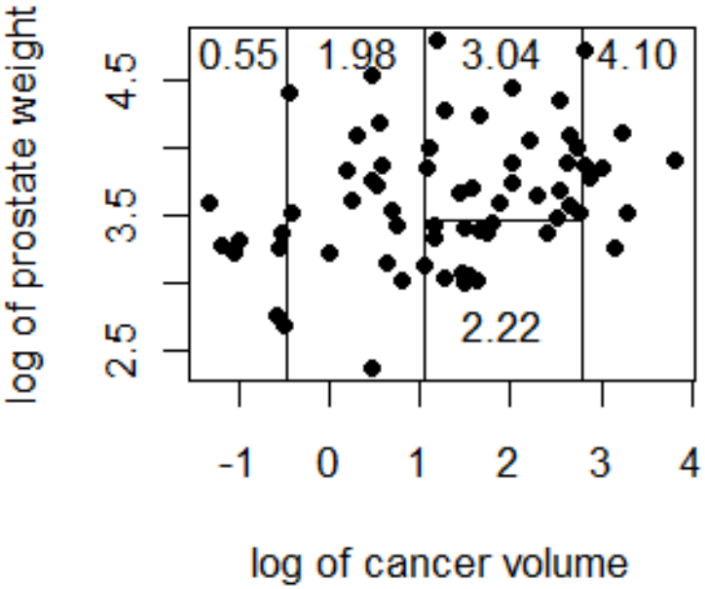
Best vertical split is at 2.79 with RSS = 25.11.

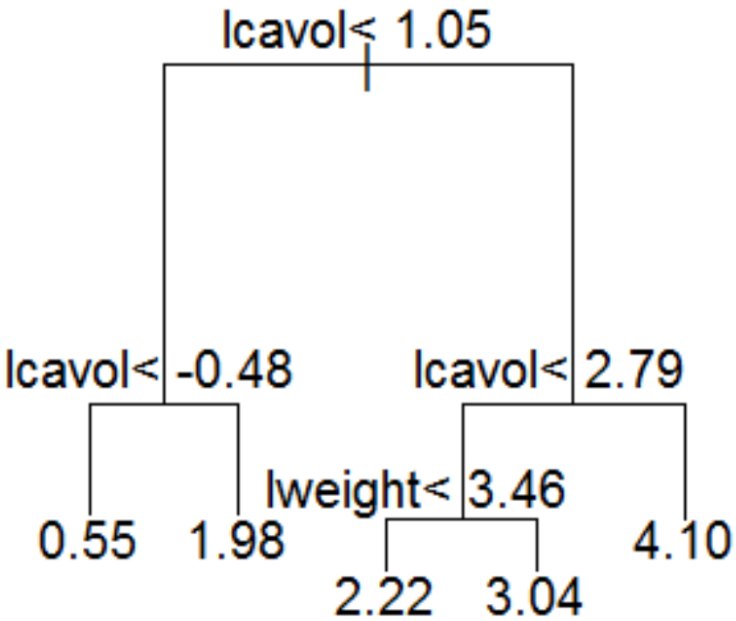
Regression tree (prostate cancer)



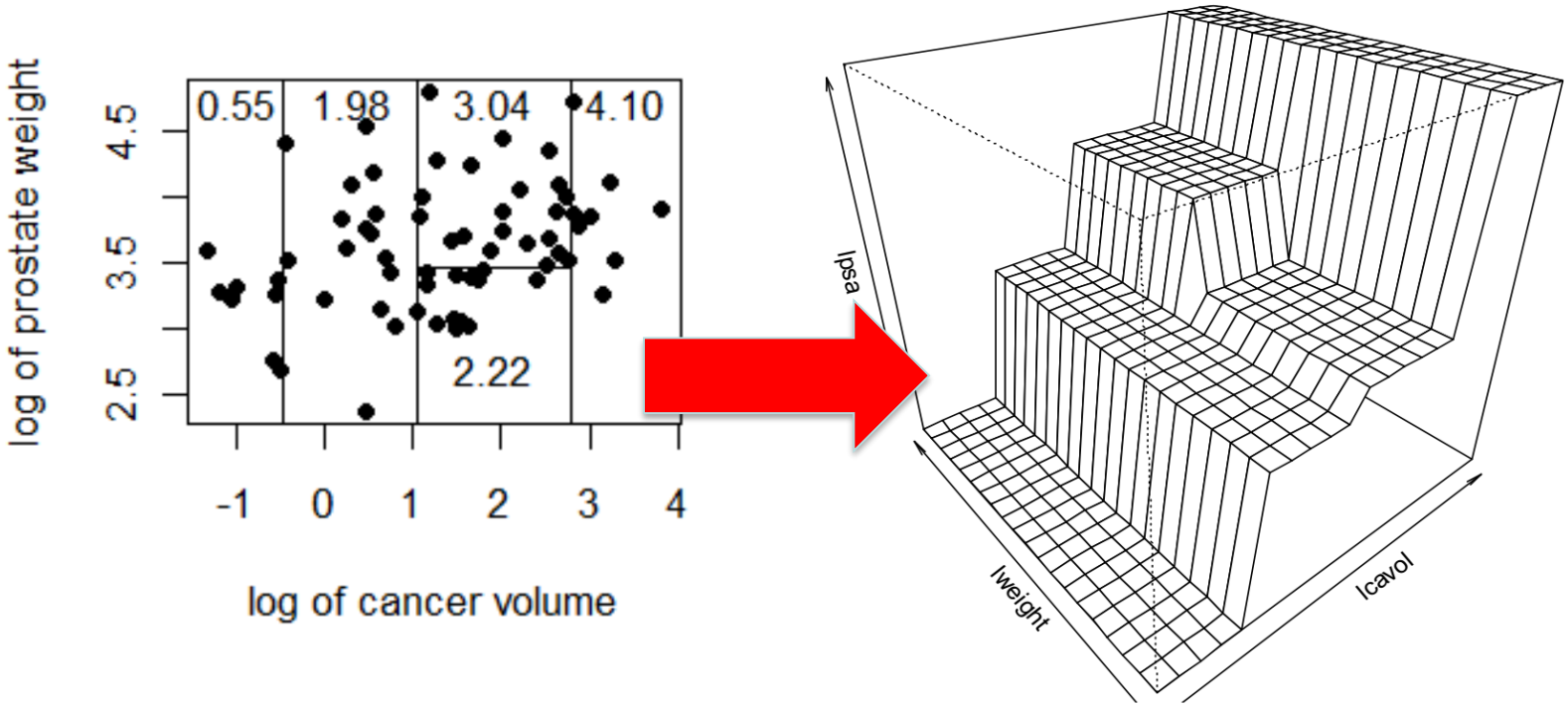


Regression tree (prostate cancer)



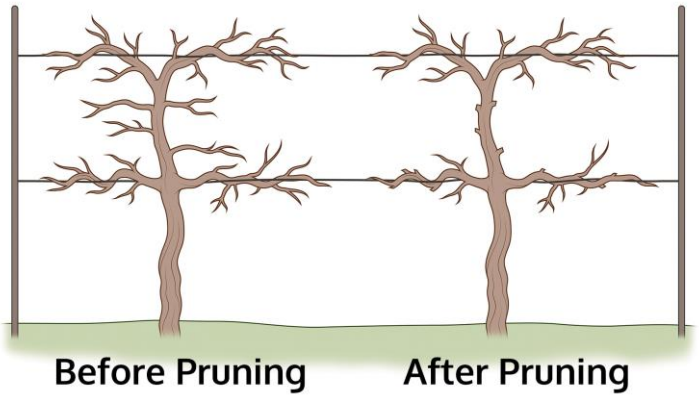


Regression tree (prostate cancer)



Pruning

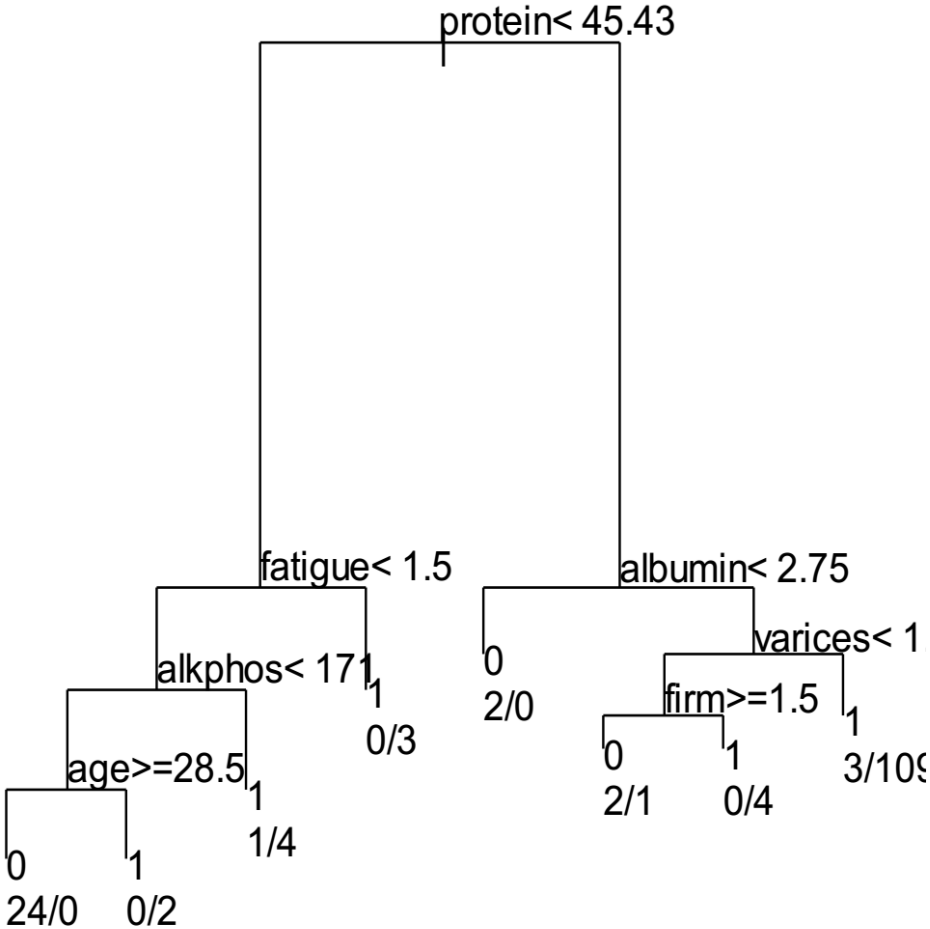
- If the tree is too big, **the lower "branches" are modeling noise in the data → "overfitting"**.
- The usual paradigm is to grow the trees large and "**prune**" back unnecessary splits.
- Methods for **pruning** trees have been developed. Most use some form of crossvalidation. Tuning may be necessary.



Classification and Regression Trees

Advantages (ctnd)

- The picture of the tree can give valuable insights into which variables are important and where.
- The terminal nodes suggest a natural clustering of data into homogeneous groups.



Classification and Regression Trees

Disadvantages

- *Accuracy* - current methods, such as support vector machines and ensemble classifiers often have 30% lower error rates than CART.
- *Instability* – if we change the data a little, the tree picture can change a lot. So the interpretation is not as straightforward as it appears.

Today, we can do better!

Random Forests

Random Forest

"Learning ensemble consisting of a bagging of ***un-pruned decision tree learners*** *with a* ***randomized selection of features at each split****."*

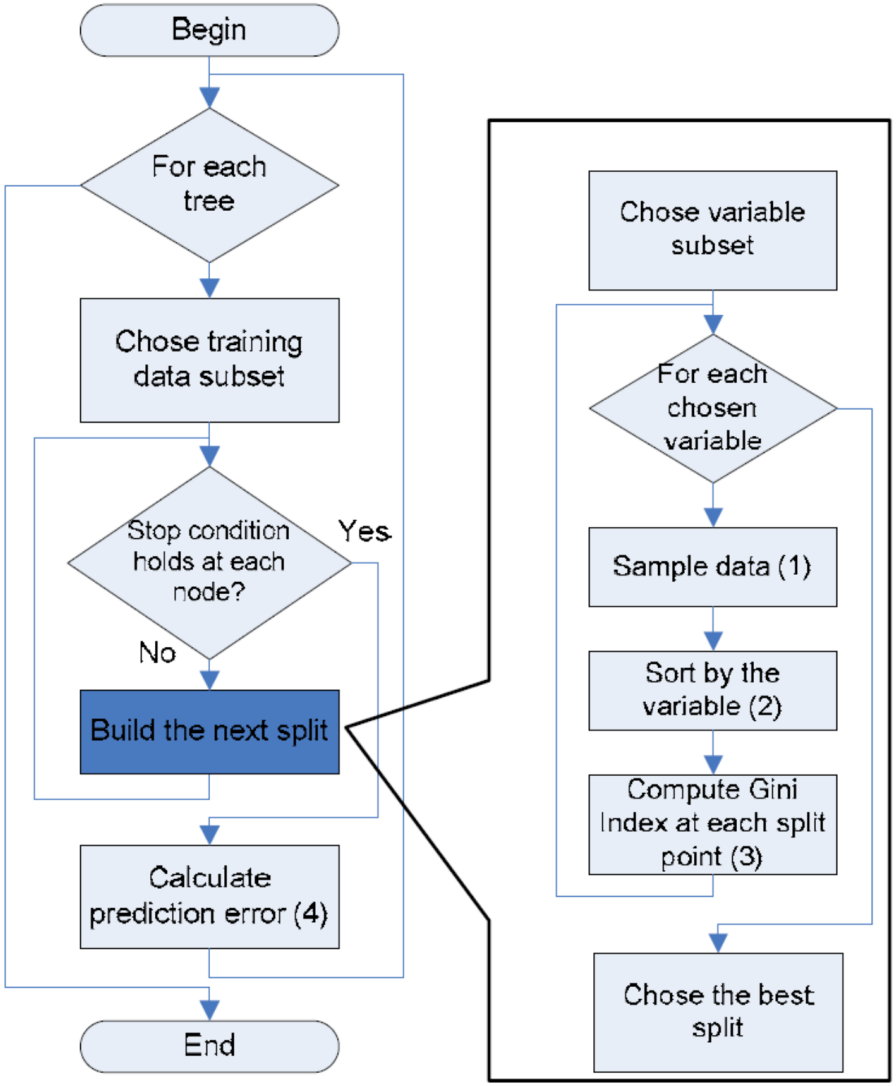
Leo Breiman (2001) "Random Forests", Machine Learning, 45, 5-32.

Random forest algorithm

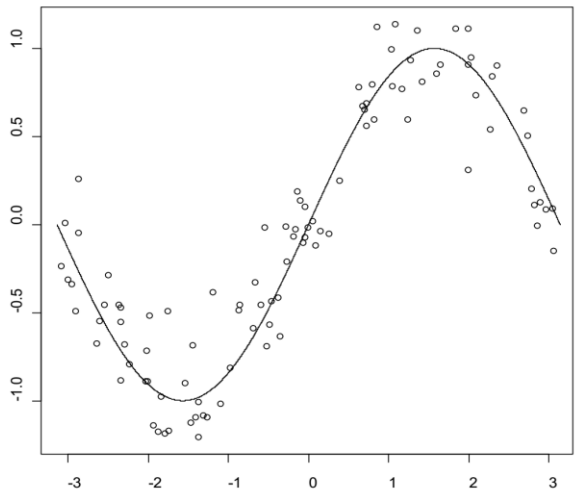
Let **N_trees** be the **number of trees to build** for each of ***N_trees* iterations**:

1. Select a new bootstrap sample from training set
2. Grow an un-pruned tree on this bootstrap.
3. At each internal node, randomly select ***m_try* predictors** and determine the **best split** using only these predictors.
4. Do not perform cost complexity pruning. Save tree as is, along side those built thus far.

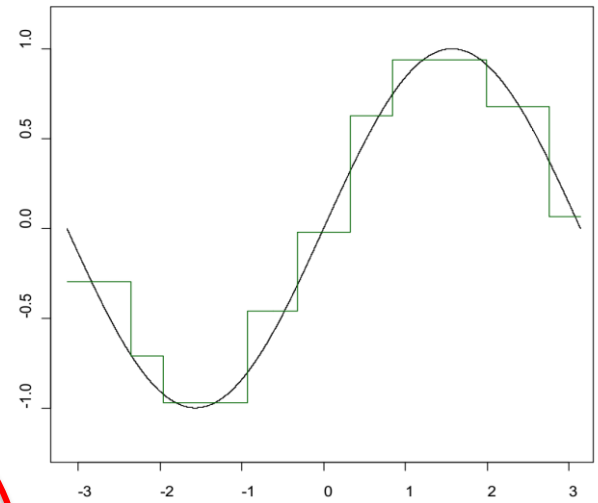
Output overall prediction as the **average response (regression)** or **majority vote (classification)** from all individually trained trees.



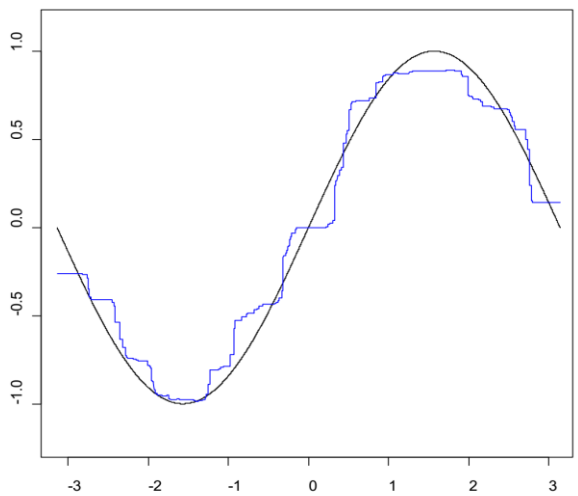
Data and Underlying Function



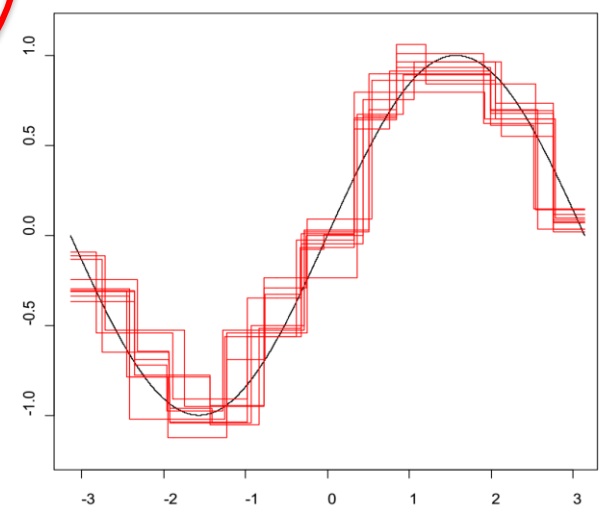
Single Regression Tree

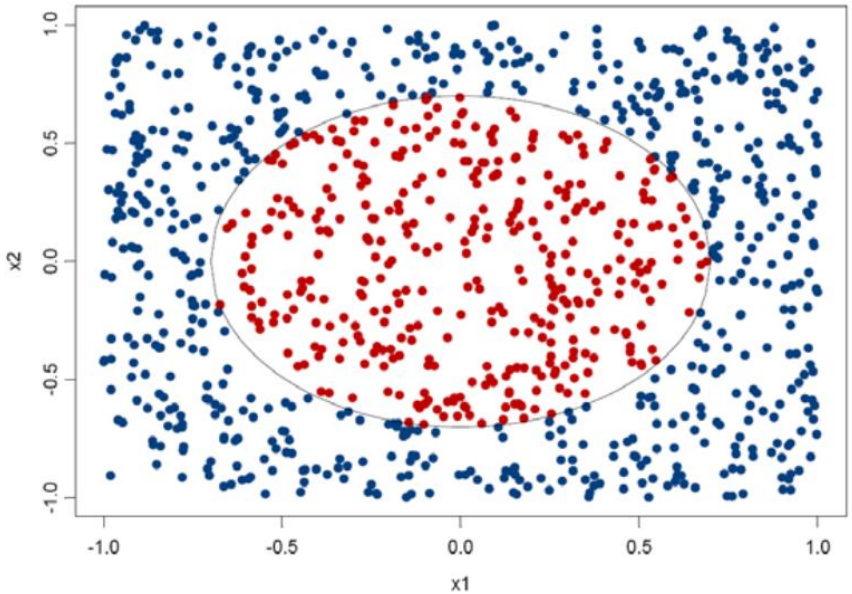


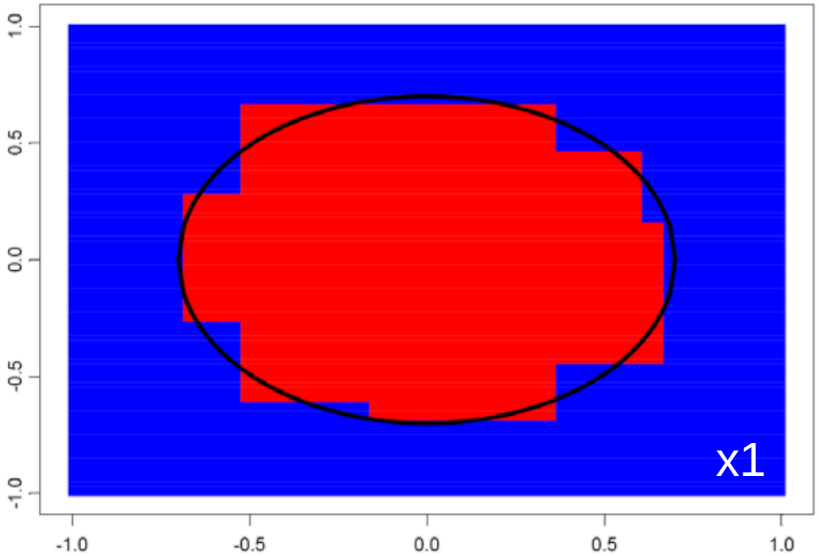
Average of 100 Regression Trees

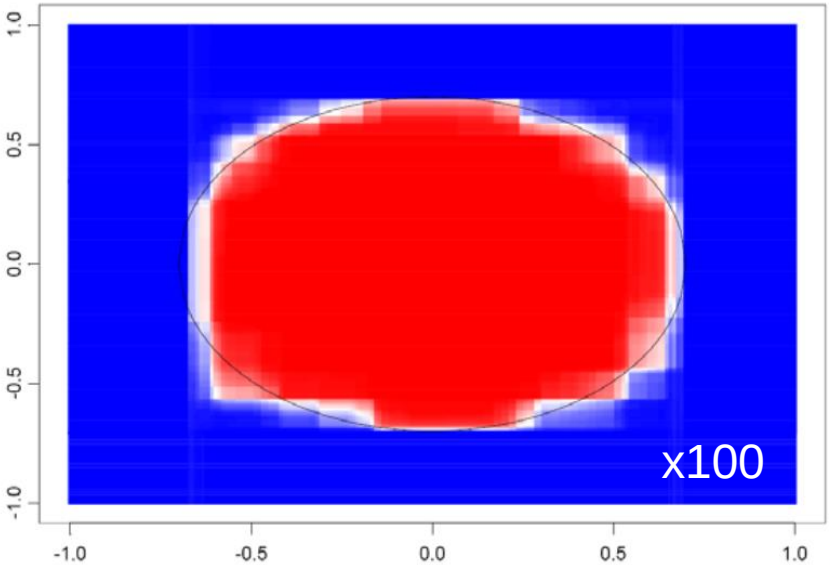


10 Regression Trees




x2
x1
1.0
0.5
0.0
-0.5
-1.0


x1
1.0
0.5
0.0
-0.5
-1.0


x100
1.0
0.5
0.0
-0.5
-1.0

Random forest: practical considerations

Splits are chosen according to a ***purity measure***:

- Squared error (RSS) → regression
- Gini index or deviance → classification

How to select ***N_trees***?

→Build trees until the error no longer decreases.

How to select ***m_try*** ?

→Try the recommended defaults, half of them and twice them and pick the best.

Comparisons: random forest vs SVMs, neural networks

Random forests have about same accuracy as SVMs and neural networks.

RF is more interpretable:

- Feature importance can be estimated during training for little additional computation
- Plotting of sample proximities
- Visualization of output decision trees

RF readily handles larger numbers of predictors.
Faster to train.
Has fewer parameters.

Cross validation is unnecessary: It generates an internal unbiased estimate of the generalization error (test error) as the forest building progresses.

Comparisons: random forest vs boosting

Main similarities

- Both derive many benefits from ensembling, with few disadvantages.
- Both can be applied to ensembling decision trees.

Main differences

- Boosting performs an exhaustive search for best predictor to split on; RF searches only a small subset.
- **Boosting grows trees in series, with later trees dependent on the results of previous trees; RF grows trees in parallel independently of one another.**

Comparisons: random forest vs boosting

Which one to use and when…

- **RF has about the same accuracy as boosting for classification**.

- Boosting may be more difficult to model and requires more attention to parameter tuning than RF.

- On very large training sets, boosting can become slow with many predictors, while **RF which selects only a subset of predictors for each split, can handle significantly larger problems before slowing**.

- **RF will not overfit the data**. Boosting can overfit.

- If parallel hardware is available, (e.g. multiple cores), RF embarrassingly parallel with out the need for shared memory as all trees are independent

Improve on CART with respect to:

Accuracy – Random Forests is competitive with the best known machine learning methods.

Instability – if we change the data a little, the individual trees may change but the forest is relatively stable because it is a combination of many trees.

1. Why bootstrap? (Why subsample?)

Bootstrapping → **out-of-bag data** →

- Estimated error rate and **confusion matrix**
- Variable importance

2. Why trees?

Trees → **proximities** →

- Missing value fill-in
- Outlier detection
- Illuminating pictures of the data (clusters, structure, outliers)

The random forest Predictor

- A case in the training data is *not* in the bootstrap sample for about one third of the trees (we say the case is “out of bag” or “oob”).
- Vote (or average) the predictions of *these trees* to give ***the RF predictor.***
- The ***oob error rate*** is the error rate of the ***RF predictor.***
- The ***oob confusion matrix*** is obtained from the ***RF predictor.***
- For new cases, vote (or average) *all* the trees to get the ***RF predictor.***

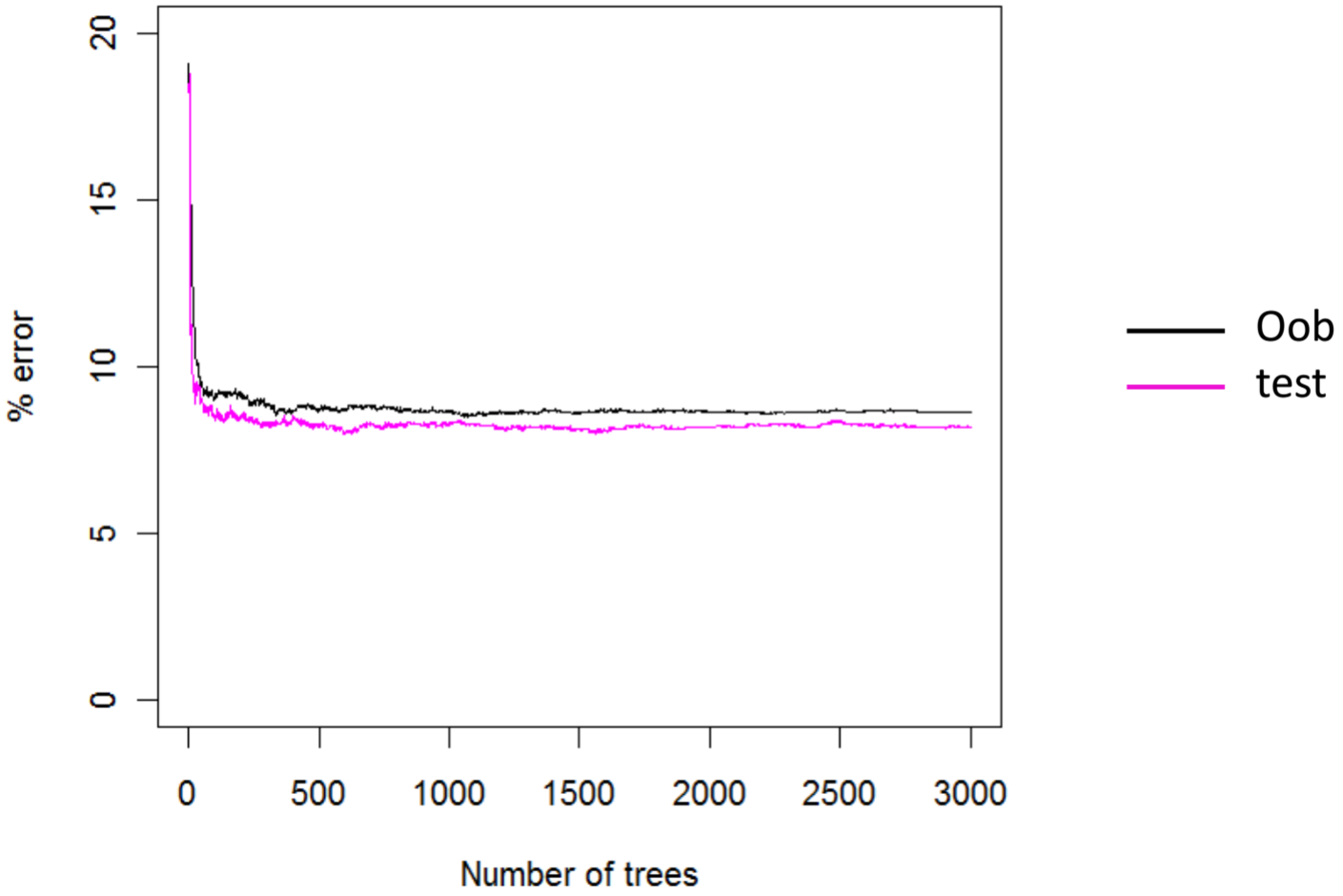
For example, suppose we fit 1000 trees, and a case is out-of-bag in 339 of them, of which:

- 283 say “class 1”
- 56 say “class 2”

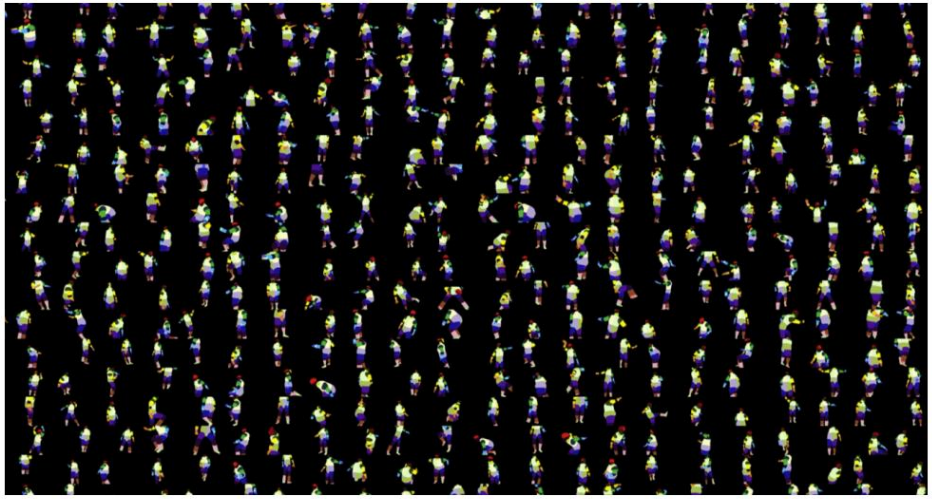
The RF predictor for this case is class 1.

The “oob” error gives an estimate of test set error (generalization error) as trees are added to the ensemble.

RFs do not overfit as we fit more trees



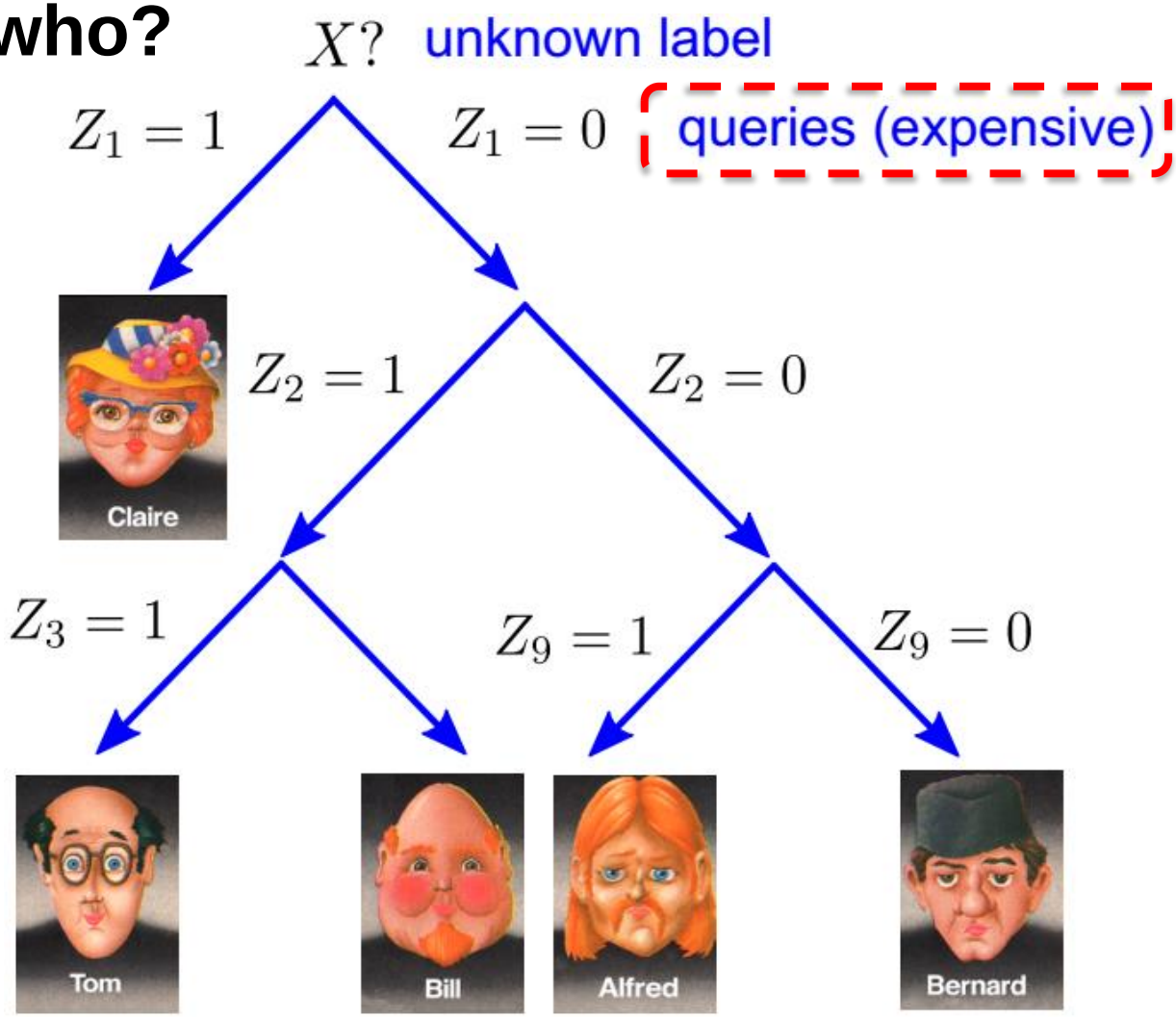
Random forests: body part classification in the kinect



1 million test images, 1 day using a 1000 core cluster

Shotton, Jamie, et al. "Real-time human pose recognition in parts from single depth images." *Communications of the ACM* 56.1 (2013): 116-124.

Guess who?


X? unknown label
$Z_1 = 1$
$Z_1 = 0$
queries (expensive)
Claire
$Z_2 = 1$
$Z_2 = 0$
$Z_3 = 1$
$Z_9 = 1$
$Z_9 = 0$
Tom
Bill
Alfred
Bernard

Accuracy in Classification

Confusion matrix

		Predicted Class		Total
		0	1	
Actual Class	0	a	b	a + b
	1	c	d	c + d
	Total	a + c	b + d	n

Specificity = a/(a + b)

Sensitivity = d/(c + d)

Error rate:

b/(a + b) for class 0

c/(c + d) for class 1

(c + b)/n overall