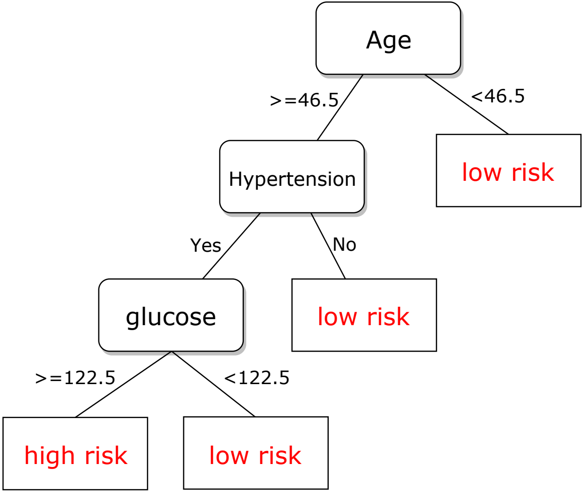
Decision Trees

Decision trees

Framingham dataset:
high risk or low risk of heart attack?

- create subsequent *rules* to split the data by the values of features
- can split at **numerical** or **categorical** features

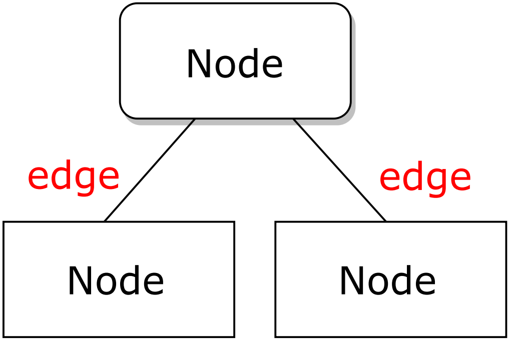


Components of a tree

Definitions

Node
A basic unit that contains *data* (can be a feature or a decision)

Edge
The connection between two nodes



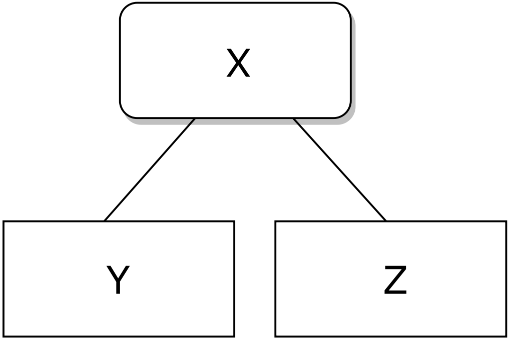
Components of a tree

Definitions

Child
A connected node below.

Parent
A connected node above.

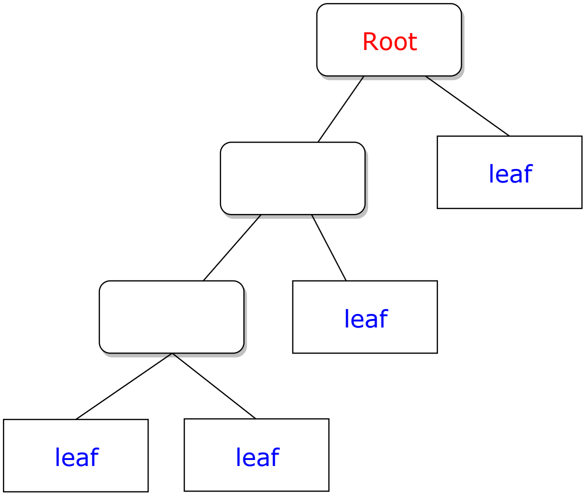
For example, Y and Z are children of X,
X is a parent of Y and Z.



Components of a tree

Root
The top node in a tree.

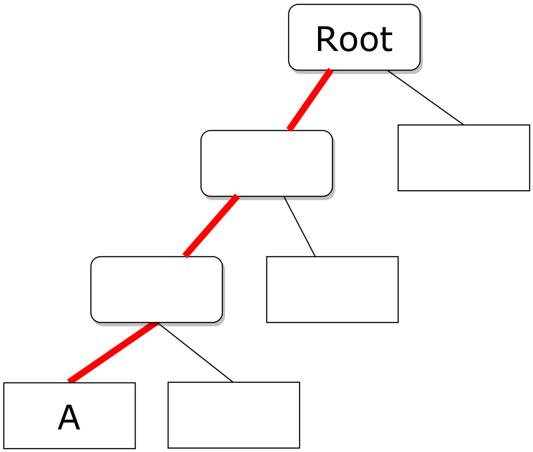
Leaf
A node with no further edge



Components of a tree

Depth of a node
The number of edges to travel from the root to that node
Example: The depth of **A** is 3

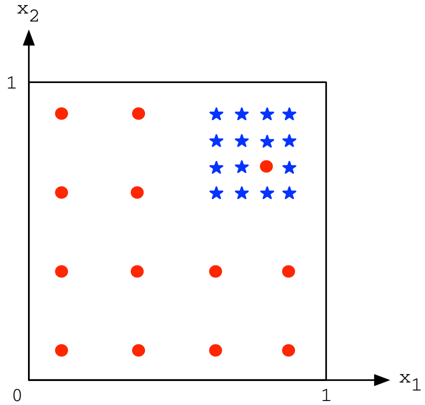
Maximum depth
Maximum of all possible depths in the tree



Example

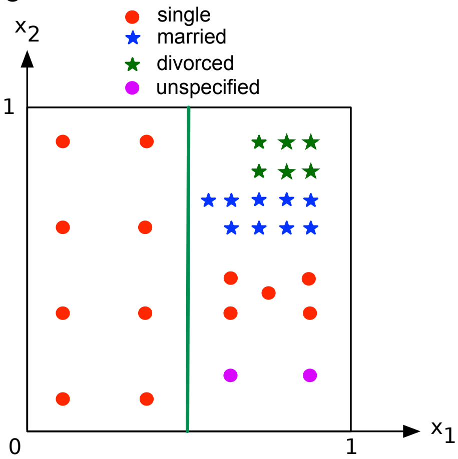
Toy data with 2 features

We'll try Maximum Depth = 2 (in other words, 2 splits)



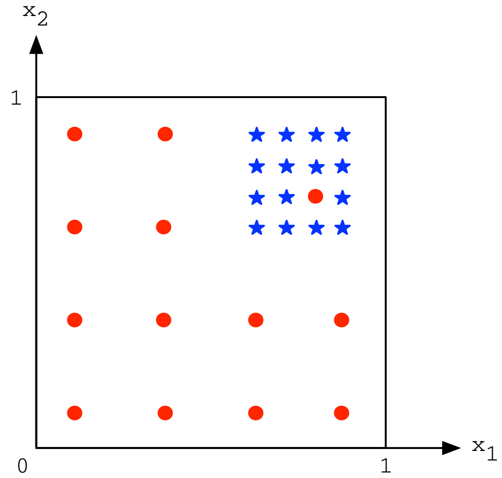
Categorical features

No one-hot encoding needed!

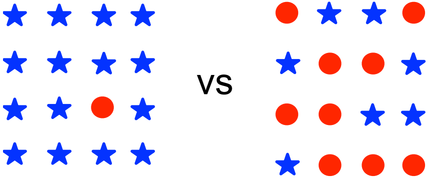


Uncertainty

How can we *quantitatively* determine the best value to split?



Information measure

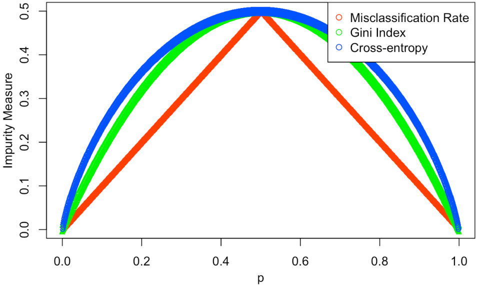


Measure the **mixture** of points by a function I:

Misclassification rate: $I(S) = 1 - \max(p, 1-p)$

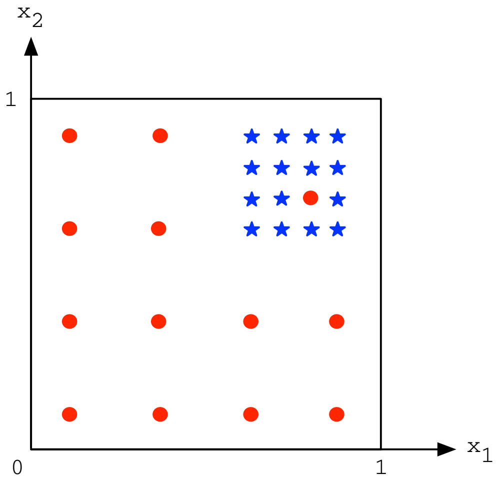
Entropy: $I(S) = -p \log p - (1-p)\log(1-p)$

Gini index: $I(S) = p(1-p)$

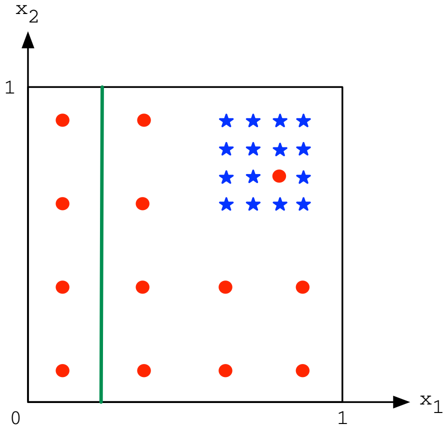


Expected information

Expected information of a split is the weighted average of the measures

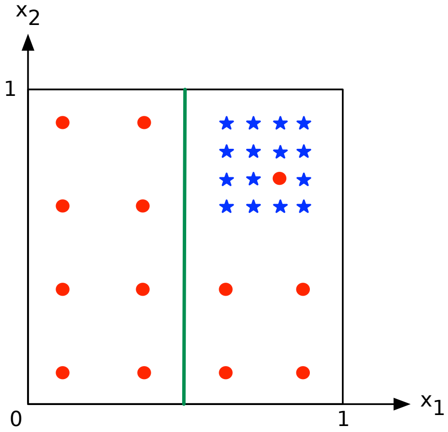


Example



Expected information $=$

Example

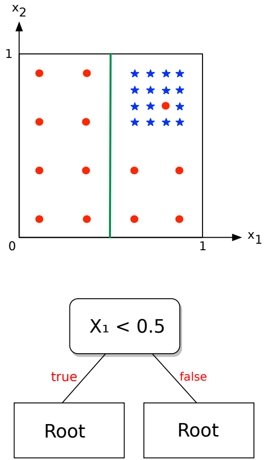


Expected information $=$

Tree splitting algorithm

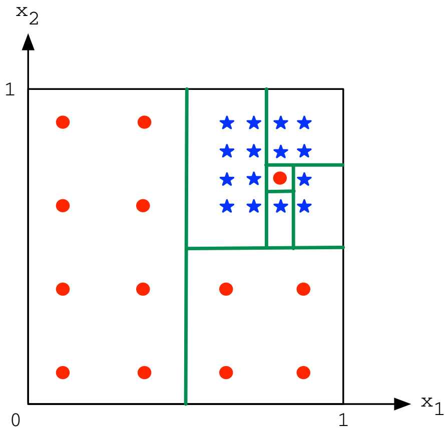
`for` each **leaf** in the tree `do`:
    `for` each **feature** `do`:
        `for` each **splitting value** `do`:
            compute **expected information**

`split` at **leaf+feature+value** with smallest expected information



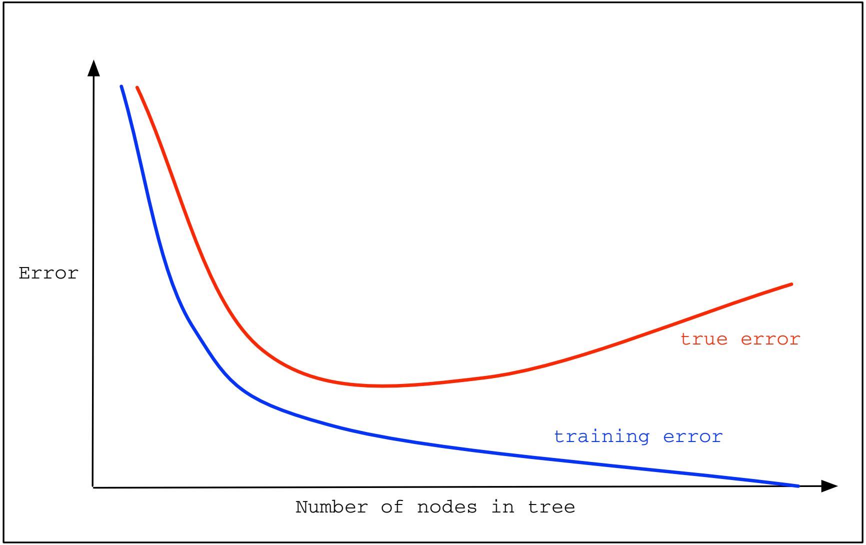
Overfitting

We can keep going until we get 100% accuracy on the training set



But the model **overfits** the data: that one red point is probably an outlier.

Test error as number of nodes grow



Preventing overfitting

How to reduce overfitting?

- specify the minimum number of samples required to split (`min_samples_split`)
- specify the maximum depth of the tree (`max_depth`)
- specify the minimum number of samples in each child node (`min_samples_leaf`)
- specify the maximum number of features to consider at each split (`max_features`)
- pruning i.e. build a full tree then remove the nodes until (cross) validation accuracy stops improving.

All these options are available in `scikit-learn`

Decision tree: pros and cons

Advantages

- Simple, Easy to interpret
- Accept both numerical and categorical features.
- Accept any number of classes.
- Can fit to any dataset.

Decision tree: pros and cons

Advantages

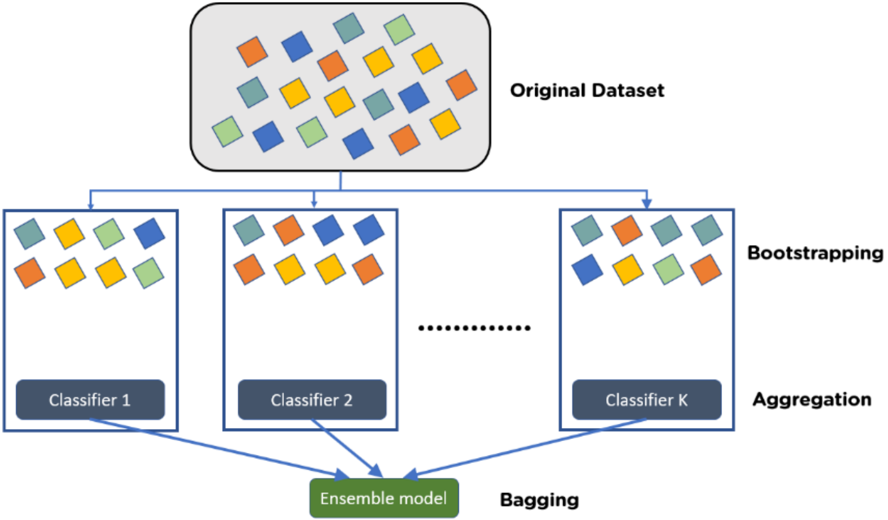
- Simple, Easy to interpret
- Accept both numerical and categorical features.
- Accept any number of classes.
- Can fit to any dataset.

Disadvantages:

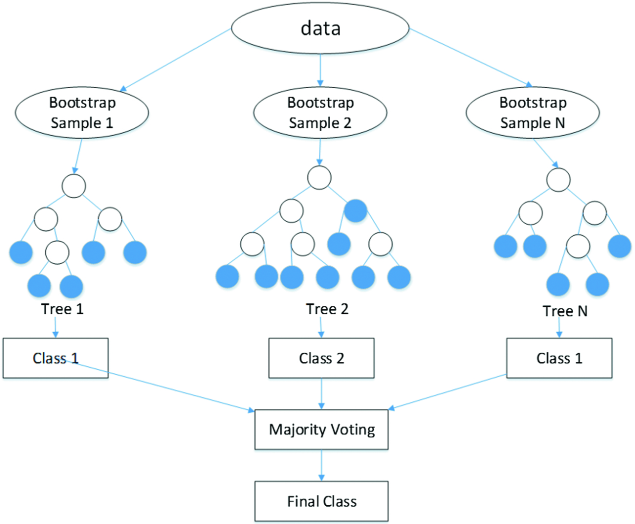
- Decision trees performs (e.g. test accuracy) worse than SVM and sometimes logistic regression
- How can we improve decision trees?

Random Forest

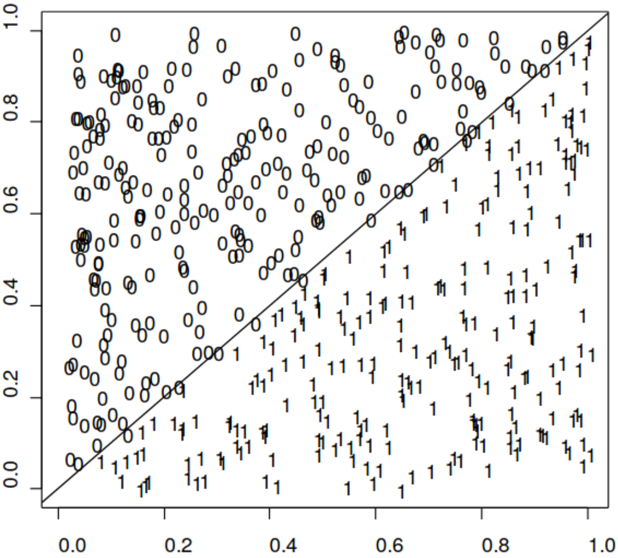
Bagging


Original Dataset
Bootstrapping
Classifier 1
Classifier 2
Classifier K
Aggregation
Ensemble model
Bagging

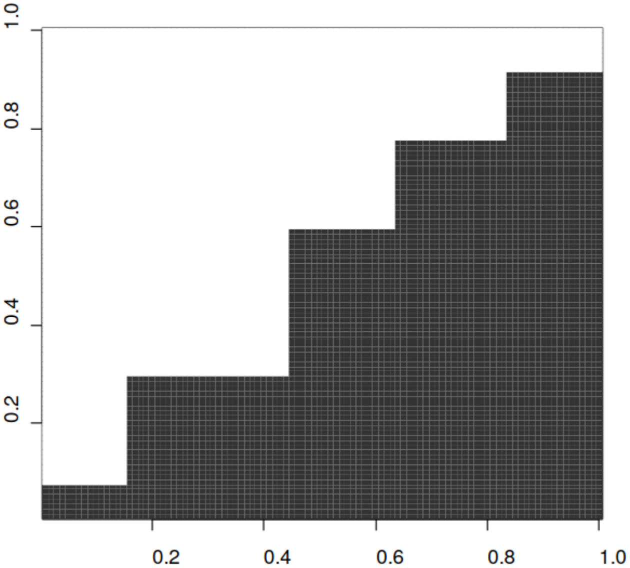
Decision trees with bagging



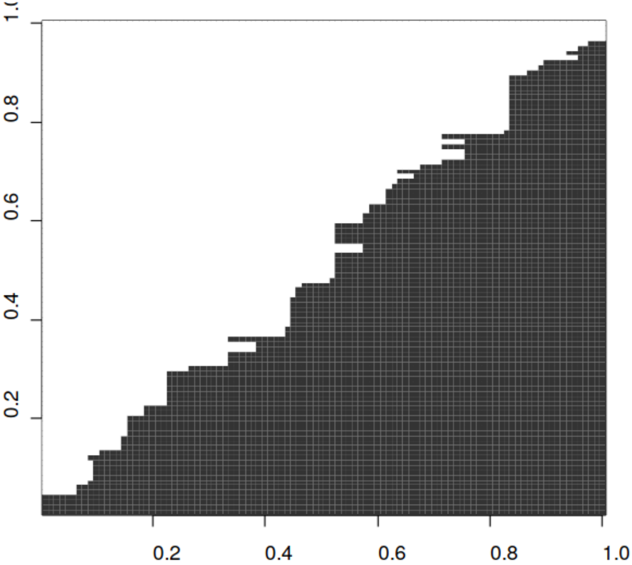
Hard problem for a single tree



A single tree



25 Voted tree

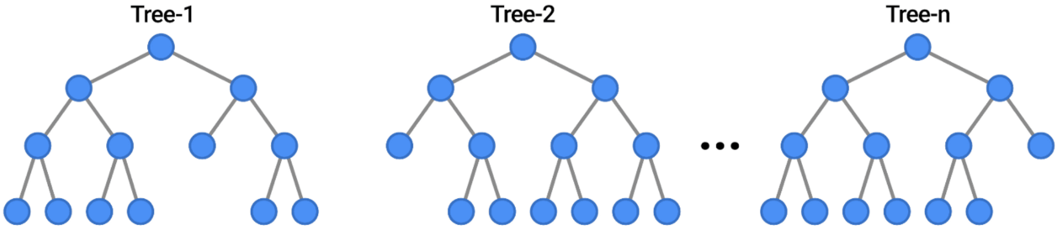


Random forest

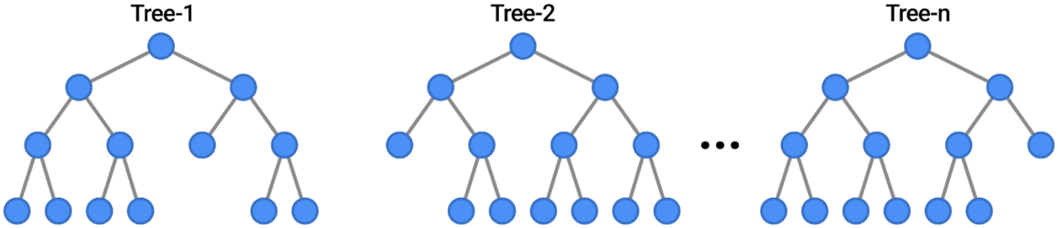
Random forest is a decision trees with bagging

$+$ one more source of randomness:

- At each split, select a **random subset of features**
- If there are d features, $\sqrt{d}$ features are used in each split



Hyperparameters in random forest



Random forest algorithm

`RandomForest(T, m, D,` n_0`)`

Given a data set S of n labeled points:

`for` $t = 1$ to T:

Bootstrap $n_0 < n$ points from S

Fit a decision tree h_t to these points

At each node,

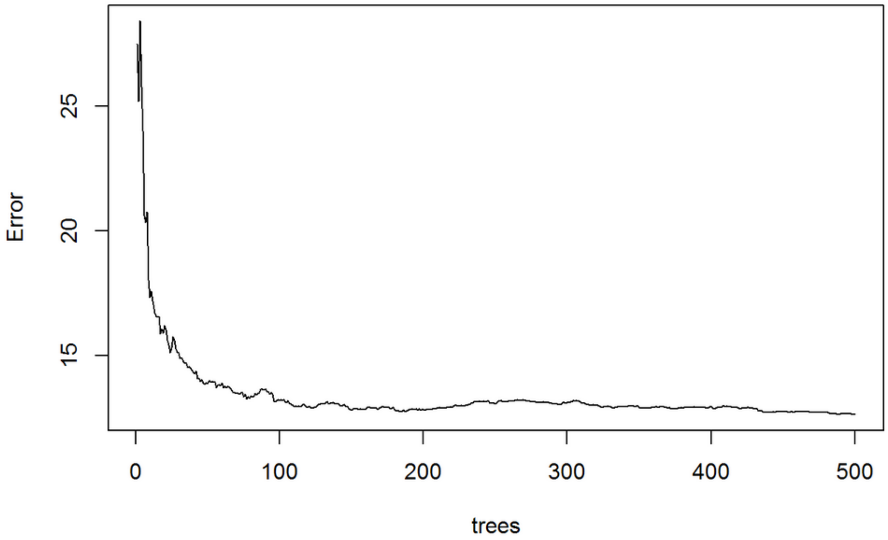
Select m variables at random from d variables

Find the best split on the selected m variables

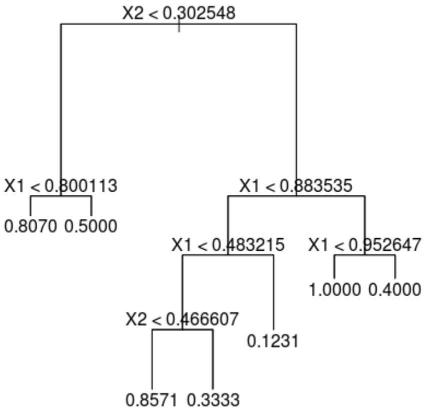
Grow the tree to maximum depth

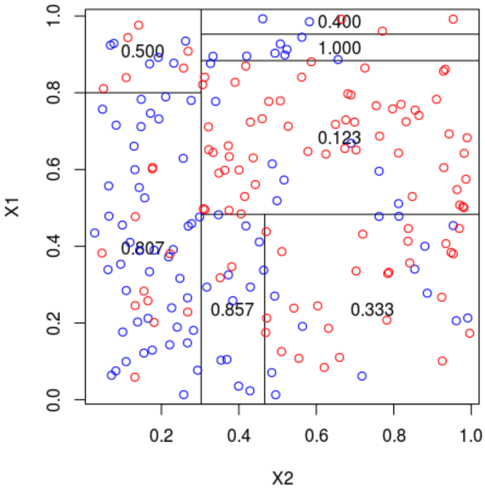
Final prediction: majority vote of $h_1, \ldots, h_T$

Test errors vs number of trees



Regression trees





Splitting in regression tree

Random forest