Ninth Session

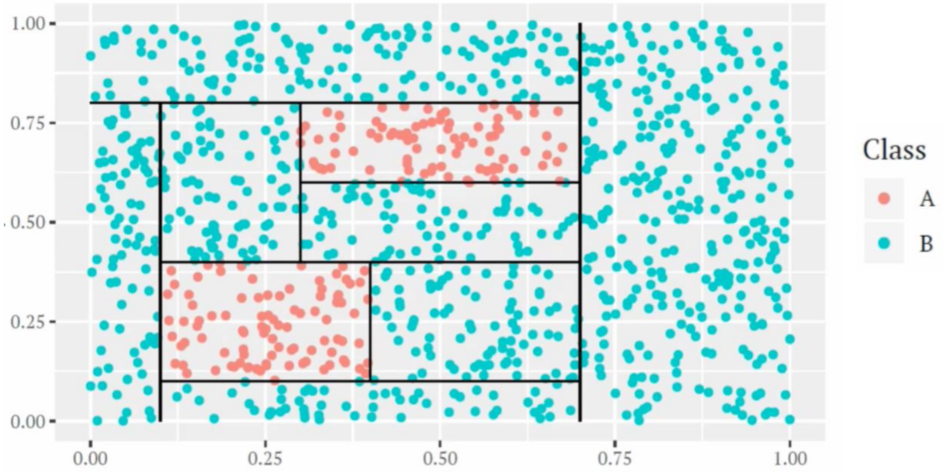
Alireza Moradi

Decision Tree

Decision tree

- **Tree-based methods** involve segmenting the predictor space into several simple regions.

- Since the set of splitting rules used to segment the predictor space can be summarized in a tree, these types of approaches are known as **decision-tree methods**.



✓ Tree-based methods are simple and useful for **interpretation**.

✓ Decision trees can be applied to both regression and classification problems.

Decision tree

- Before we see the terminologies for trees, let's have a dataset example:

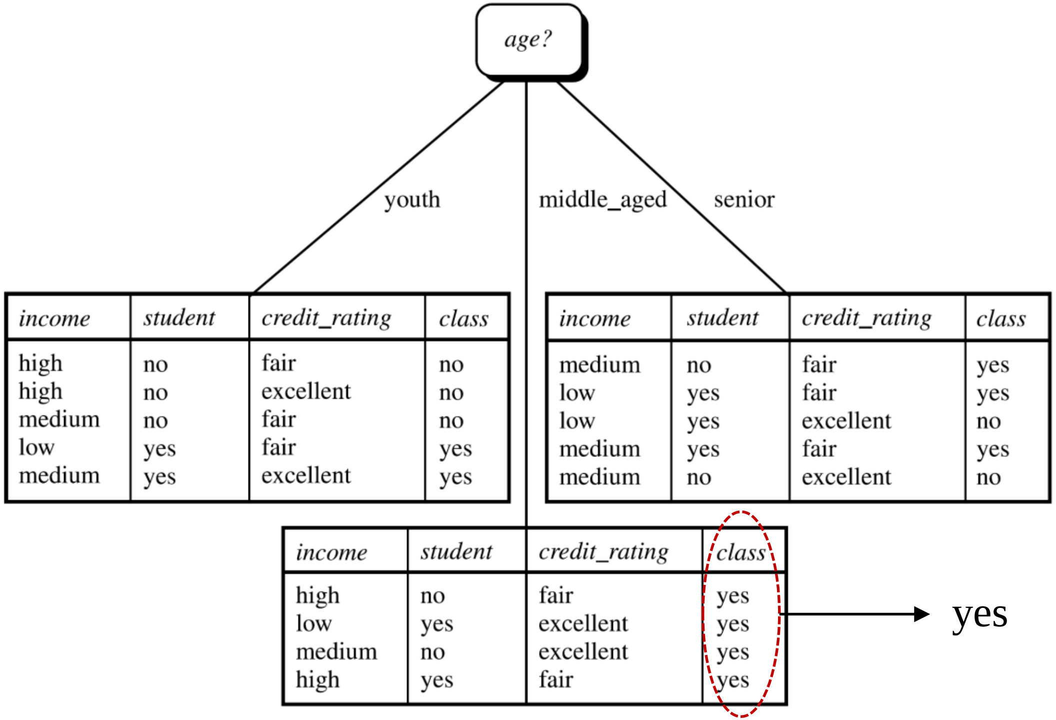
✓ Features:
✓ Target:

index	age	income	student	credit rating	buys computer
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle-aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle-aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle-aged	medium	no	excellent	yes
13	middle-aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

Branching

- Asking questions about each feature:

index	age	income	student	credit rating	buys computer
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle-aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle-aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle-aged	medium	no	excellent	yes
13	middle-aged	high	yes	fair	yes
14	senior	medium	no	excellent	no



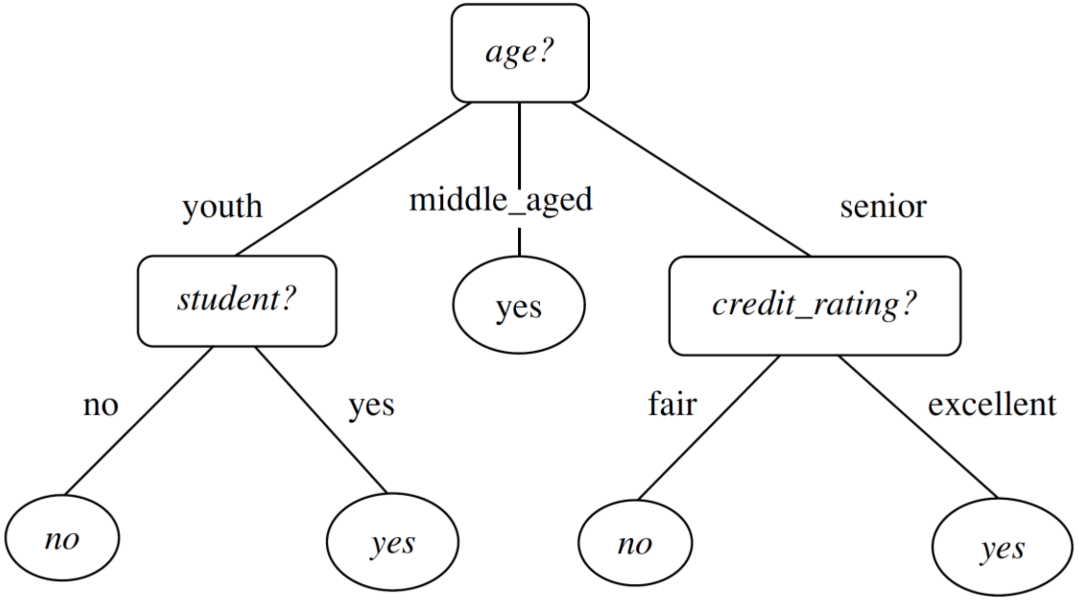
Terminology for Trees

- Decision trees are typically drawn upside down, in the sense that the leaves are at the bottom of the tree.

- **Decision nodes** are denoted by rectangles, and **leaf nodes** are denoted by ovals:

- decision nodes = questions

- Decision trees can easily be converted to **classification rules**.



Terminology for Trees

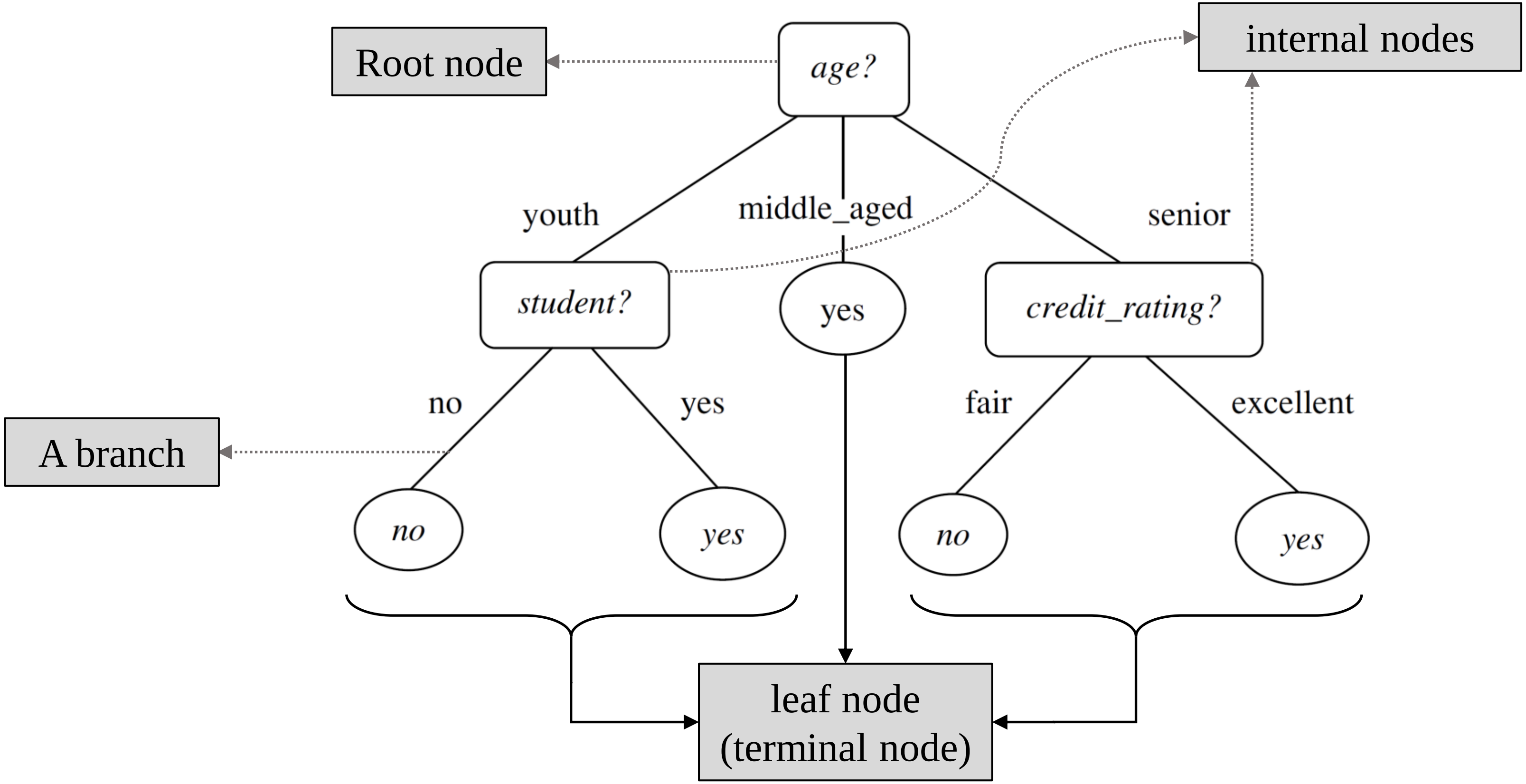
- The topmost node in a tree is the **Root node**. It has no incoming edges, only outgoing ones leading to further decisions. This node contains all the training data.

- **Internal nodes** are non-leaf nodes within the tree, representing further splitting of the data based on specific features. They have one incoming edge from the previous node and multiple outgoing edges leading to different branches.

- The connections between nodes are called **branches**. Each branch represents an outcome of the question.

- **Leaf Nodes (Terminal Nodes)** are the end points of the tree, representing the final classification or prediction. They have one incoming edge from the previous node but no outgoing edges.

Terminology for Trees



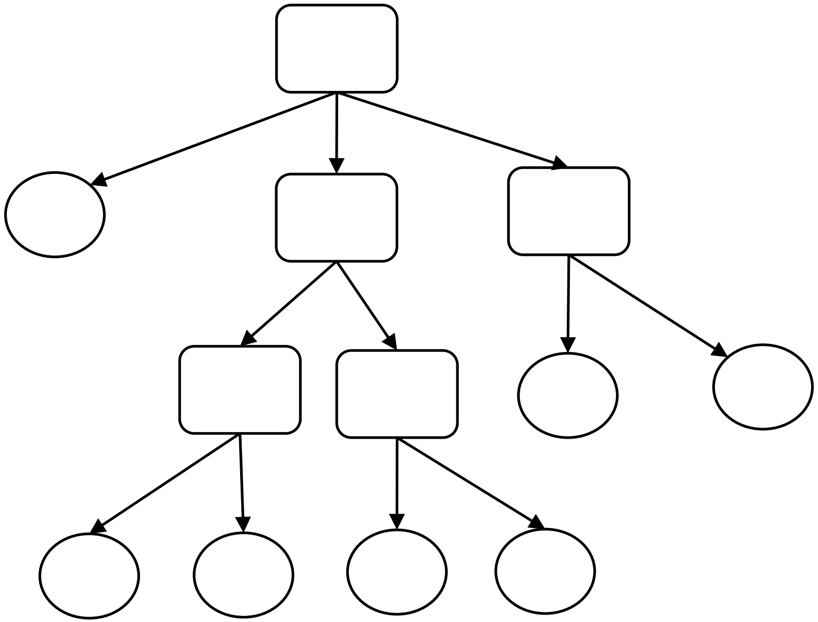
Terminology for Trees

- **Depth** of a tree is the length of the longest path from a root to a leaf.

- **Size** of a tree is the **number of terminal nodes** in the tree.

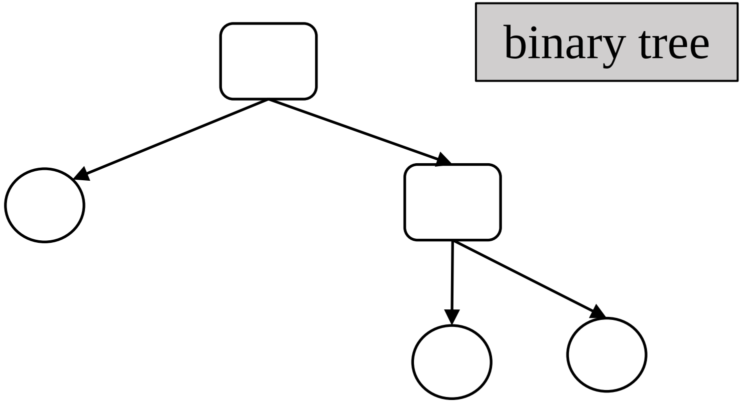
- Example:

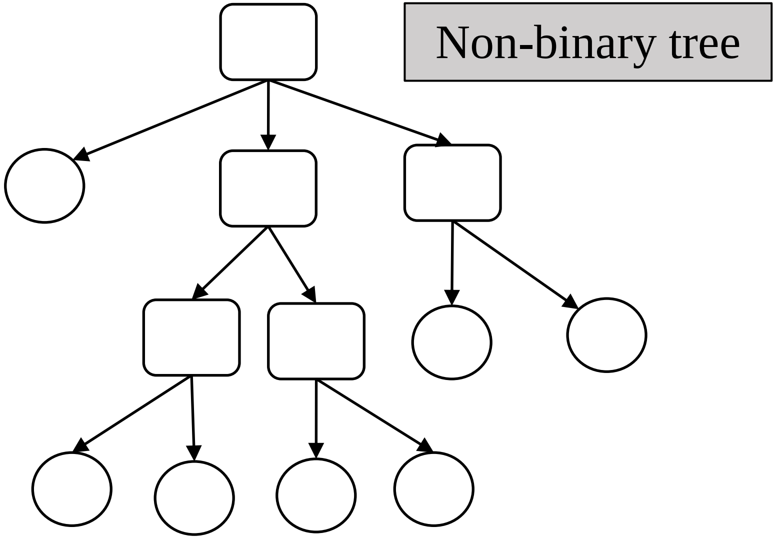
✓ Depth =
✓ Size =



Terminology for Trees

- Some decision tree algorithms produce only **binary trees** where each internal node branches to exactly two other nodes, whereas others can produce **nonbinary trees**.



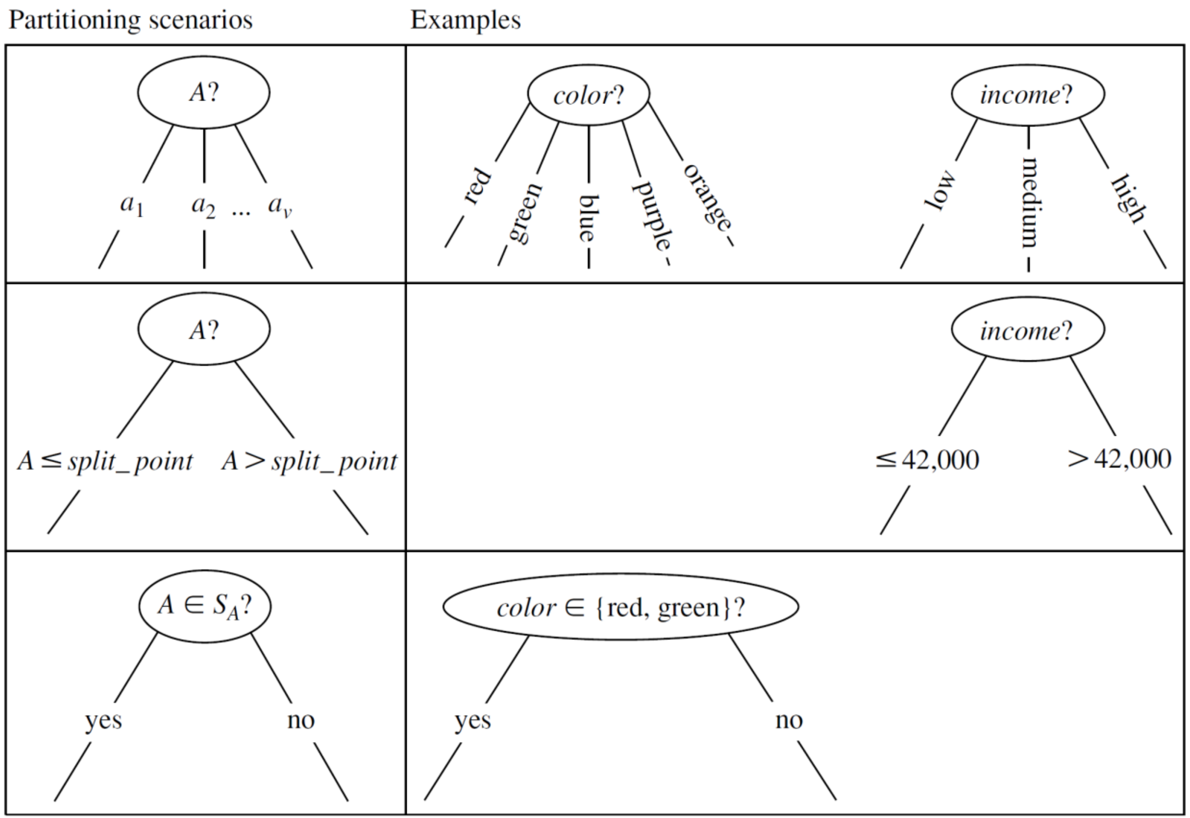


- In this course, we focus on non-binary trees.

Branching

- Branching different types of attributes in decision trees:

Branching = partitioning

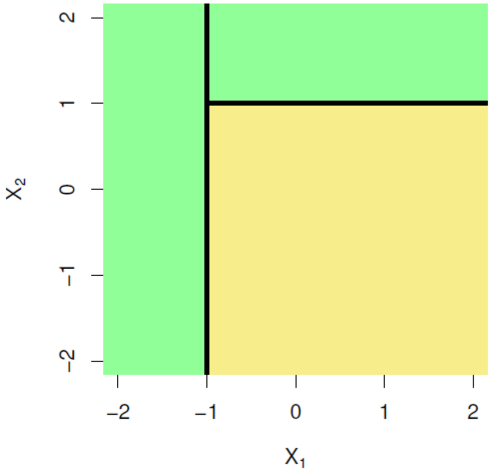


Decision tree; summary

1. We divide the predictor space -that is, the set of possible values for $X_1, X_2, \dots, X_p$- into J distinct and **non-overlapping regions, $\boldsymbol{R_1, R_2, \dots, R_J}$.**

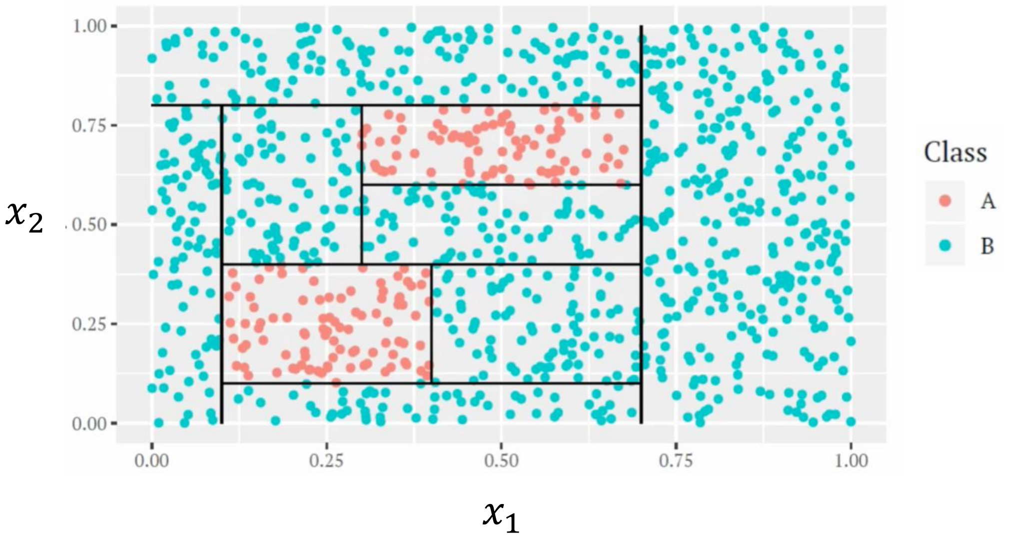
2. For every observation that falls into the region R_j , we make the same prediction. For a classification tree, we predict that each observation belongs to the most commonly occurring class of training observations in the region to which it belongs.

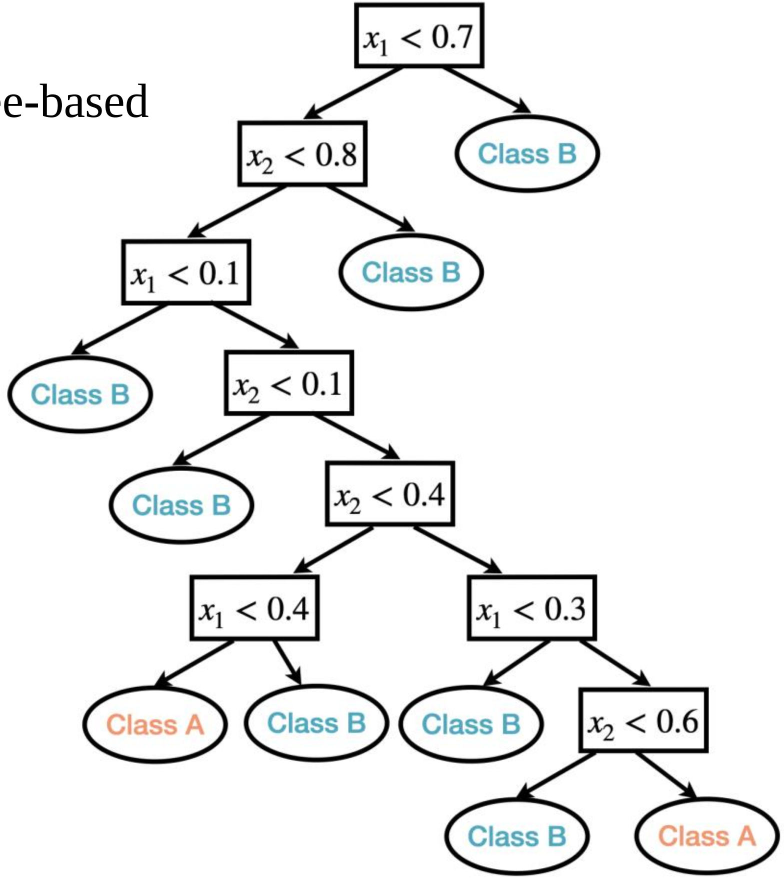
Majority voting



Terminal nodes

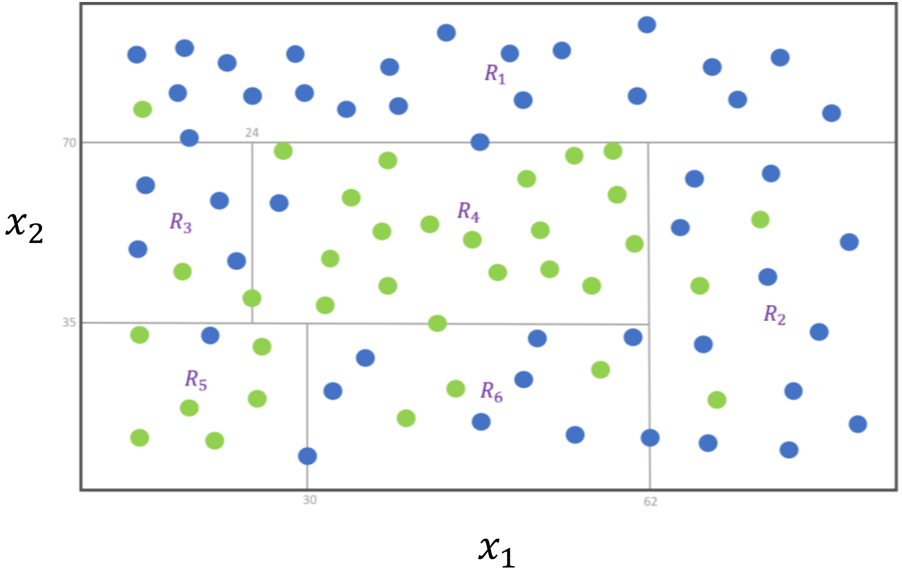
- For example, the following figures show a tree-based classification model built on two predictors.

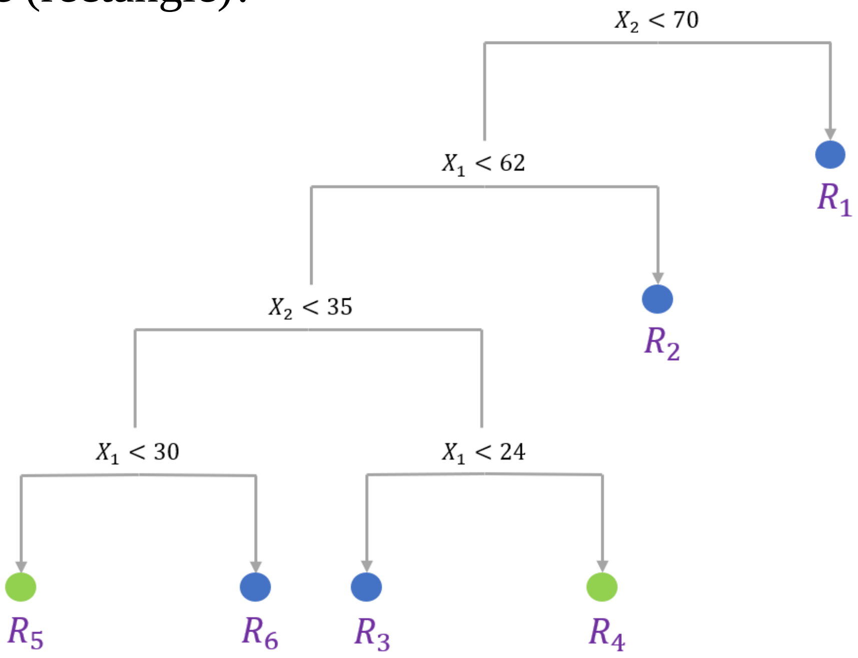




Terminal nodes

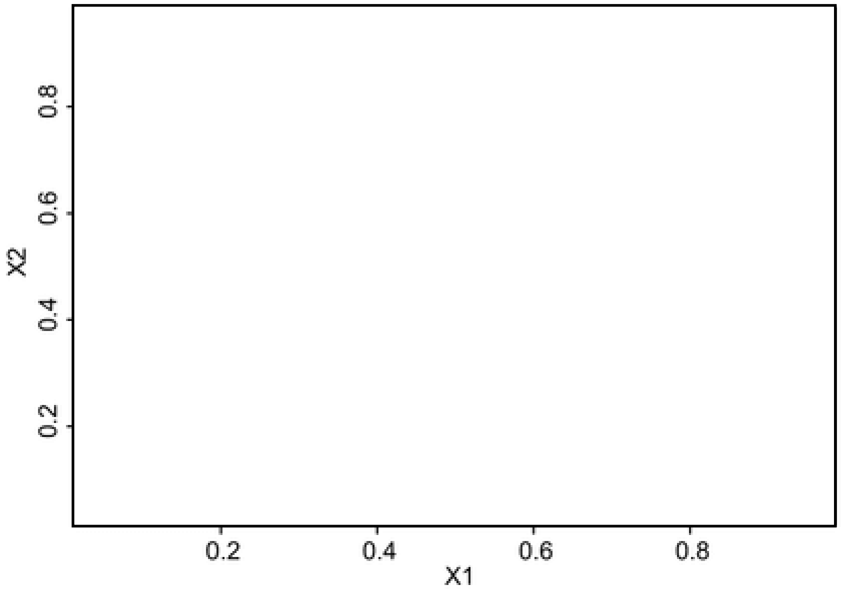
✓ If we fix the depth of the tree, what happens if we cannot achieve perfect separation of observations within each terminal node (rectangle)?

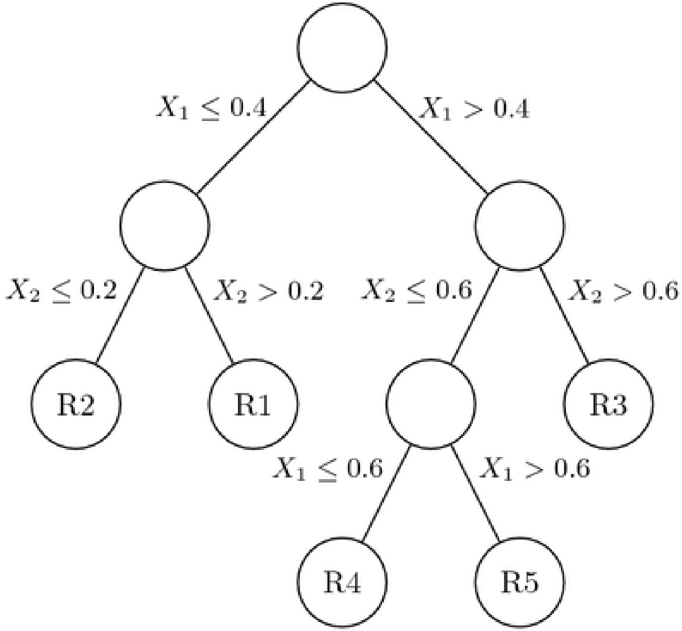




Decision tree

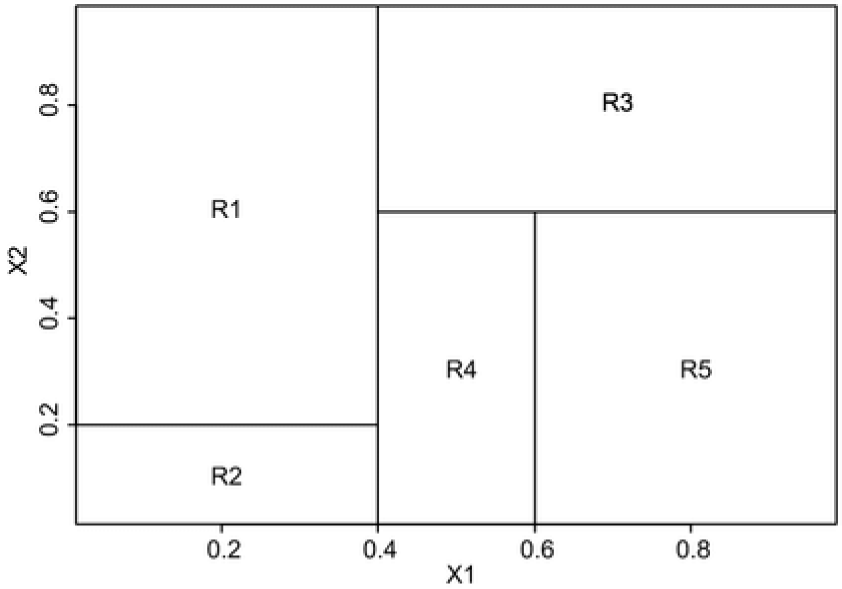
- Example:

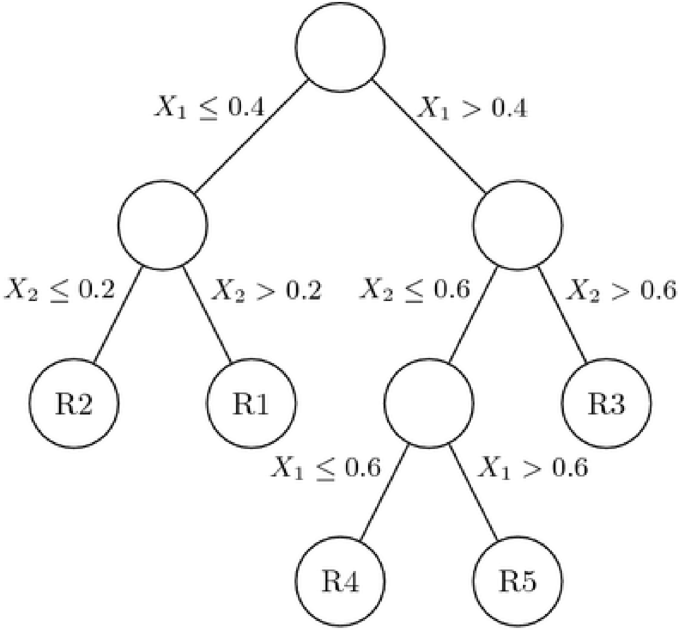




Decision tree

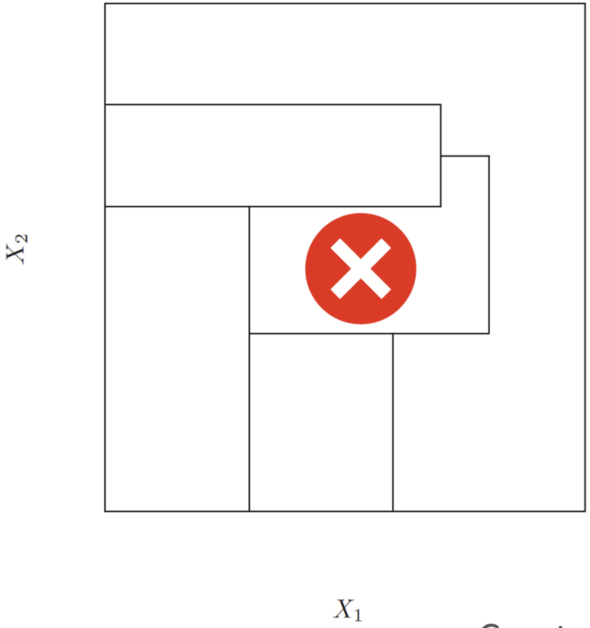
- Solution:

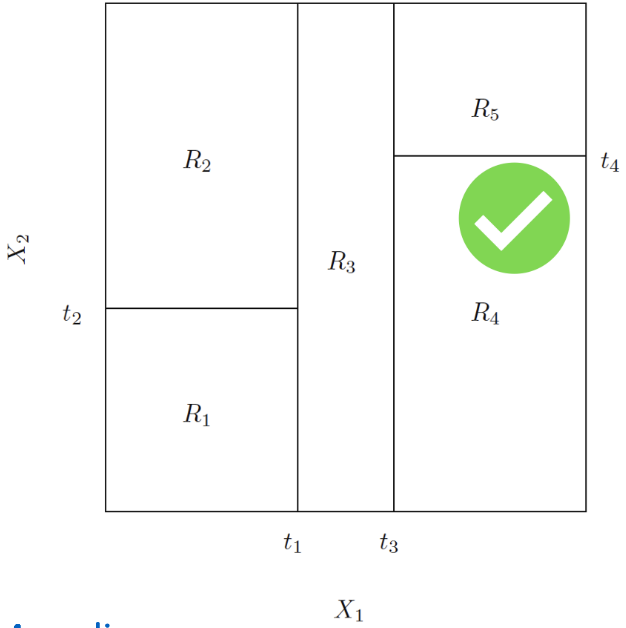




Decision tree

- In theory, the regions R_j could have any shape. But we choose **hyper-rectangles** for simplicity. The left panel is an example of an invalid partitioning, and the right is an example of a valid one.





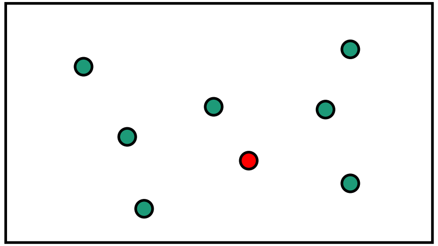
Decision tree induction

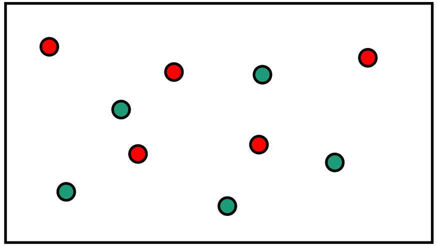
- **Decision tree induction** is the learning of decision trees from class-labeled training tuples.

- The goal is to find hyper-rectangles $R_1, R_2, \dots, R_J$ that maximize purity.

- In other words, if we split up the observations according to $R_1, R_2, \dots, R_J$, we hope for the **resulting partitions to be as pure as possible**.

- A partition is **pure** if all the observations in it belong to the same class.





Measuring purity

- The **Gini index** is referred to as a measure of purity. A small value indicates that a node contains predominantly observations from a single class.

$$G(\mathrm{D}) = 1 - \sum_{i=1}^{C} p_i^2$$

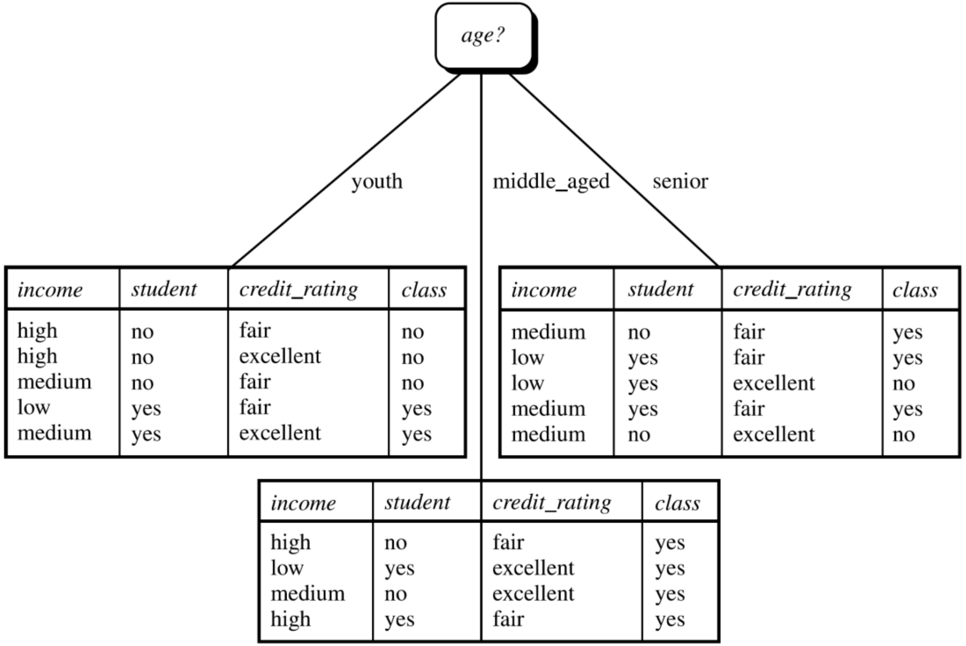
- Where p_i is the probability that an observation in D belongs to class C_i and is estimated by:

$$p_i = \frac{|C_{i,D}|}{|D|}$$

- D is a data partition or set of training examples.

Measuring purity

- D (data partition):



youth:

income	*student*	*credit_rating*	*class*
high	no	fair	no
high	no	excellent	no
medium	no	fair	no
low	yes	fair	yes
medium	yes	excellent	yes

middle_aged:

income	*student*	*credit_rating*	*class*
high	no	fair	yes
low	yes	excellent	yes
medium	no	excellent	yes
high	yes	fair	yes

senior:

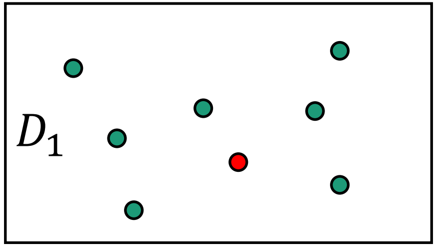
income	*student*	*credit_rating*	*class*
medium	no	fair	yes
low	yes	fair	yes
low	yes	excellent	no
medium	yes	fair	yes
medium	no	excellent	no

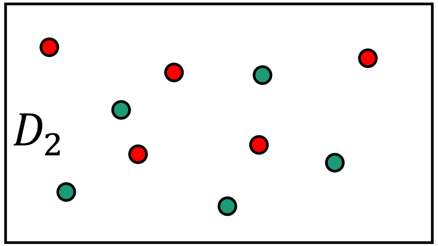
index	age	income	student	credit rating	buys computer
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle-aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle-aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle-aged	medium	no	excellent	yes
13	middle-aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

Measuring purity

- Example:

$$G(\mathrm{D}) = 1 - \sum_{i=1}^{C} p_i^2 \qquad p_i = \frac{|C_{i,D}|}{|D|}$$





$$p_{red} = \frac{1}{8}$$

$$p_{green} = \frac{7}{8}$$

$$G(D_1) = 1 - \left(\frac{1}{8}\right)^2 - \left(\frac{7}{8}\right)^2 = 0.22$$

$p_{red} =$

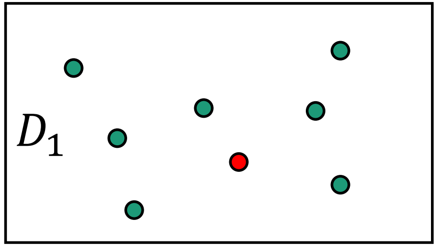
$p_{green} =$

$G(D_1) =$

Measuring purity

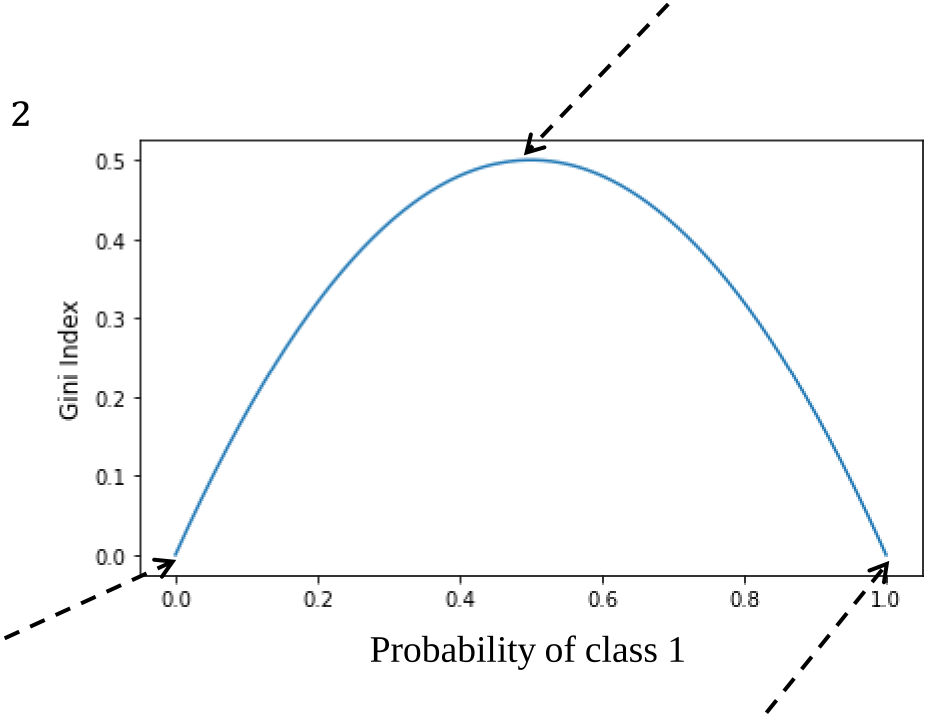
- In a binary classification problem:

$$p_{Class\,1} = 1 - p_{Class\,2}$$



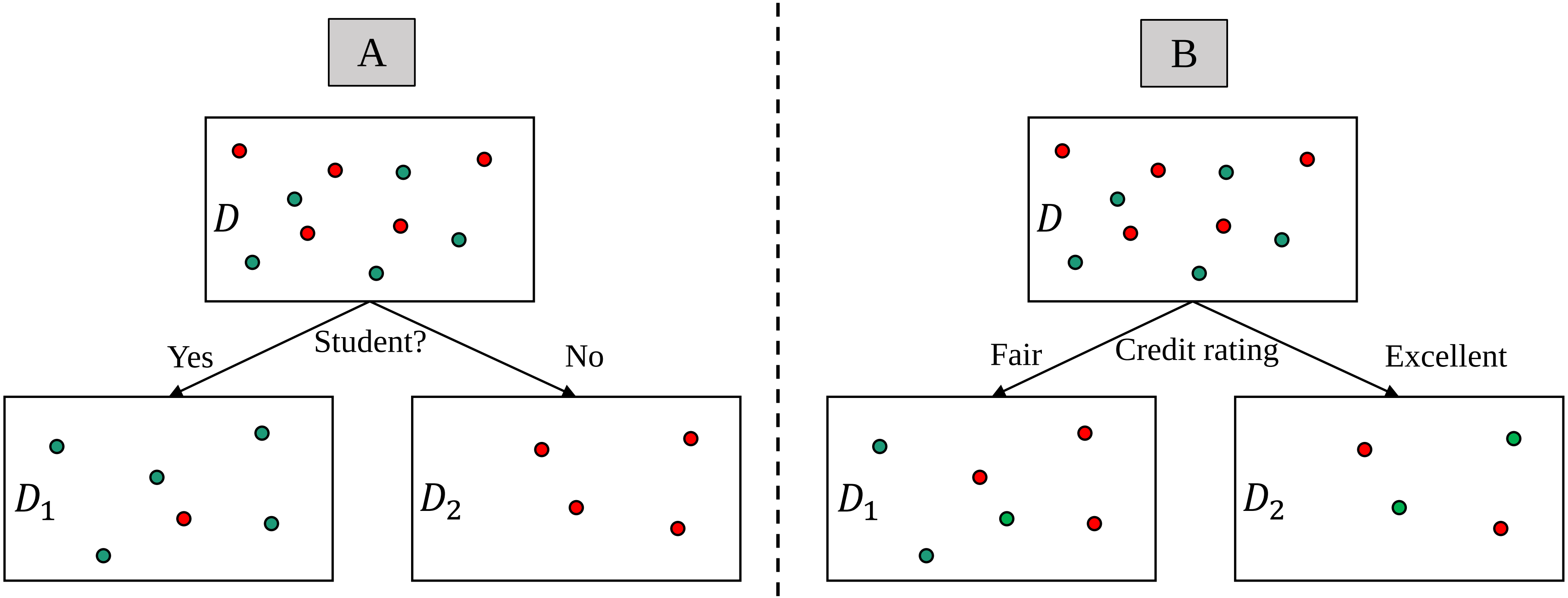
$p_{red} = \frac{1}{8}$

$p_{green} = \frac{7}{8}$



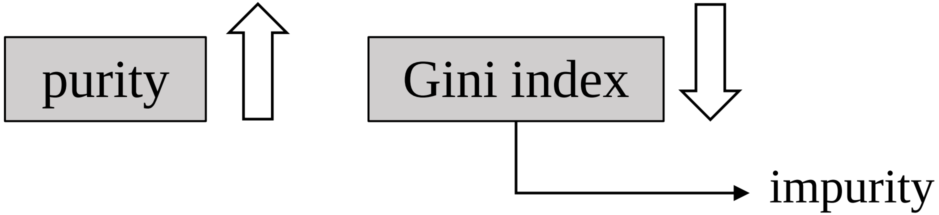
Measuring purity

- Which one you prefer?



Measuring purity

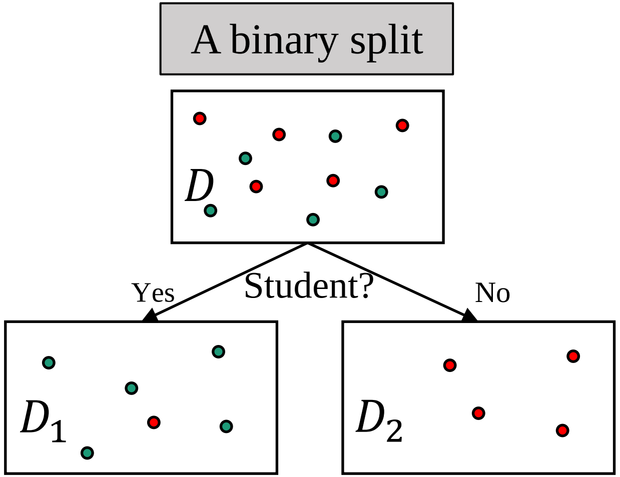
- We want to increase purity!



- When considering a split**, we compute a weighted sum of the impurity of each resulting partition**.

- For example, if a binary split on A partitions D into D1 and D2, the Gini index of D given that partitioning is:

$$Gini_A(D) = \frac{|D_1|}{|D|} Gini(D_1) + \frac{|D_2|}{|D|} Gini(D_2)$$



Measuring purity

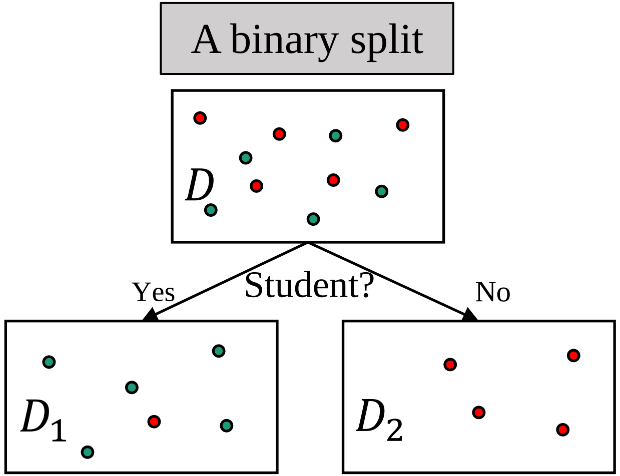
- The reduction in impurity (Gini index):

$$\Delta\, Gini(A) = Gini\,(D) \;-\; Gini_A(D)$$

- Example :

$$\Delta\, Gini(Student?\,) = Gini\,(D) \;-\; Gini_{Student?}(D)$$

$$Gini_{Student?}(D) = \frac{|D_1|}{|D|} Gini(D_1) + \frac{|D_2|}{|D|} Gini(D_2)$$

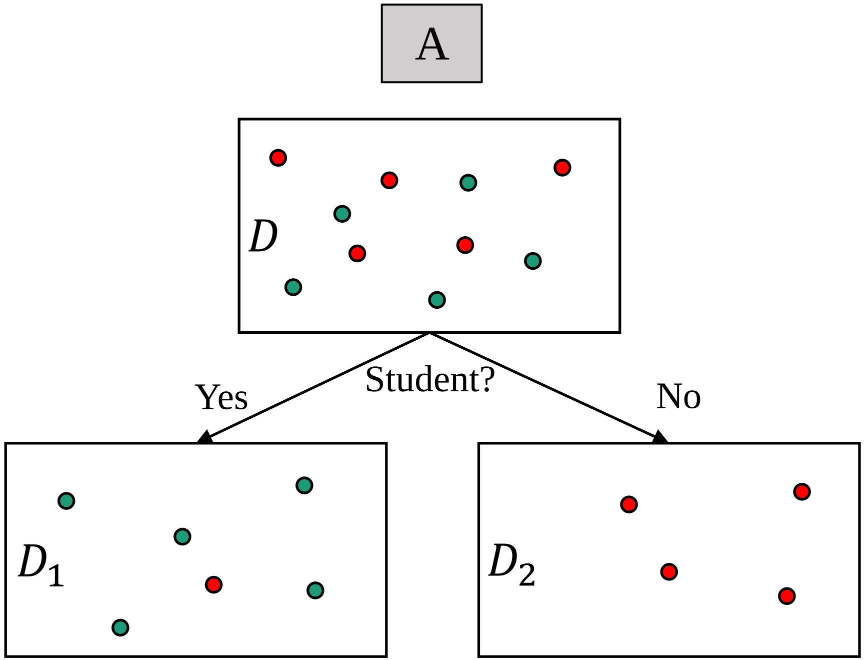


Measuring purity

$$\Delta\, Gini(A) = Gini\,(D) \;-\; Gini_A(D)$$

$$Gini_A(D) = \frac{|D_1|}{|D|} Gini(D_1) + \frac{|D_2|}{|D|} Gini(D_2)$$

$G(D) = 0.5$
$G(D_1) = 0.22$
$G(D_2) = 0$



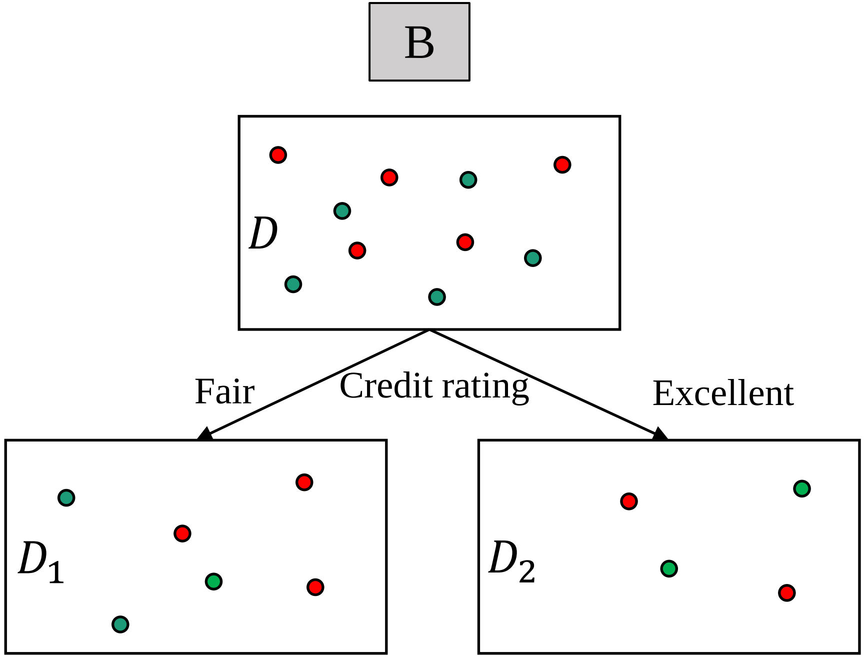
Measuring purity

$$G(\mathrm{D}) = 1 - \sum_{i=1}^{C} p_i^2 \qquad p_i = \frac{|C_{i,D}|}{|D|}$$

$$G(D) = 0.5$$

$$\Delta\, Gini(A) = Gini\,(D) - Gini_A(D)$$

$$Gini_A(D) = \frac{|D_1|}{|D|} Gini(D_1) + \frac{|D_2|}{|D|} Gini(D_2)$$



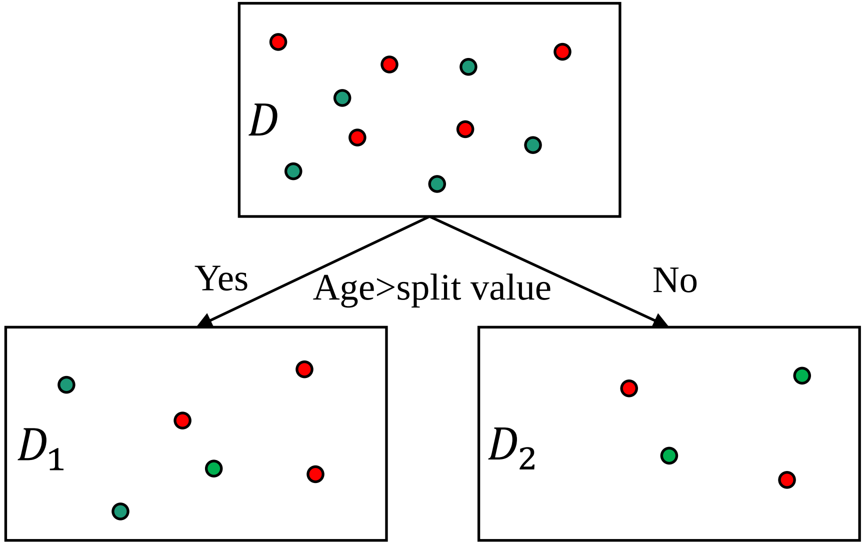
- ✓ Which feature you choose for branch to split step?

Splitting on a continuous variable

- For continuous-valued attributes, **each possible split-point must be considered**.

- The point giving the minimum Gini index for a given (continuous-valued) attribute is taken as the split-point of that attribute.

- Example:



Decision tree induction

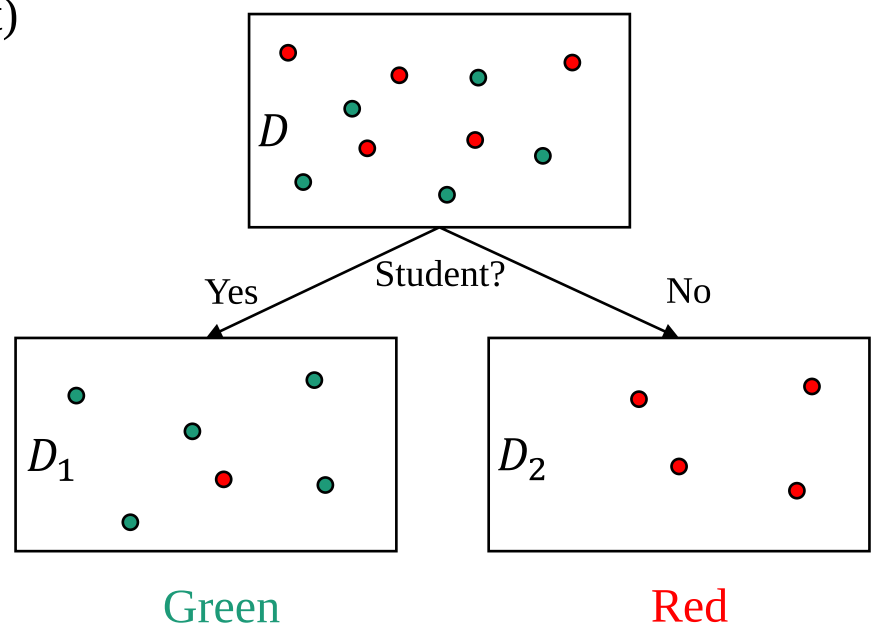
- How to choose what feature to split on at each node?(Branching)
  - ✓ **Maximize purity** (or **minimize impurity**)

- When do you stop splitting?
  - ✓ When a partition is completely pure
  - ✓ When splitting a node will result in the tree exceeding a maximum depth
  - ✓ When improvements in purity score are below a threshold
  - ✓ When number of examples in a node is below a threshold

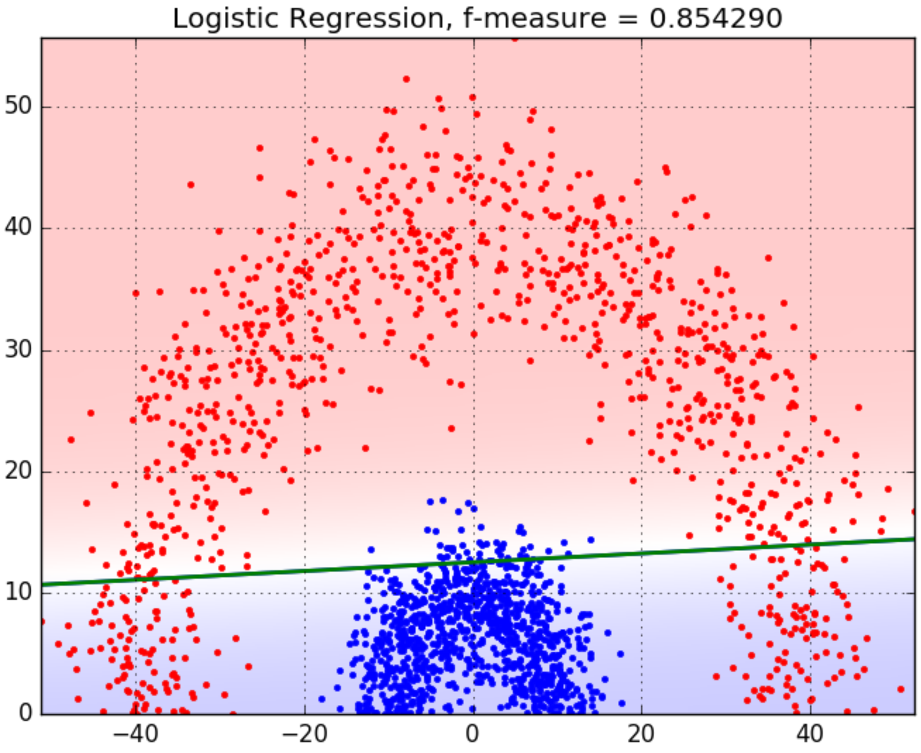
❖ The choice of purity function and stop rule is dependent on the problem.

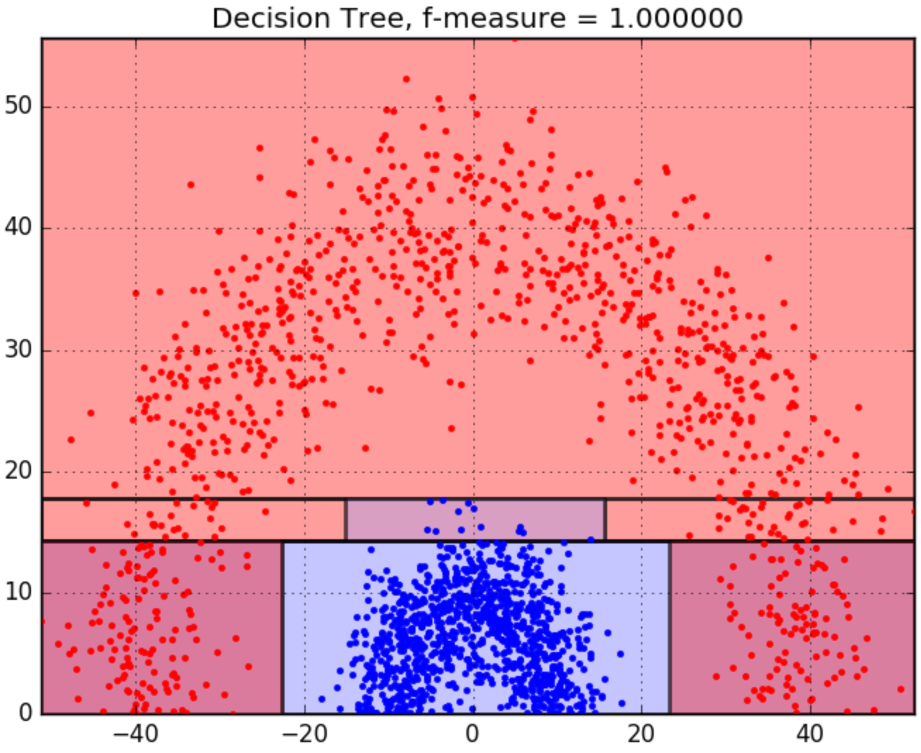
Decision tree inference

- How can we classify a new data? (Test)

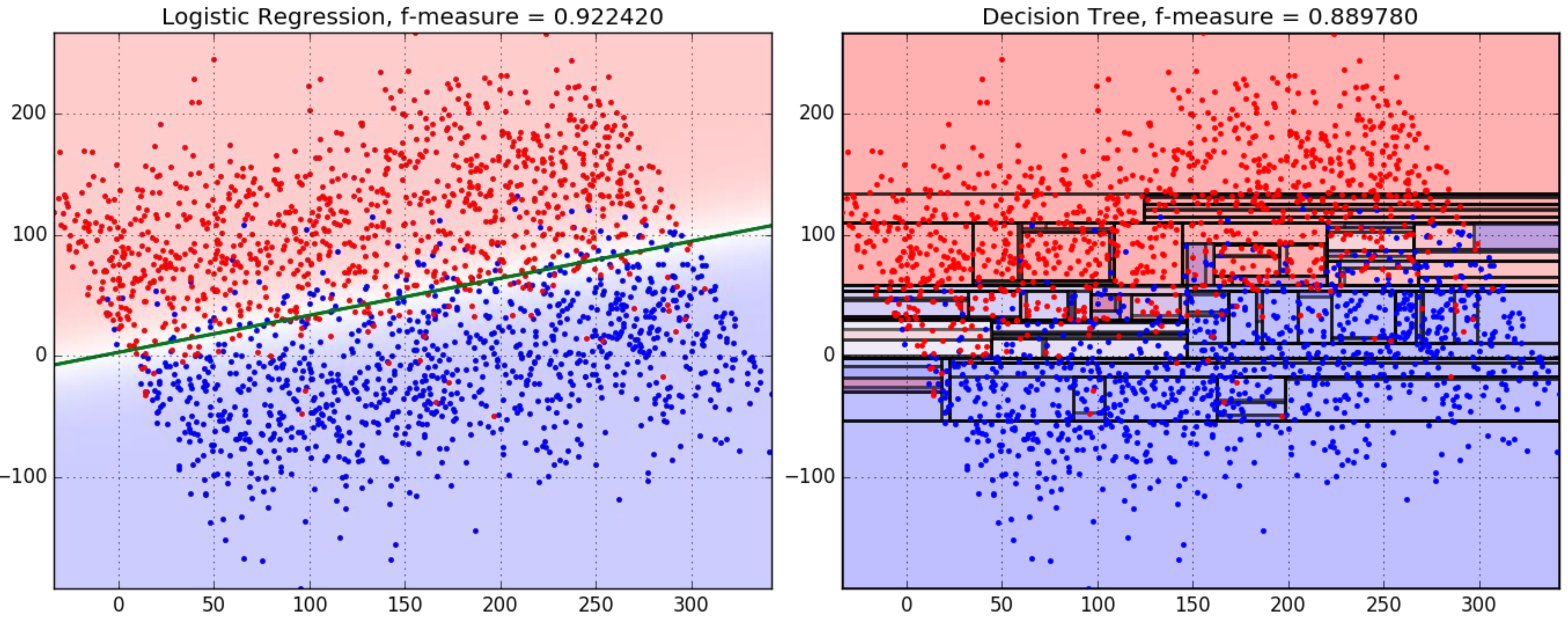


Decision tree





Decision tree



Ensemble Learning

Ensemble learning

- 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.

- **Ensemble learning** is a machine learning technique that enhances accuracy by merging predictions from multiple models.

- It aims to mitigate errors or biases that may exist in individual models by leveraging the **collective intelligence of the ensemble**.

- The individual models that we combine are known as **weak learners**. We call them weak learners because they either have a high bias or high variance.

Bagging

- **Bootstrap aggregation**, or **bagging**, is a general-purpose procedure for reducing the variance of a statistical learning method.

- Recall that given a set of n independent observations $Z_1, Z_2, \dots, Z_n$, each with variance σ^2, the variance of the mean $\bar{Z}$ of the observations is given by $\frac{\sigma^2}{n}$.

- In other words, averaging a set of observations, reduces variance.

- Of course, this is not practical because we generally do not have access to multiple training sets.

✓ What can we do?

Bagging

- Instead, we can **bootstrap**, by taking repeated samples from the (single) training data set.

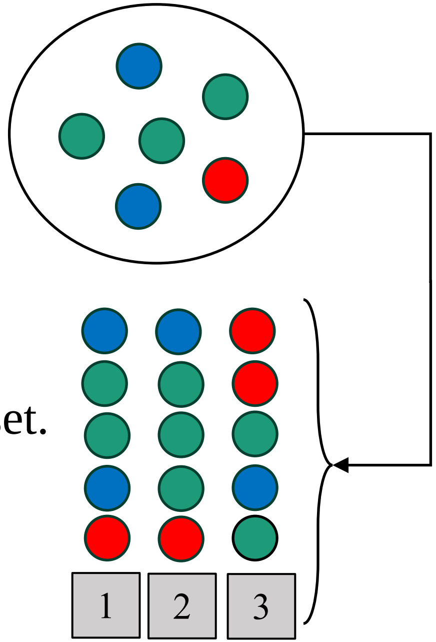
- **Bootstrap** = sampling with replacement

- Bagging steps :

1. we generate B different bootstrapped training data sets.

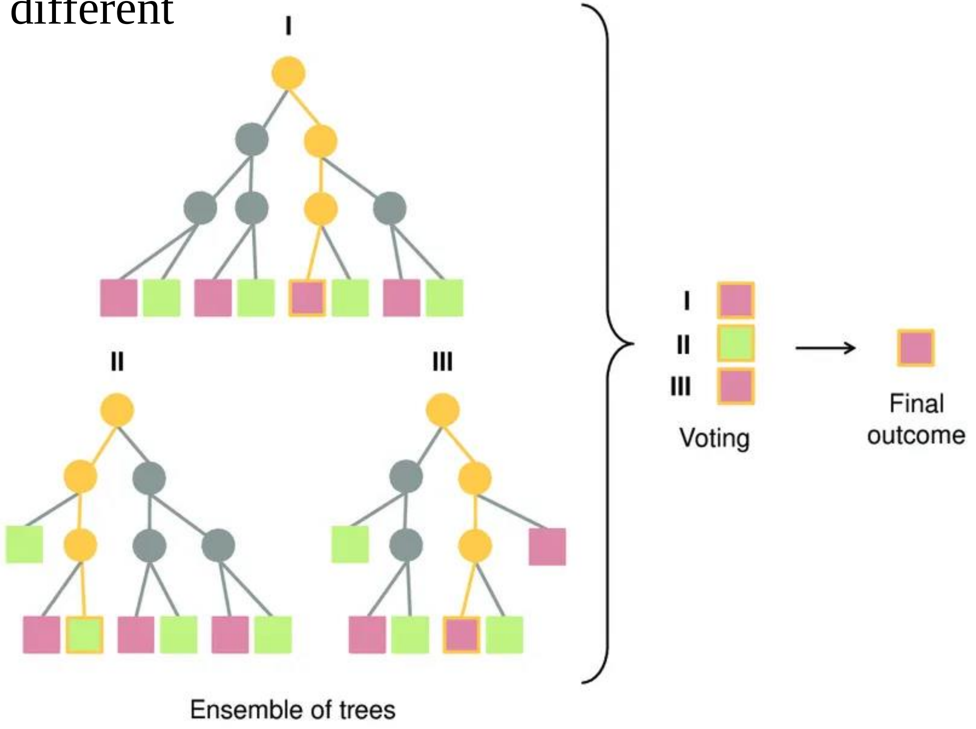
2. We then train our method on different bootstrapped training set.

3. Finally for classification, we use majority vote.



Bagging

- Example for majority voting among different trees:



Random forest

- Given training set of size n;
  For $b = 1$ to B:
  Use sampling with replacement to create a new training set of size n
  Train a decision tree on the new dataset

- For every decision tree:
  At each node, when choosing a feature to use to split, if p features are available, pick a random subset of $m < p$ features and allow the algorithm to only choose from that subset of features.

A random selection of k predictors

Random forest

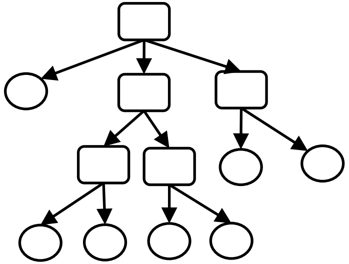
- For our example:
  $B = 2$, $m = 3$

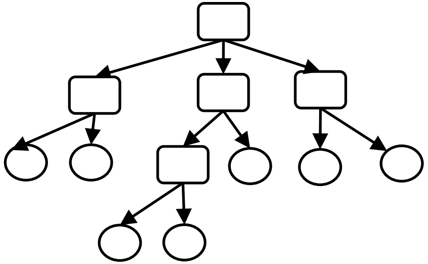
index	age	income	student	credit rating	buys computer
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle-aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle-aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle-aged	medium	no	excellent	yes
13	middle-aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

Random forest

index	income	student	credit rating	buys computer
1	high	no	fair	no
2	high	no	excellent	no
3	high	no	fair	yes
4	medium	no	fair	yes
5	low	yes	fair	yes
2	high	no	excellent	no
7	low	yes	excellent	yes
8	medium	no	fair	no
9	low	yes	fair	yes
4	medium	no	fair	yes
11	medium	yes	excellent	yes
12	medium	no	excellent	yes
13	high	yes	fair	yes
2	high	no	excellent	no

index	age	student	credit rating	buys computer
5	senior	yes	fair	yes
2	youth	no	excellent	no
3	middle-aged	no	fair	yes
10	senior	yes	fair	yes
5	senior	yes	fair	yes
6	senior	yes	excellent	no
7	middle-aged	yes	excellent	yes
8	youth	no	fair	no
9	youth	yes	fair	yes
10	senior	yes	fair	yes
11	youth	yes	excellent	yes
12	middle-aged	no	excellent	yes
12	middle-aged	no	excellent	yes
14	senior	no	excellent	no



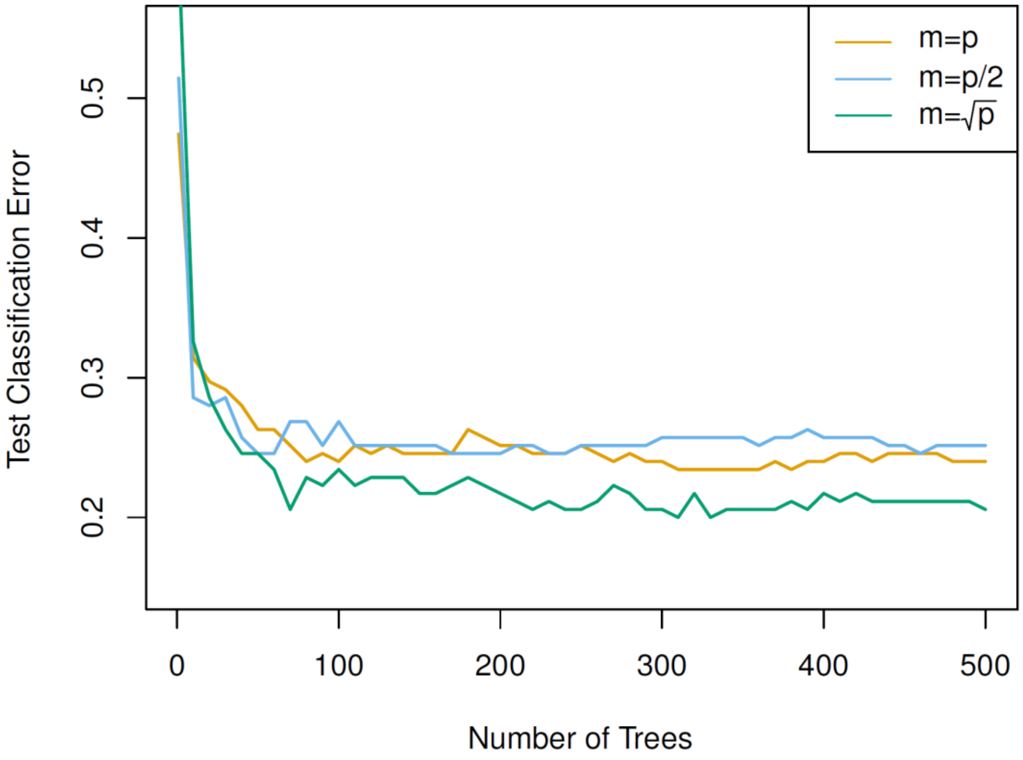


Random forest

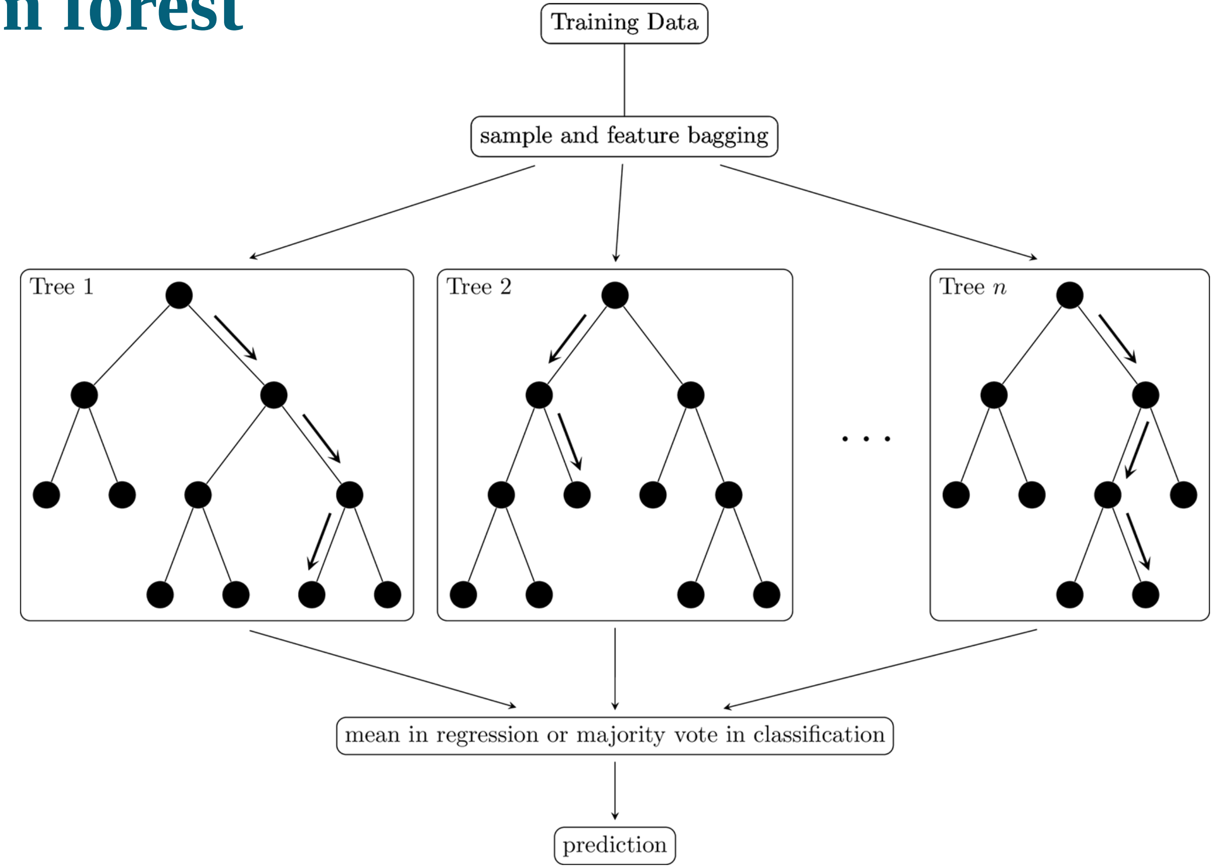
- Random forests provide an improvement over bagged trees by way of a small tweak that decorrelates the trees.

✓ What is this tweak?

- typically, we choose $m = \sqrt{p}$



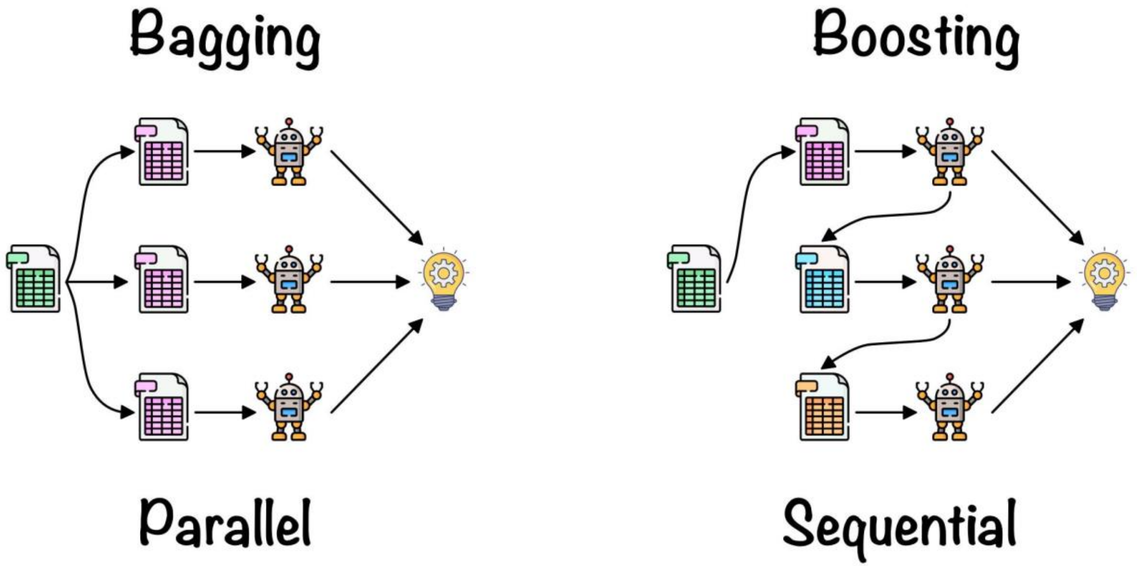
Random forest



Boosting

- **Boosting** works in a similar way, except that the trees are grown sequentially.

- Each tree is grown using information from previously grown trees.



Boosted trees intuition

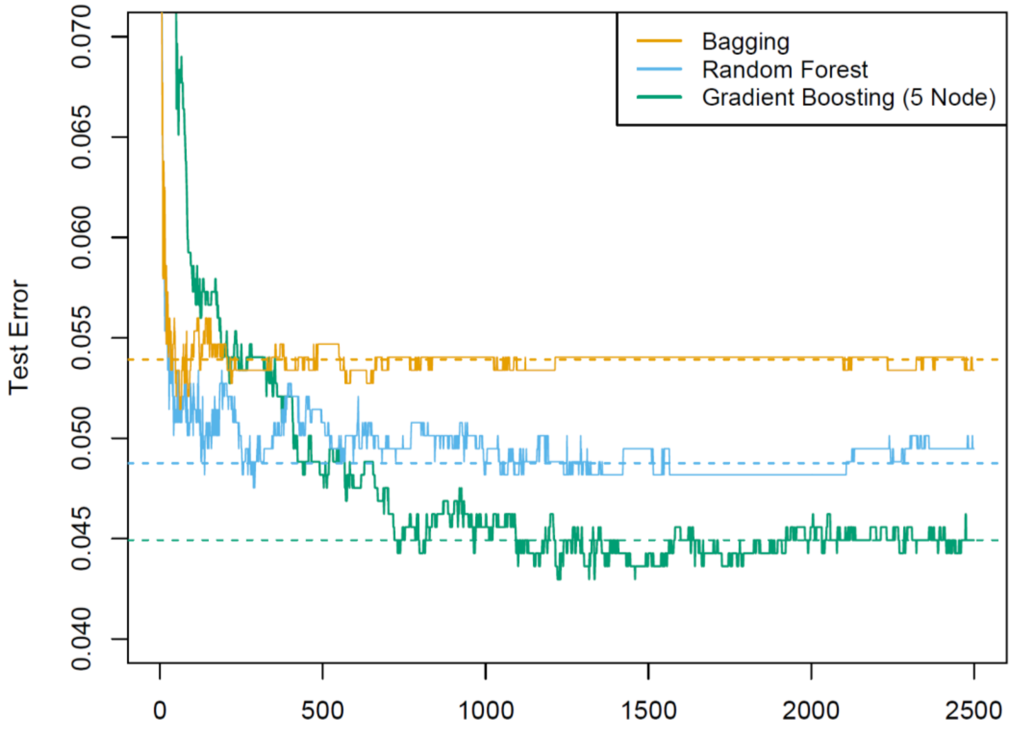
- Given training set of size n;
  For $b = 1$ to B:
    Use sampling with replacement to create a new training set of size n
    Train a decision tree on the new dataset
    But instead of picking from all examples with equal (1/n) probability, make it more likely to pick examples that the previously trained trees misclassify

focusing on the error made by the previous model.

❖ E.g., XGBoost (eXtreme Gradient Boosting)

Bagging vs Random forest vs Boosting

- Spam email classification:



Number of trees