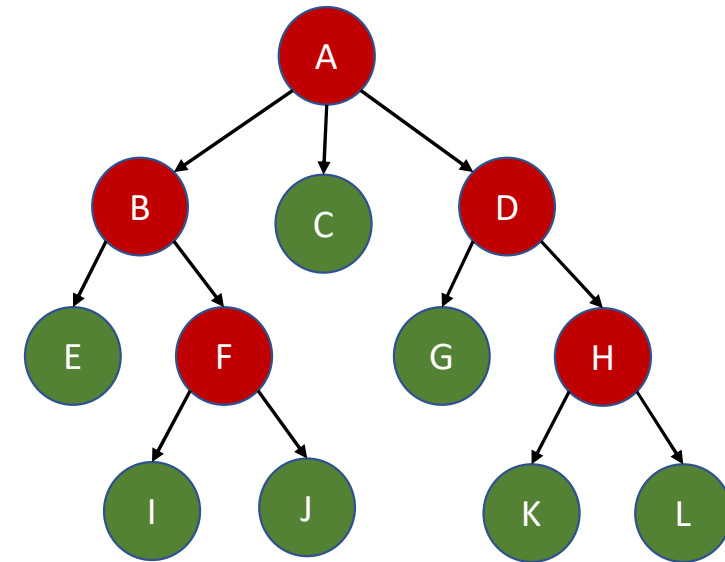


Decision Trees Learning

Decision Trees

- Decision Trees can be represented by logical formulas with
 - Each internal node testing an attribute
 - Each branch corresponding to the attribute value
 - Each leaf node assigning a classification

1. $A \leftarrow$ the “best” decision attribute for the node.
2. Assign A as decision attribute for node.
3. For each value of A, create a new descendant of node.
4. Sort training examples to leaf nodes.
5. If training examples are perfectly classified, stop.
Else, recurse over new nodes.

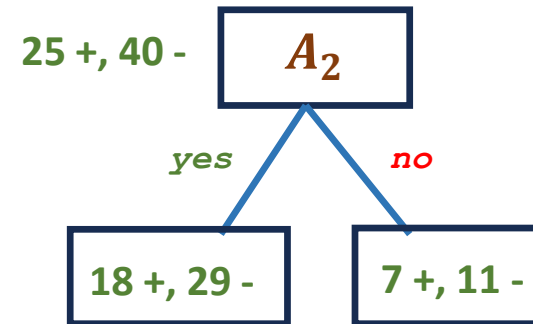
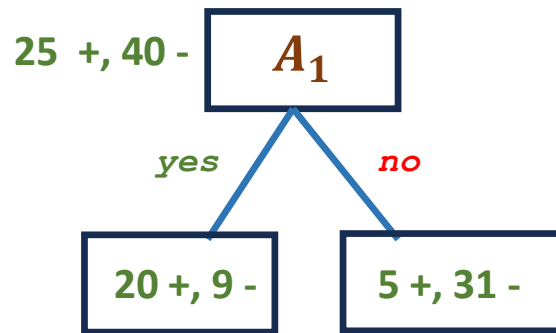


✓ How do we choose the best attribute?

Choosing Best Attribute?

- Consider 65 samples, with 25 belonging to class + and 40 belonging to class -

Which one is better?

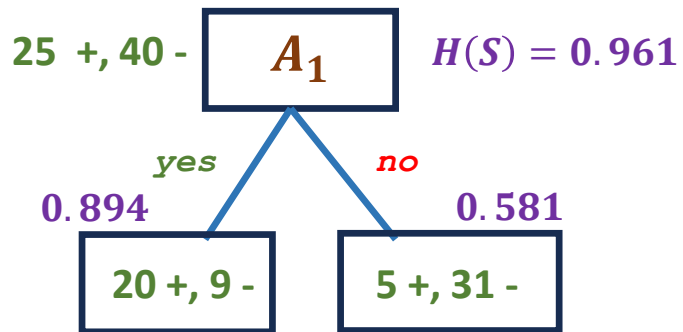


- Can be determined using Information Gain.
- Calculate Entropy first, which is a measurement of the uncertainty

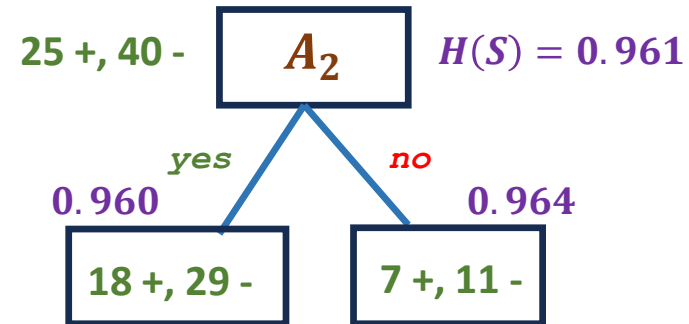
$$H(S) = -P_+ \log_2 P_+ - P_- \log_2 P_- \text{ (for binary classes)}$$

Choosing Best Attribute?

$$H(S) = -P_+ \log_2 P_+ - P_- \log_2 P_-$$



✓ Gain = $0.961 - \frac{29}{65} \times 0.894 - \frac{36}{65} \times 0.581$
 $= 0.240$



Gain = $0.9612 - \frac{47}{65} \times 0.960 - \frac{18}{65} \times 0.964$
 $= 0.000$

- Information Gain: It is the measurement of changes in entropy after the segmentation of a dataset based on an attribute

$$IG(S, F) = H(S) - \sum_{f \in F} \frac{|S_f|}{|S|} H(S_f)$$

Information Gain

- Information Gain is a measure of how much you can reduce uncertainty
- Value lies between 0 and 1
- Gain of 0 - example which have 50/50 split of +/- both before and after discriminating on attributes values
- Gain of 1 - example of going from “perfect uncertainty” to perfect certainty after splitting example with predictive attribute

Training Example

$$P(Yes) = \frac{9}{14} = 0.64,$$

$$P(No) = \frac{5}{14} = 0.36$$

$$\begin{aligned} & \text{Entropy}[H(Play)] \\ &= -(0.36 \log_2 0.36) - (0.64 \log_2 0.64) \\ &= 0.94 \end{aligned}$$

Outlook	Temp	Humidity	Windy	Play
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes
Rainy	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Sunny	Mild	High	True	No

Attribute Selection

- Split into different attributes : **Outlook, Temp, Humidity, Windy**
 - Compute the entropy for each attribute
 - Subtract from the entropy of the target to obtain information gain


		Play		
		Yes	No	
Outlook	Sunny	3	2	5
	Overcast	4	0	4
	Rainy	2	3	5
				14

$$IG(S, F) = H(S) - \sum_{f \in F} \frac{|S_f|}{|S|} H(S_f)$$

$$\begin{aligned}
 E(\text{Play}, \text{Outlook}) &= P(\text{Sunny}) \times H(3, 2) + P(\text{Overcast}) \times H(4, 0) + P(\text{Rainy}) \times H(2, 3) \\
 &= \frac{5}{14} \times 0.971 + \frac{4}{14} \times 0 + \frac{5}{14} \times 0.971 = 0.693
 \end{aligned}$$

$$\text{Information Gain}(\text{Outlook}) = 0.940 - 0.693 = 0.247$$

Attribute Selection

		Play	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3
Information Gain = 0.247			

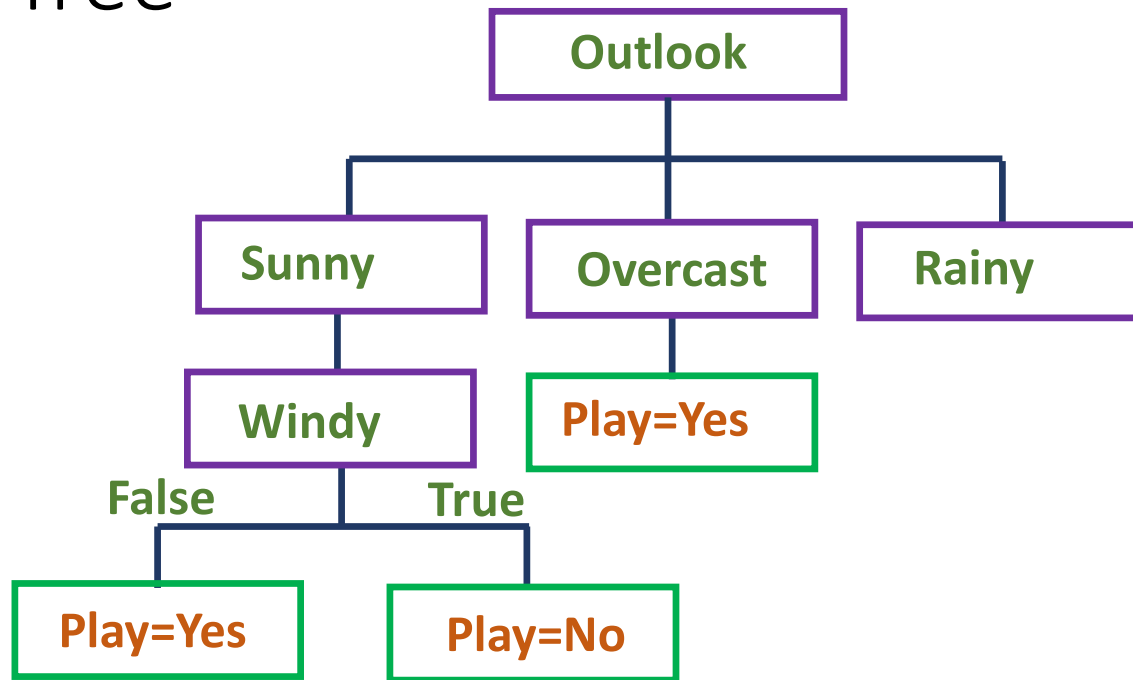
		Play	
		Yes	No
Temp.	Hot	2	2
	Mild	4	2
	Cool	3	1
Information Gain = 0.029			

		Play	
		Yes	No
Humidity	High	3	4
	Normal	6	1
Information Gain = 0.152			

		Play	
		Yes	No
Windy	False	6	2
	True	3	3
Information Gain = 0.048			

Outlook	Play
Rainy	No
Rainy	No
Rainy	No
Rainy	Yes
Rainy	Yes
Sunny	No
Sunny	No
Sunny	Yes
Sunny	Yes
Sunny	Yes
Overcast	Yes
Overcast	Yes
Overcast	Yes
Overcast	Yes

Decision Tree



- Run the algorithm recursively on the non-leaf branches, until all data is classified.

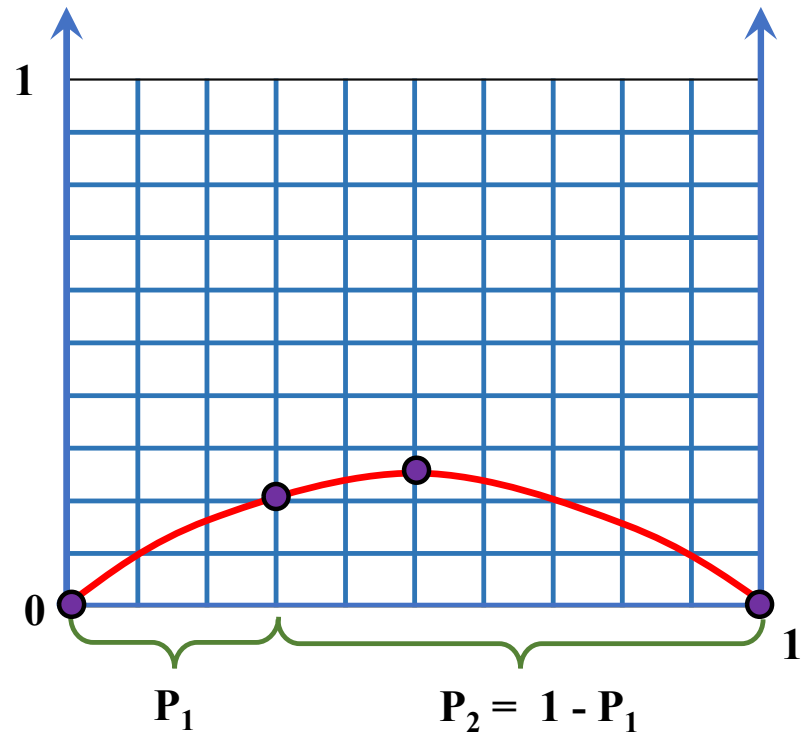
Variance Impurity

- Consider the case of only two-class
 - We need zero impurity with only one class present.

- A possible metric:

$$P(c_1)P(c_2)$$

- Zero when either of $P(c_1)$ or $P(c_2)$ is zero.
- Maximum when $P(c_1) = P(c_2)$.



Gini Index

- Extending variance impurity to 3 classes: $P(c_1)P(c_2)P(c_3)$

$$P(c_1)P(c_2) + P(c_1)P(c_3) + P(c_2)P(c_3)$$

- Gini Impurity is the extension of variance impurity to more classes

$$\text{Gini} = \sum_{i \neq j} P(c_i)P(c_j) = 1 - \sum_{i=1}^J P^2(c_i) \quad \text{➤ Prove both the equations are similar}$$

- The Gini Index or Impurity measures the probability for a random instance being misclassified when chosen randomly
- The lower the Gini Index, the better ➡ the lower the likelihood of misclassification

Example

		Play		
		Yes	No	
Outlook	Sunny	3	2	5
	Overcast	4	0	4
	Rainy	2	3	5
				14

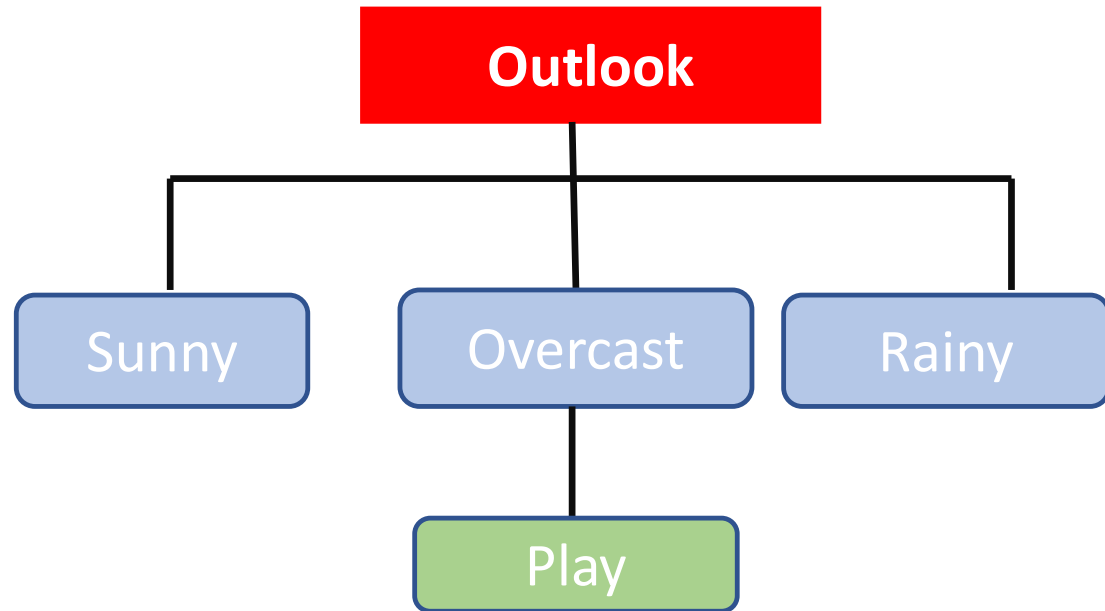
$$\begin{aligned}
 &\text{Gini(Outlook)} \\
 &= P(\text{Sunny})G(\text{Sunny}) + P(\text{Overcast})G(\text{Overcast}) \\
 &+ P(\text{Rainy})G(\text{Rainy})
 \end{aligned}$$

$$\begin{aligned}
 &= \frac{5}{14} \left(1 - \left(\frac{3}{5} \right)^2 - \left(\frac{2}{5} \right)^2 \right) + \frac{5}{14} \left(1 - \left(\frac{2}{5} \right)^2 - \left(\frac{3}{5} \right)^2 \right) \\
 &= 0.34
 \end{aligned}$$

Outlook	Temp	Humidity	Windy	Play
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes
Rainy	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Sunny	Mild	High	True	No

Gini Index

- $\text{Gini}(\text{Outlook}) = 0.34$
- $\text{Gini}(\text{Humidity}) = 0.37$
- $\text{Gini}(\text{Windy}) = 0.44$
- $\text{Gini}(\text{Temp}) = 0.43$



Misclassification Impurity

- What is the expected misclassification (error) rate at node N if we were to consider it as a leaf and assign a label?

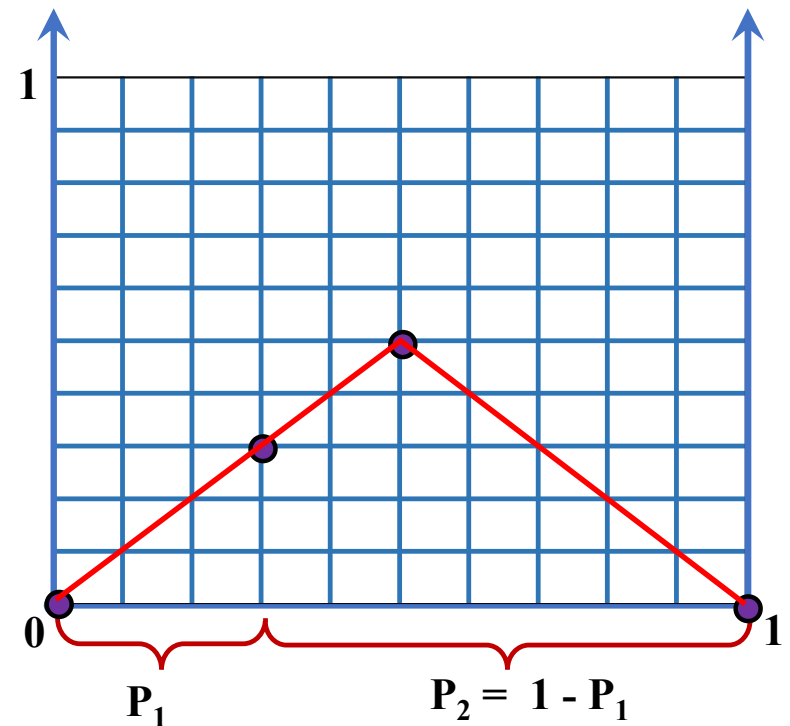
- Which label to assign?

$$\text{label} = \underset{k}{\operatorname{argmax}} P(c_k)$$

- Misclassification impurity:

$$1 - \max_j P(c_j)$$

- Strongly peaked
- Discontinuous derivative



Metric for Deciding the Split

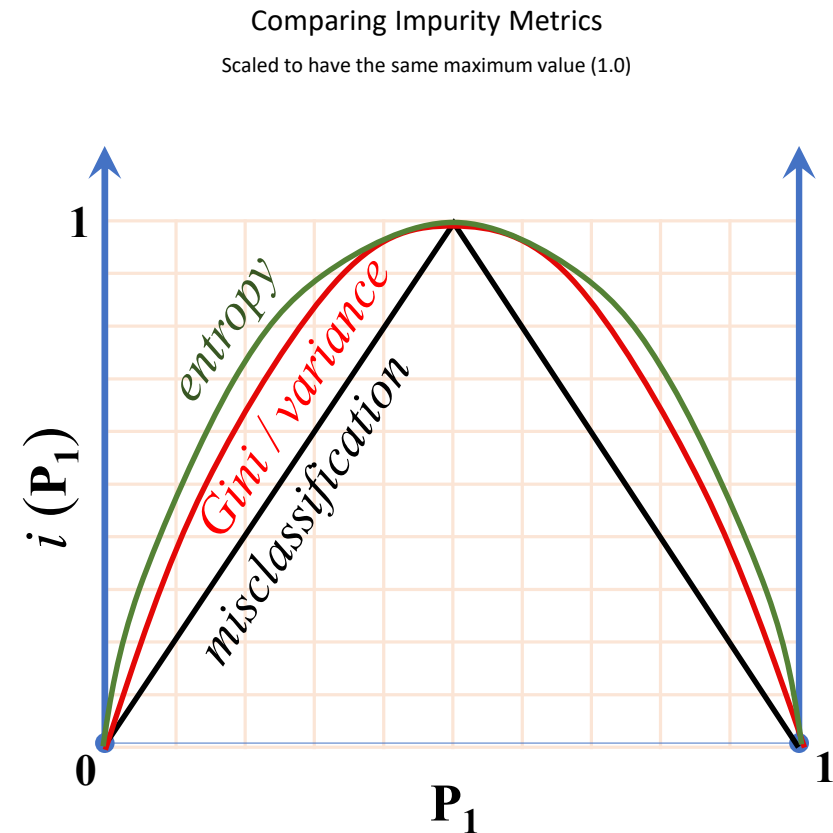
- For the best split, we want the impurity to reduce as much as possible.

$$\Delta i(T, N) = i(N) - P_L i(N_L) - P_R i(N_R),$$

where $\Delta i(T, N)$ is the change in impurity at node N when the test T is used for splitting; $P_{L/R}$ is the fraction of samples at N that moves to $N_{L/R}$ when using test T .

- Generic Case:

$$\Delta i(T, N) = i(N) - \sum_k P_k i(N_k)$$

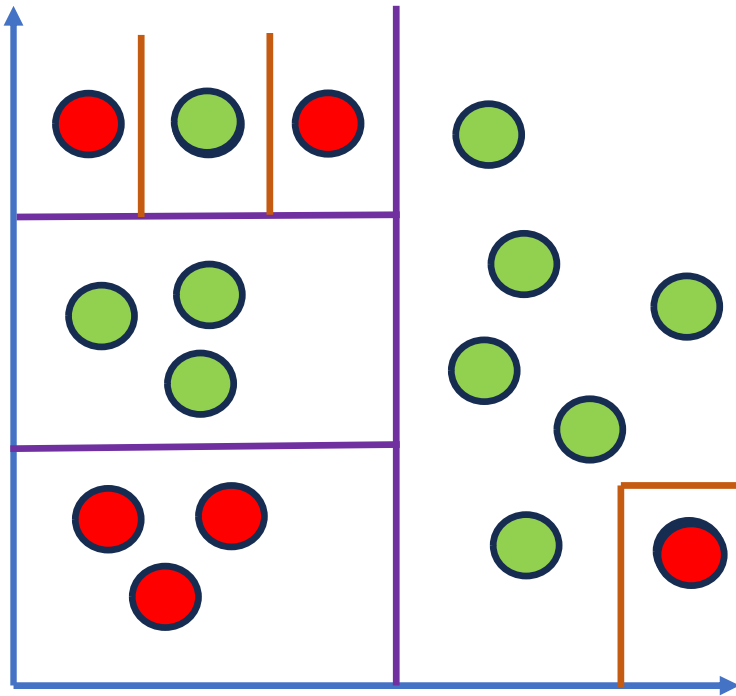


Refining Impurity Reduction Scores

- A high branching factor → greater impurity reduction
 - Need to scale impurity reduction based on branching factor, B
- Prefer balanced splits
 - Highly imbalanced splits may result in overfitting, increase model size, and reduce efficiency during classification
- Scaled criterion: **Gain Ratio Impurity**

$$\Delta i_B(N) = \frac{\Delta i(N)}{-\sum_k P_k \log(P_k)}$$

When to stop splitting?



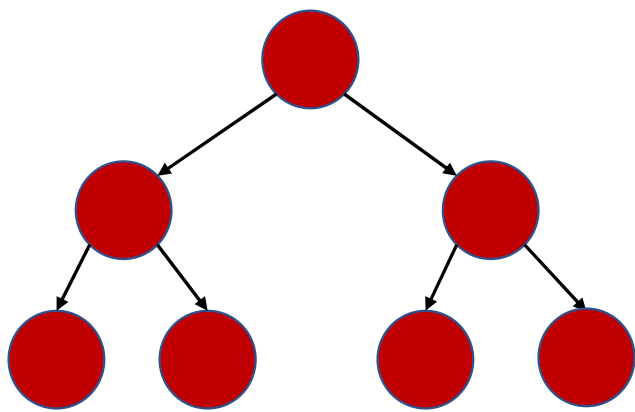
- Should you treat this as noise or go ahead with the splitting?
- Decision trees tend to "overfit" the data and hence do not provide good generalization

Solutions for Overfitting the Data

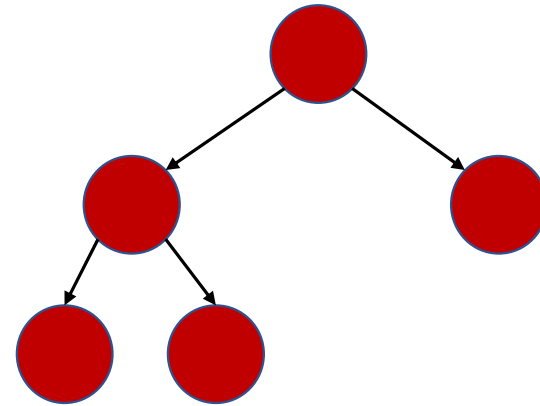
Two basic approaches

- Prepruning: Stop growing the tree at some point during construction before it begins to overfit the data
 - ✓ This solution is hard to implement in practice because it is not clear what is a good stopping point
- Postpruning: Grow the full tree and then remove nodes that seem not to have sufficient evidence.
 - ✓ This method is mostly used in the machine learning community

Decision Tree Pruning



Grow the tree to learn the training data



Prune tree to avoid overfitting the data

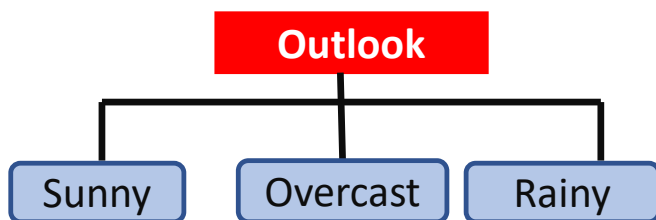
- By pruning the nodes that are far too specific to the training set, it is hoped the tree will have better generalization. In practice, we use techniques such as cross-validation and held-out training data to better calibrate the generalization properties.

CART Algorithm Structure

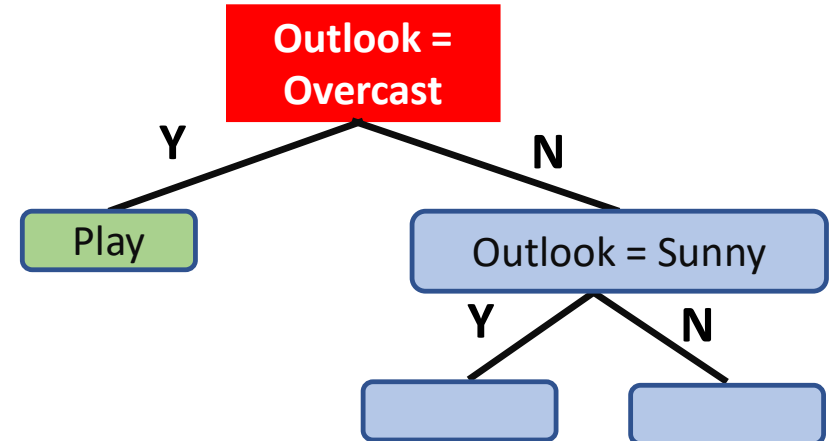
- A Classification And Regression Tree (CART), is a predictive model, explaining how an outcome variable's values can be predicted based on other values.
- A CART output is a decision tree where each fork is a split in a predictor variable and each end node contains a prediction for the outcome variable.

1. Branching Factor or Branching Ratio (B):

- Each decision or outcome is a split
- A p-way split can be expressed as multiple binary splits



- Due to the simplicity, CART focuses on binary splits



CART Algorithm Steps

2. Test at each node.

- Best Gain Ratio Impurity criterion
- Cost Function: Gini (classification), MSE (regression)

3. Deciding a node to be a leaf

- Use cross-validation
- Threshold: $\Delta i(N) < \beta$
- Threshold on number of samples (10 samples or 5%)
- Use a global criterion: $\alpha.size + \sum_{leaf} i(N)$

CART Algorithm Steps

4. Pruning a Tree

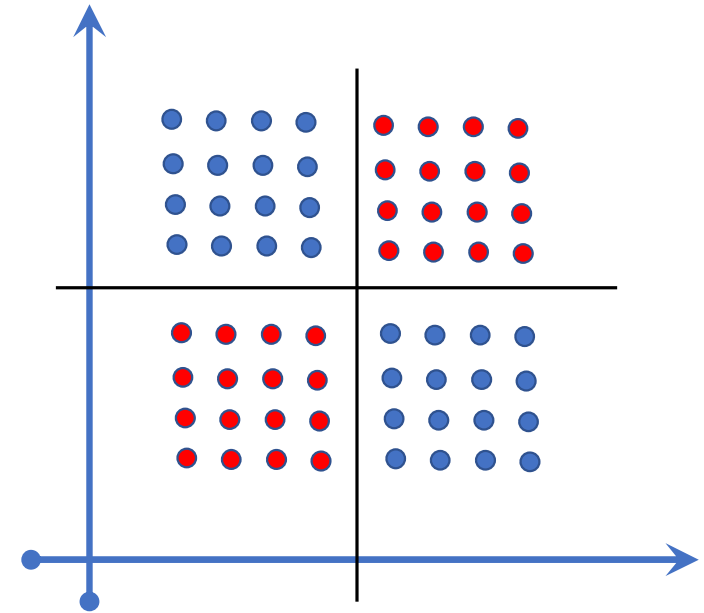
- Horizon Effect: Develop further and prune
- Remove children if results in minimum increase in impurity
- Increases classification speed; Accuracy

5. Assigning Labels to Leaves

- Assign label of most frequent sample
- Use weighted voting if priors are given

6. Handling Missing Data

- Use sample with available feature values only at a node while computing impurity



Other Algorithms

Features	CART [Classification and Regression]	ID3 [Classification] (Iterative Dichotomiser 3)	C4.5 [Classification]
Types of data	Continuous and nominal	Categorical	Continuous and Categorical
Speed	Average	Low	Faster than ID3
Pruning	Post pruning	No	Pre-Pruning
Missing Values	Can deal with	Can't deal with	Can deal with
Boosting	Supported	Not Supported	Not Supported
Formula	Gini index and MSE	Information Gain and Entropy	Split information gain ratio