Classification (Part1)

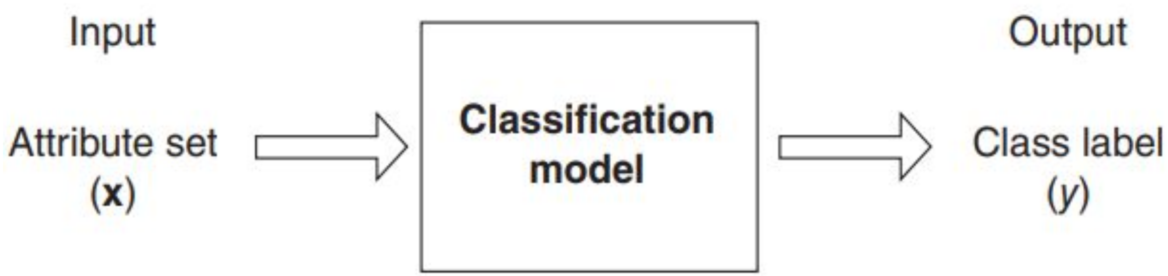
Mohammed Brahimi & Sami Belkacem

Chapter Overview

- Introduction to Classification

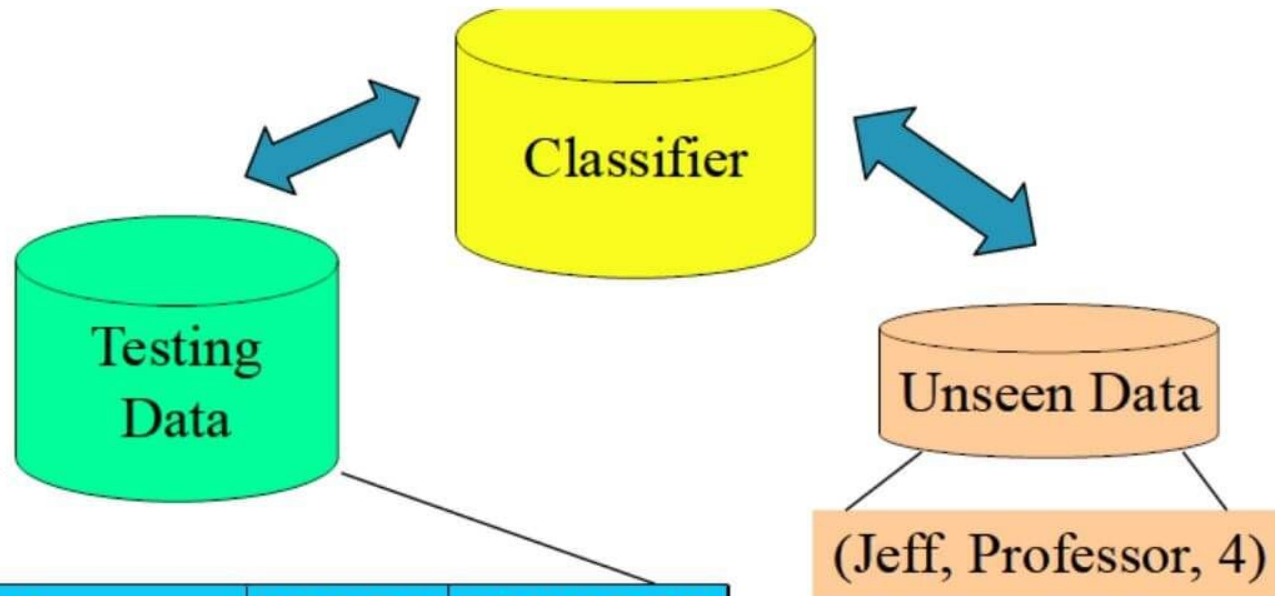
- Decision Tree Induction
  - Introduction to Decision tree
  - Attribute Test Conditions
  - Impurity Measures and Splitting Strategies
  - Gain Ratio

What is classification ?



- **Data Instances**
  - **Attributes:** Descriptive features of instances (x).
  - **Class Labels:** Categorical labels representing the class of an instance (Y).
- **Classification**
  - Assigning class labels to instances based on attributes.
- **Classifier (Model)**
  - A function **f(x)** used to classify **unseen data x** by assigning a class **label y**.

Classification example



NAME	RANK	YEARS	TENURED
Tom	Assistant Prof	2	no
Merlisa	Associate Prof	7	no
George	Professor	5	yes
Joseph	Assistant Prof	7	yes

Tenured?

Yes

Attributes **Class labels**

Applications of classification

- **Health**
  - Medical diagnosis
  - Patient risk categorization
- **Computer Security**
  - Spam filtering
  - Malware classification
- **Banking and Finance**
  - Loan default prediction
  - Credit card fraud detection
- **Retail and E-commerce**
  - Customer purchase pattern classification
  - Sentiment analysis from customer reviews
- **Transportation and Logistics**
  - Cargo classification for customs
  - Driver behavior classification for insurance

Role of Classification Models

- **Predictive Models**
  - Used to predict class labels for new, unseen data.
  - Learn patterns from historical data to make predictions.

- **Descriptive Models**
  - Help understand distinguishing features of different classes.
  - Analyzes the data to find common characteristics and patterns.

Role of Classification Models

- **Predictive Models**
  - *__Example:__* Classifying email messages as 'urgent' or 'non-urgent' based on their content.

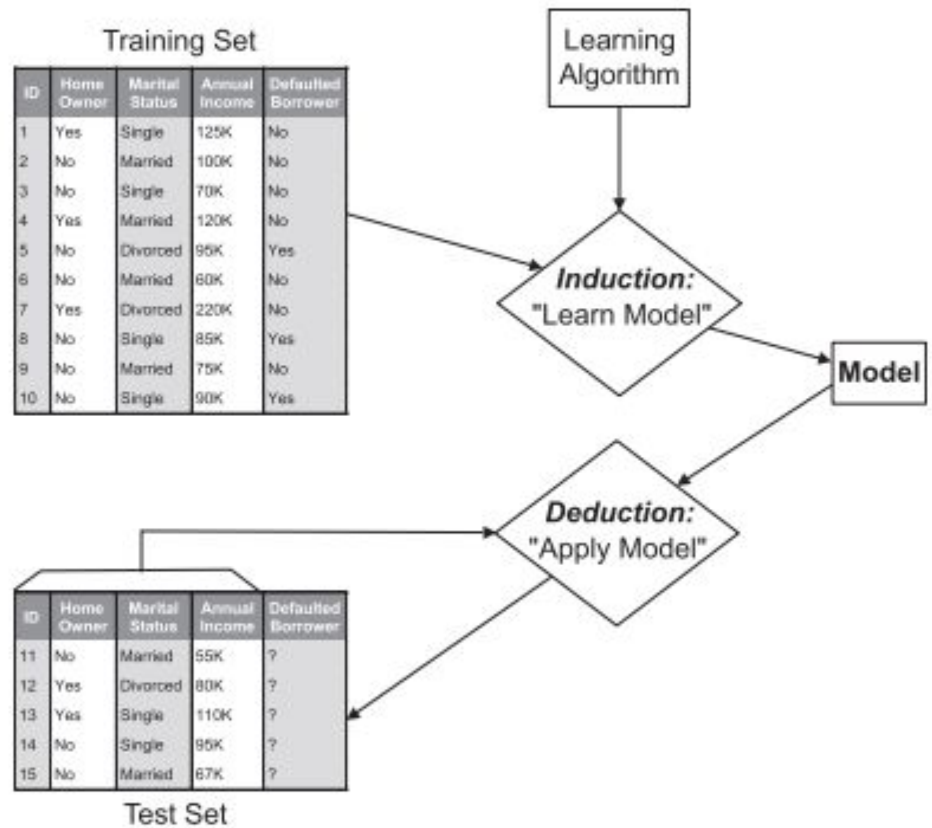
- **Descriptive Models**
  - *__Example:__* Identifying key factors that differentiate high-risk patients in healthcare.

General Framework for Classification

- **Induction (Training)**
  - Learning a model from a **labeled training dataset**.
  - Use learning algorithm to **build models**.

- **Deduction (Testing)**
  - Applying the learned model to **new instances (unseen) to predict class labels**.
  - **Assessing the model's performance** to measure its generalization capability.

The feedback from testing is often used to refine and improve the training process.



Training Set

ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Test Set

ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
11	No	Married	55K	?
12	Yes	Divorced	80K	?
13	Yes	Single	110K	?
14	No	Single	95K	?
15	No	Married	67K	?

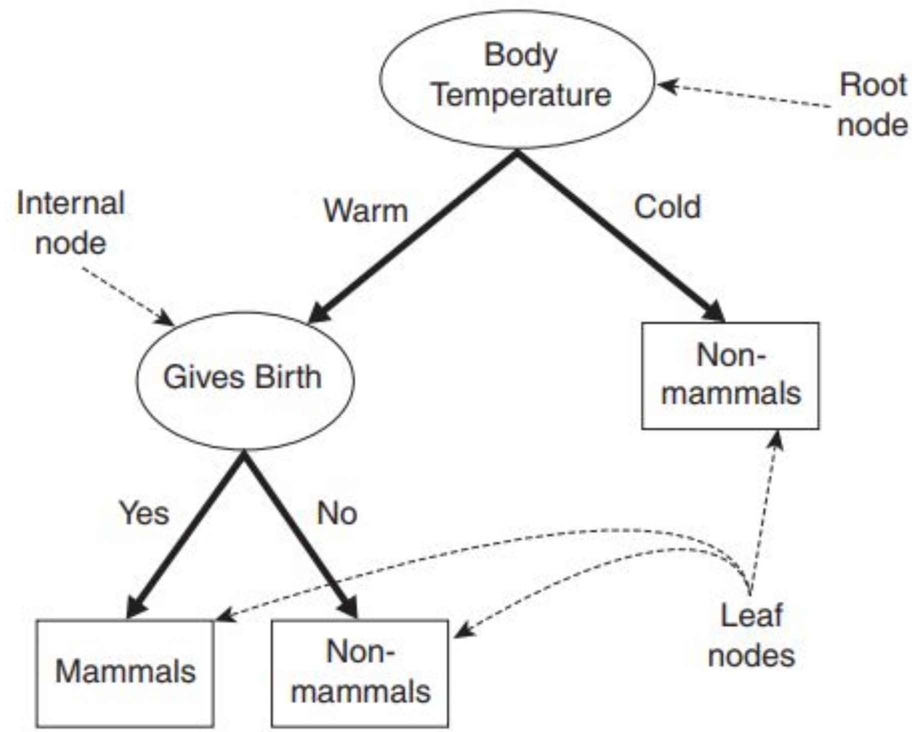
Chapter Overview

- Introduction to Classification

- Decision Tree Induction
  - Introduction to Decision Tree
  - Attribute Test Conditions
  - Impurity Measures and Splitting Strategies
  - Gain Ratio

Decision Tree

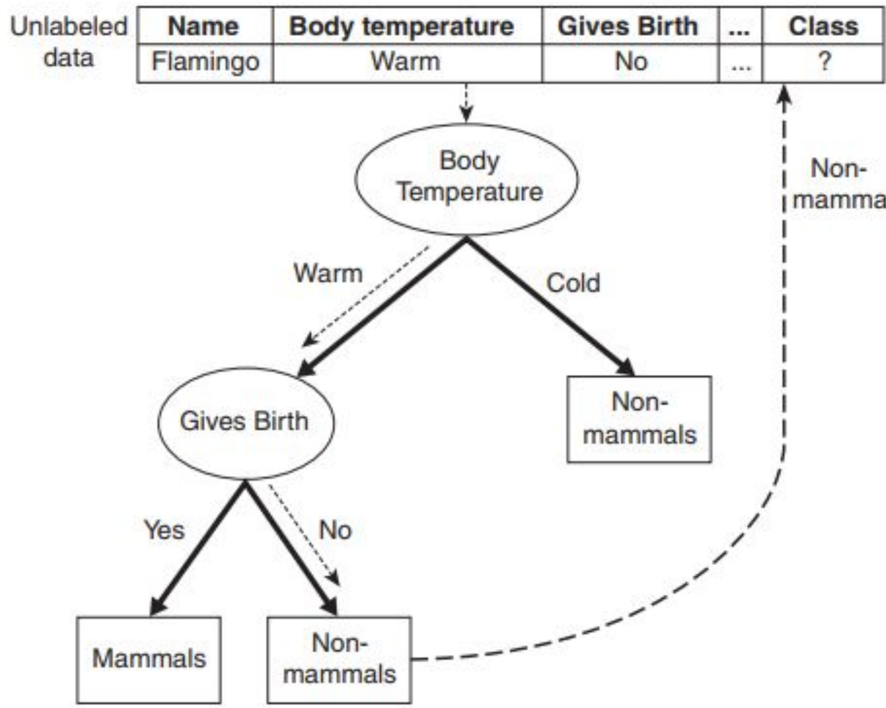
- **Root Node**
  - Initial point of decision-making.
- **Internal Nodes**
  - Question based on a single attribute
  - Attribute test.
- **Leaf Nodes**
  - The final classification outcome.



Mammal classification tree

Deduction in Decision Trees

- **Start at Root Node**
  - Apply initial attribute test.
  - Follow Test Outcome.
- **Visit next node**
  - Follow Test Outcome.
- **Reach Leaf Node**
  - Determine final classification.



Mammal classification tree

Hunt’s Algorithm for decision tree building

- **Initial Node**
  - Start with a root node containing all instances.

All instances

- **Expansion and Child Nodes Formation**
  - Expand nodes with **mixed class instances**.
  - Select the attribute test based on a **splitting criterion**.
  - Create child nodes for each test outcome
  - Distribute instances accordingly.

- **Recursive Process**
  - Continue expansion for nodes with mixed class instances.

- **Termination**
  - Stop when a node has instances of only one class.

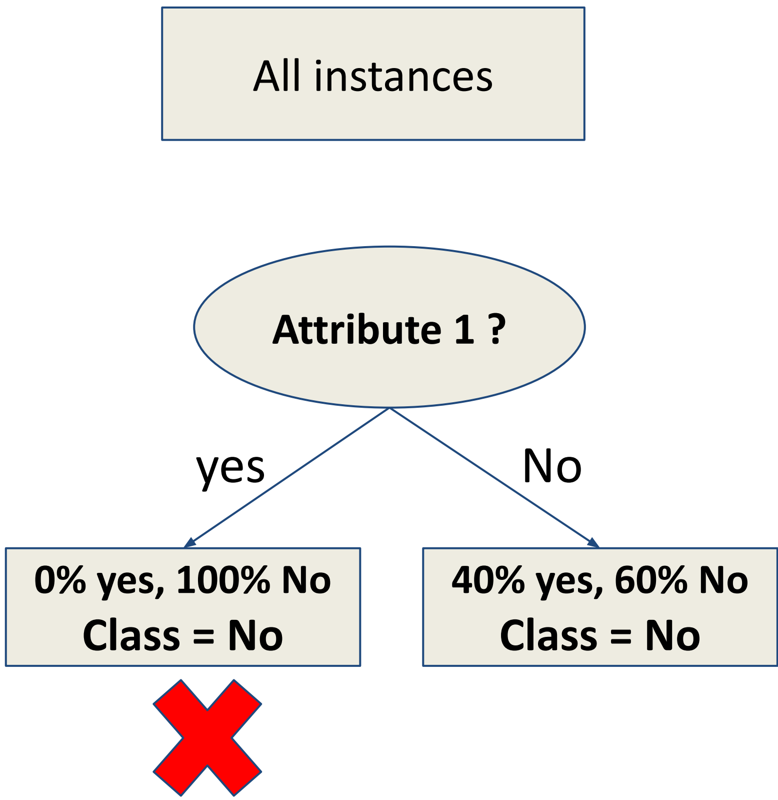
Hunt’s Algorithm for decision tree building

- **Initial Node**
  - Start with a root node containing all instances.

- **Expansion and Child Nodes Formation**
  - Expand nodes with **mixed class instances**.
  - Select the attribute test based on a **splitting criterion**.
  - Create child nodes for each test outcome
  - Distribute instances accordingly.

- **Recursive Process**
  - Continue expansion for nodes with mixed class instances.

- **Termination**
  - Stop when a node has instances of only one class.



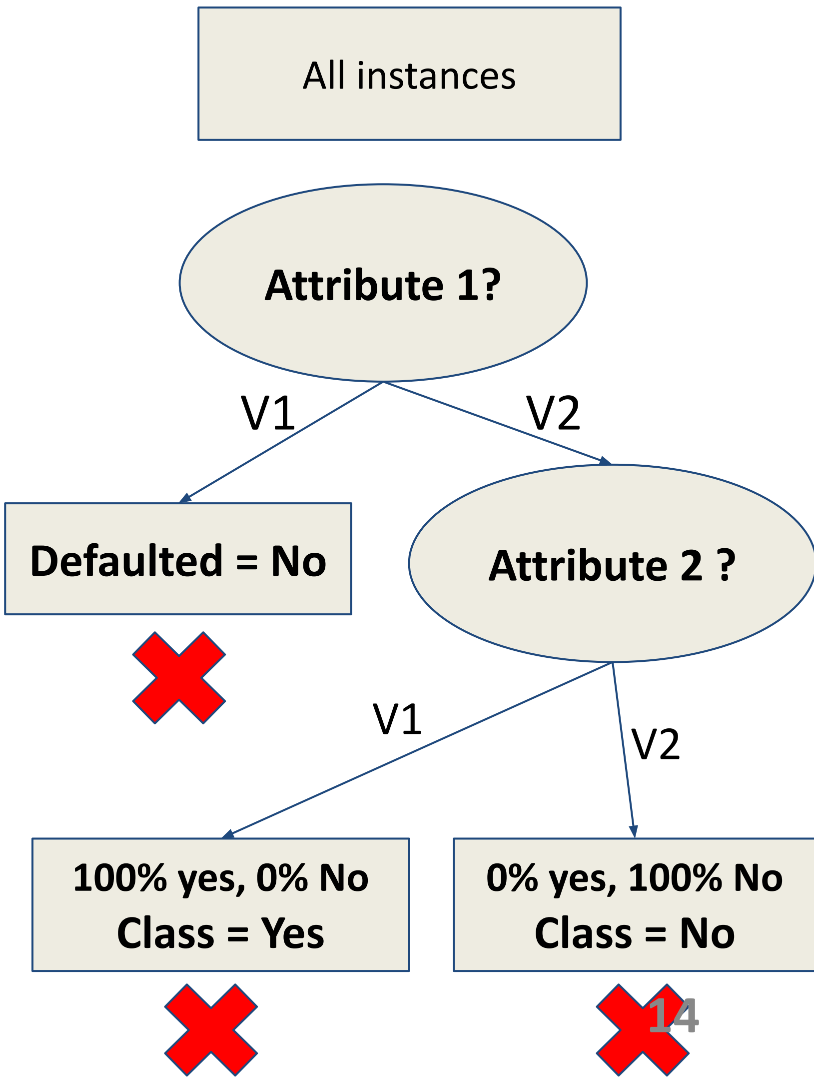
Hunt's Algorithm for decision tree building

- **Initial Node**
  - Start with a root node containing all instances.

- **Expansion and Child Nodes Formation**
  - Expand nodes with **mixed class instances**.
  - Select the attribute test based on a **splitting criterion**.
  - Create child nodes for each test outcome
  - Distribute instances accordingly.

- **Recursive Process**
  - Continue expansion for nodes with mixed class instances.

- **Termination**
  - Stop when a node has instances of only one class.



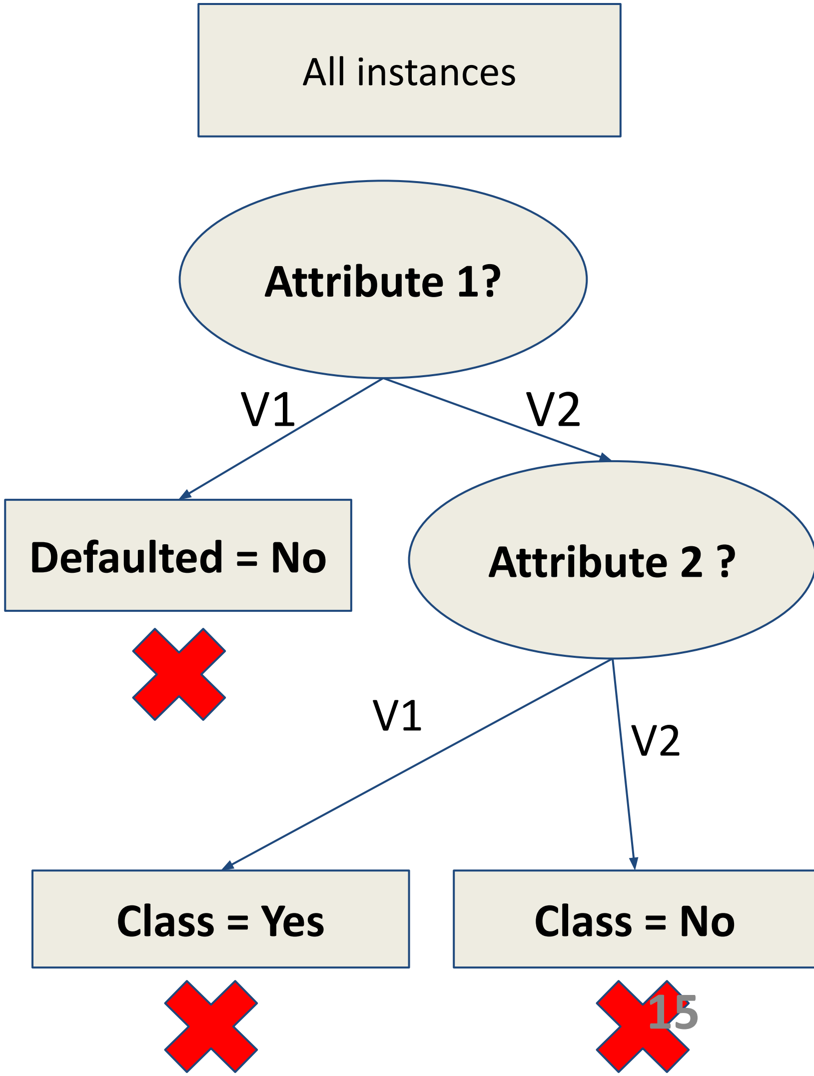
Hunt’s Algorithm for decision tree building

- **Initial Node**
  - Start with a root node containing all instances.

- **Expansion and Child Nodes Formation**
  - Expand nodes with **mixed class instances**.
  - Select the attribute test based on a **splitting criterion**.
  - Create child nodes for each test outcome
  - Distribute instances accordingly.

- **Recursive Process**
  - Continue expansion for nodes with mixed class instances.

- **Termination**
  - Stop when a node has instances of only one class.



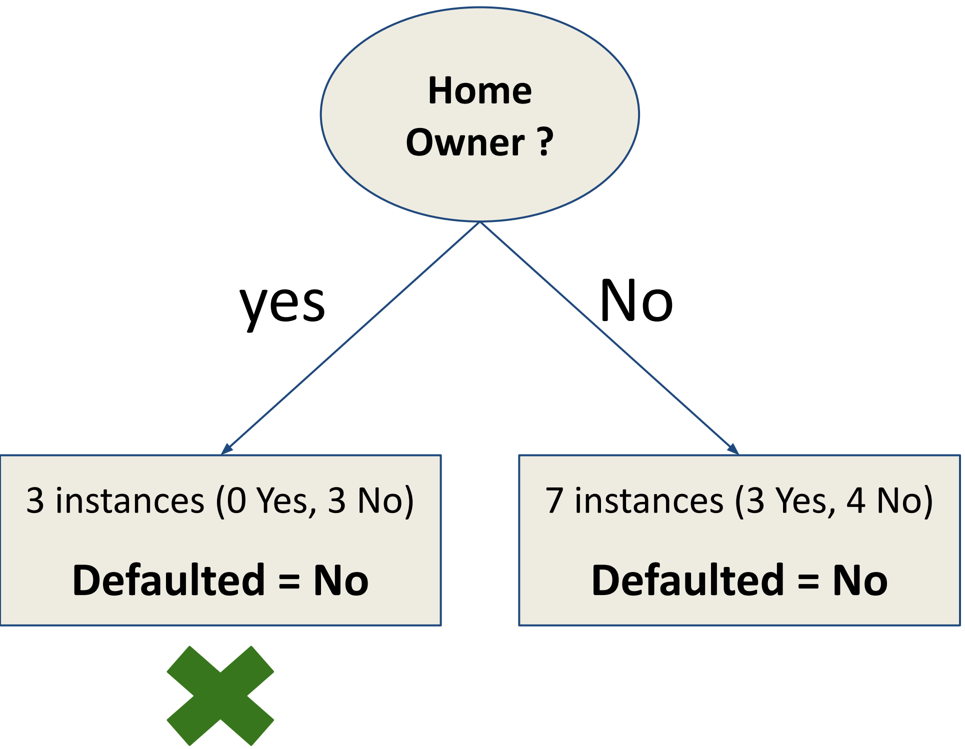
Hunt's Algorithm: Example

10 instances (3 Yes, 7 No)

Defaulted = No

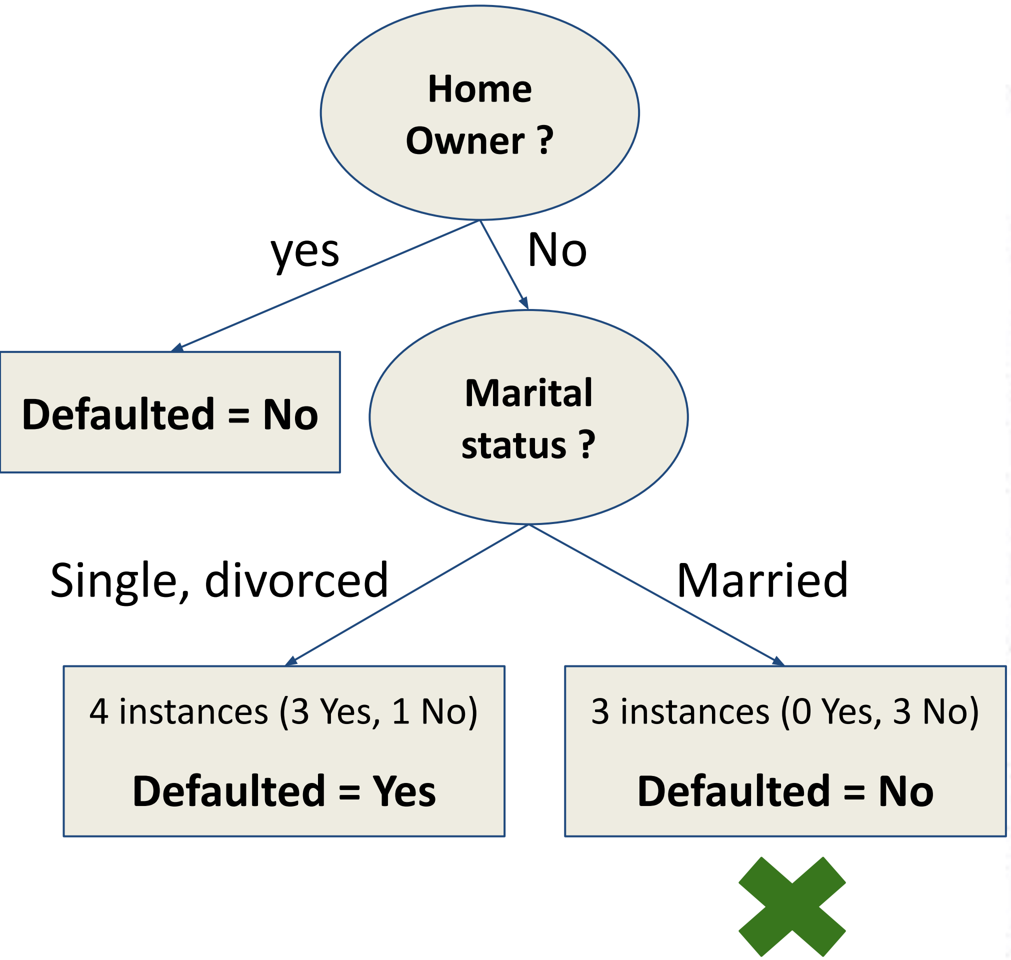
ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Hunt’s Algorithm: Example



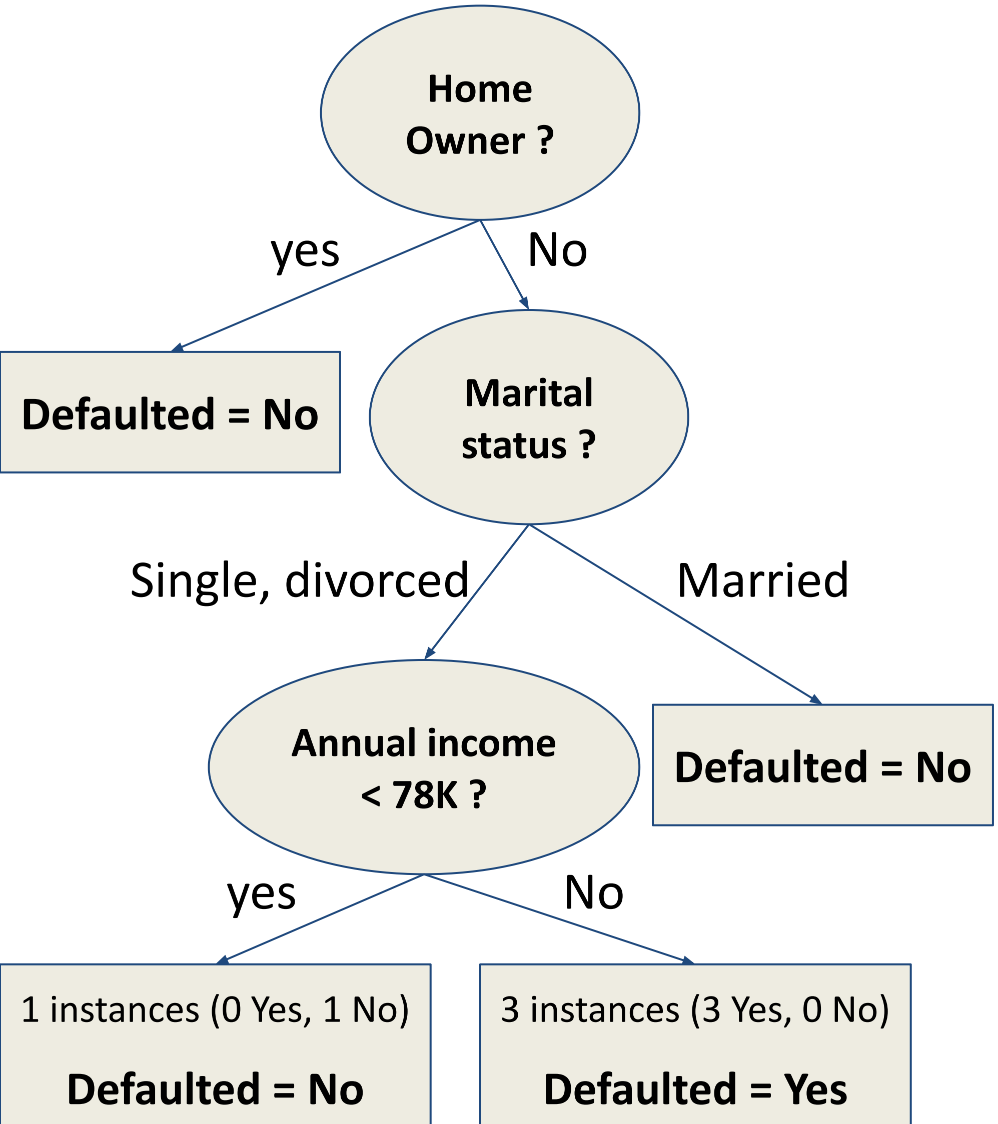
ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Hunt’s Algorithm: Example



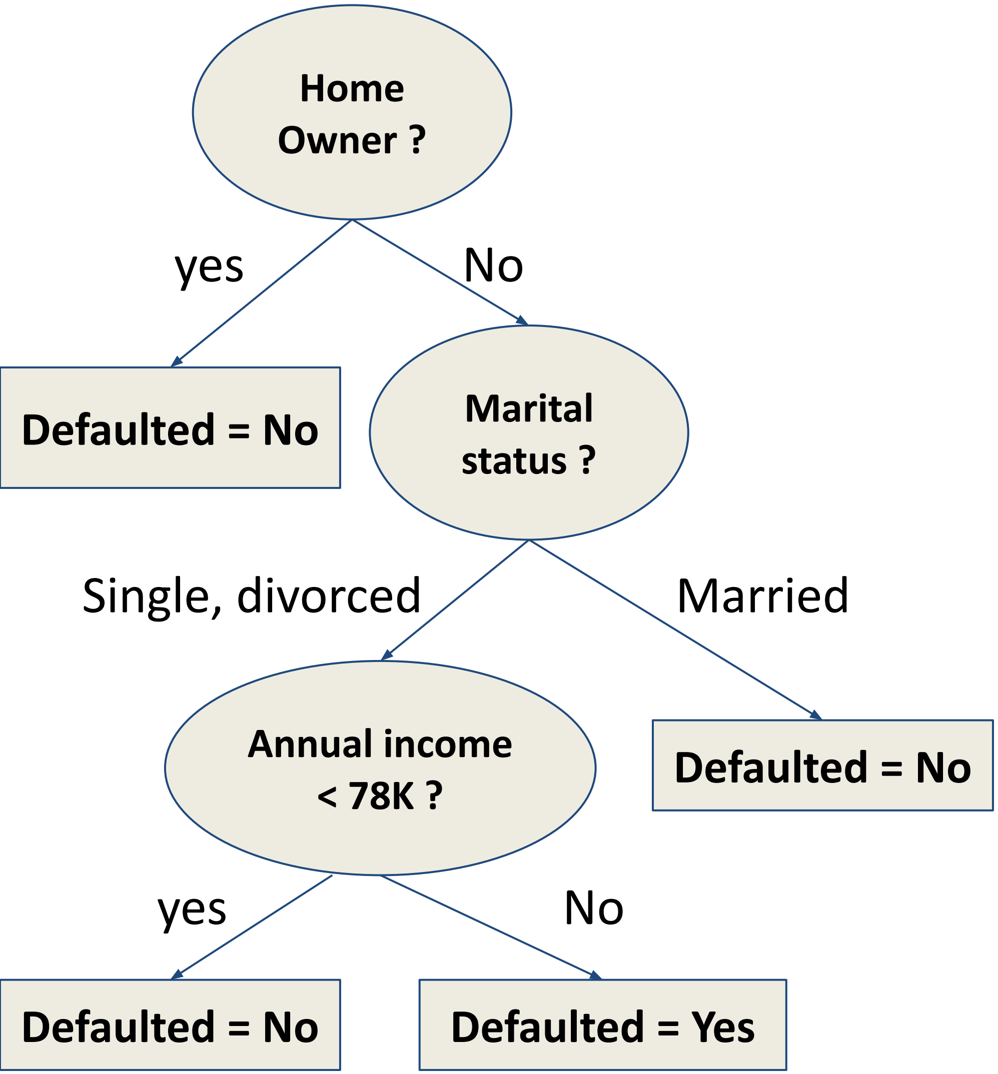
ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	[redacted]			
2	No	Married	100K	No
3	No	Single	70K	No
4	[redacted]			
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	[redacted]			
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Hunt’s Algorithm: Example



ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	[redacted]			
2	[redacted]			
3	No	Single	70K	No
4	[redacted]			
5	No	Divorced	95K	Yes
6	[redacted]			
7	[redacted]			
8	No	Single	85K	Yes
9	[redacted]			
10	No	Single	90K	Yes

Hunt's Algorithm: Example



ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Decision tree algorithm questions

- **How to handle an empty test outcome?**
- **All attributes values are identical BUT different class labels?**
- **How to determine the best attribute test?**
- **What are the stopping criteria for the algorithm?**

How to handle an empty test outcome?

When Does This Occur?

- No training instances with specific attribute values.
- These attribute values can happen in testing instances.

Approach

Assign the most common class label from parent node to empty nodes.

All attributes values are identical BUT different class labels?

When Does This Occur?

- Expansion is impossible.
- We can’t build leaves contains the same class.

Approach

Declare it a leaf node and assign it the most common class label in the training instances associated with this node.

Chapter Overview

- Introduction to Classification
- Decision Tree Induction
  - Introduction to Decision Tree
  - Attribute Test Conditions
  - Impurity Measures and Splitting Strategies
  - Gain Ratio

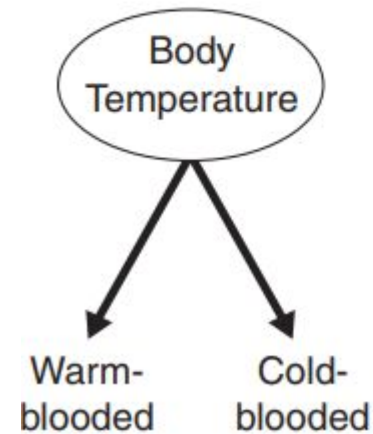
How to determine the best attribute test?

How to split?
Splitting criterion

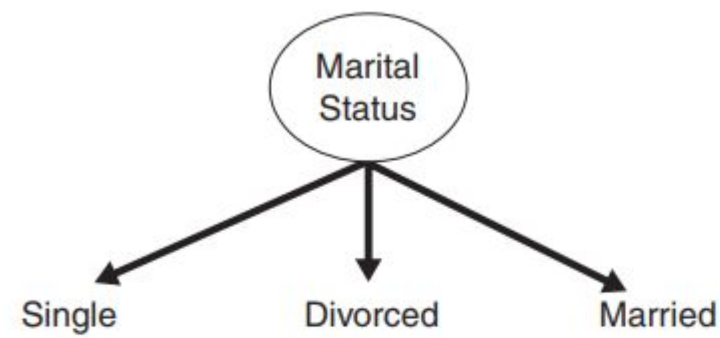
Attribute Test Conditions & Attribute type

- **Binary Attributes**
  - **Outcomes:** True or False.
  - ***Binary Split:*** Two outcomes

- **Nominal Attributes**
  - ***Multiway Split:*** More than two possible outcomes.
  - ***Binary Split:*** Two outcomes

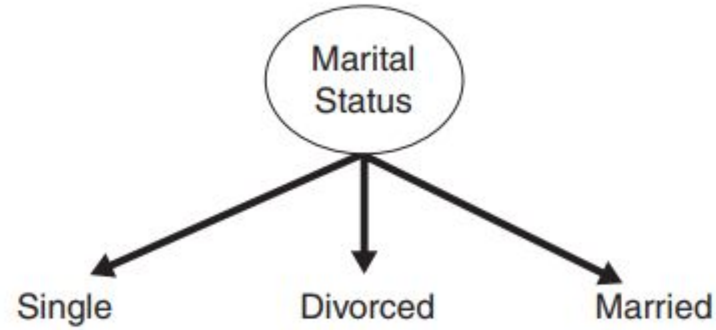


Binary split

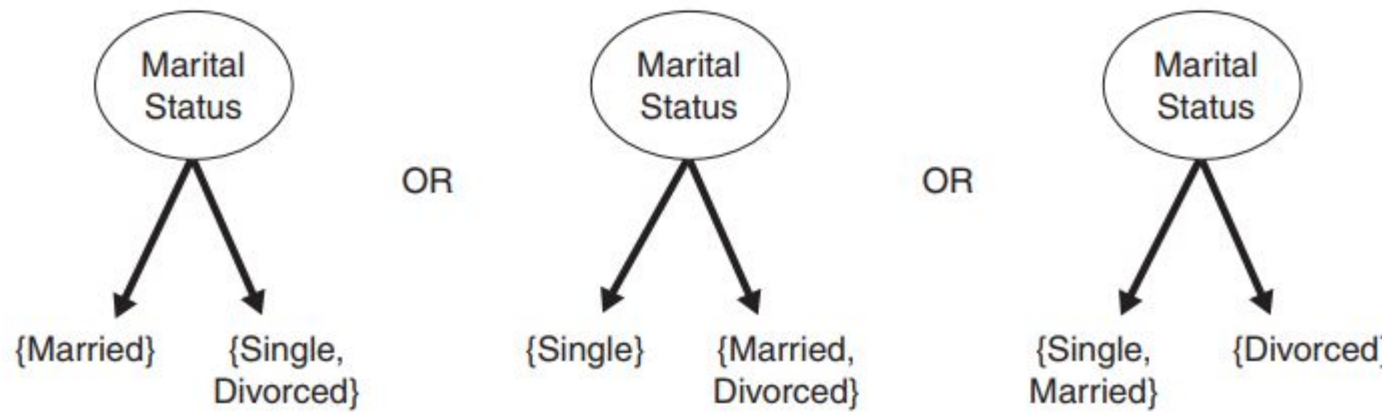


Multiway split

Attribute Test Conditions & Attribute type



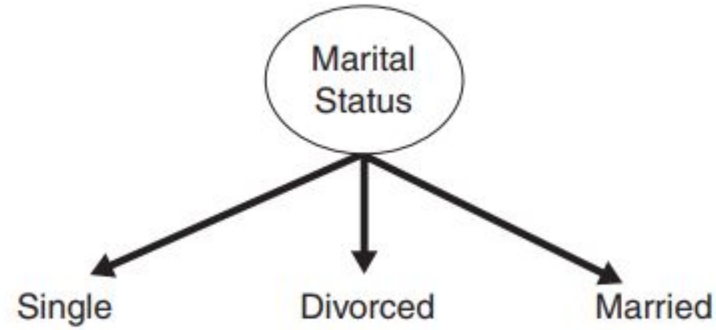
Multiway split



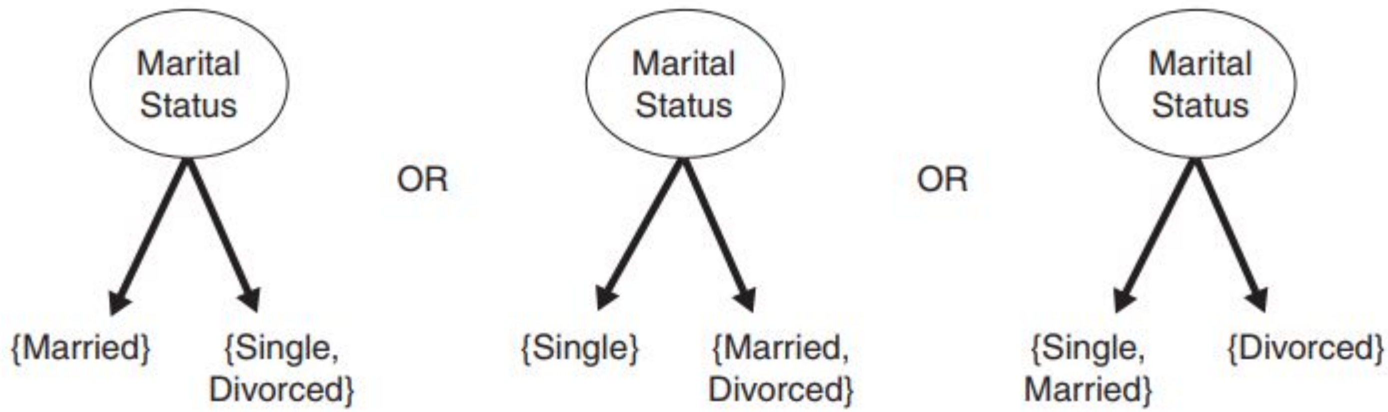
Binary split {by grouping attribute values}

Binary split can be used with more than two outcomes (Ex. CART algorithm).

Attribute Test Conditions & Attribute type



Multiway split



Binary split {by grouping attribute values}

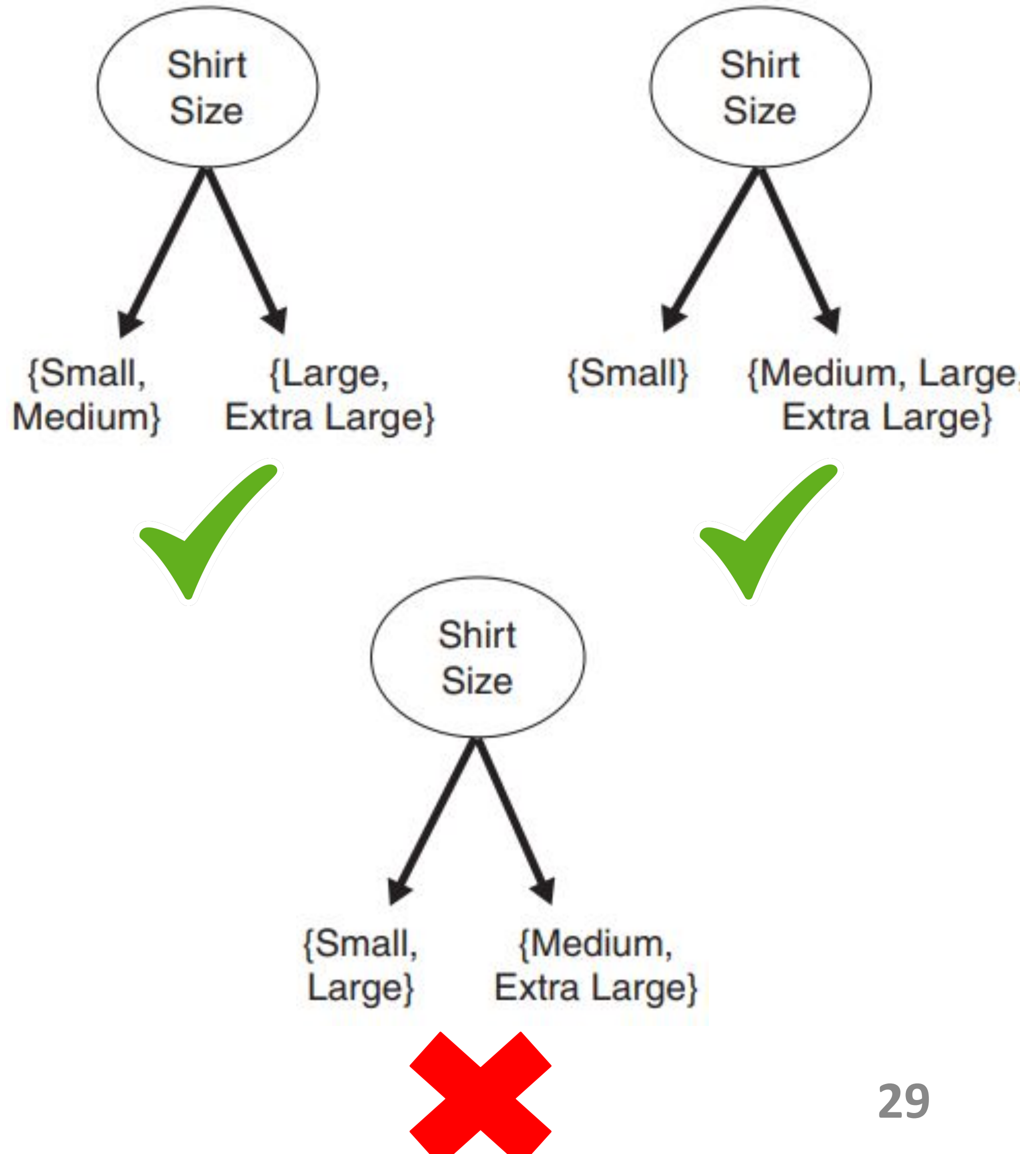
2^(k)-1 potential groupings.

How to select the optimal grouping?

Attribute Test Conditions & Attribute type

- **Ordinal Attributes**
  - **Binary or multiway splits** (like nominal attributes).
  - **Binary split:** grouping should not violate the order.

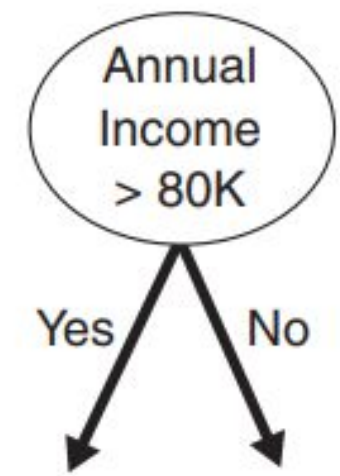
How many possible grouping ?



Attribute Test Conditions & Attribute type

Continuous Attributes

- **Binary Split**
  - Comparison test (**A < V**).
  - MinTrain(attribute) < **V** < MaxTrain(attribute).
  - Training Attributes values can be considered for splits **V**.

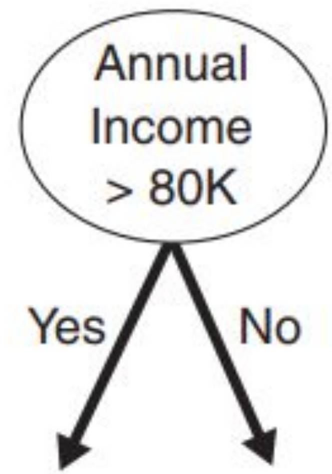


- **Multiway Split**
  - Intervals (Vi ≤ A < Vi+1) for i = 1, ..., k.
  - Non Overlapping intervals.
  - Discretization to convert to ordinal attribute.
  - Test condition defined like ordinal attribute.

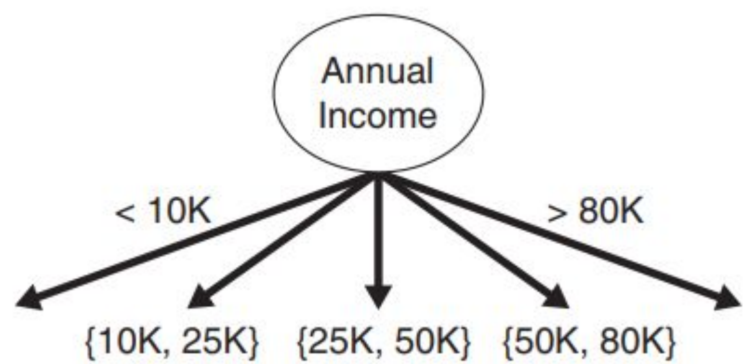
Attribute Test Conditions & Attribute type

Continuous Attributes

- **Binary Split**
  - Comparison test (**A < V**).
  - MinTrain(attribute) < **V** < MaxTrain(attribute).
  - Training Attributes values can be considered for splits **V**.



- **Multiway Split**
  - Intervals (**Vi ≤ A < Vi+1**) for i = 1, ..., k.
  - Non Overlapping intervals.
  - Discretization to convert to ordinal attribute.
  - Test condition defined like ordinal attribute.



Chapter Overview

- Introduction to Classification
- Decision Tree Induction
  - Introduction to Decision Tree
  - Attribute Test Conditions
  - Impurity Measures and Splitting Strategies
  - Gain Ratio

Chapter Overview

- Introduction to Classification

- Decision Tree Induction
  - Introduction to Decision Tree
  - Attribute Test Conditions
  - Impurity Measures and Splitting Strategies
  - Gain Ratio

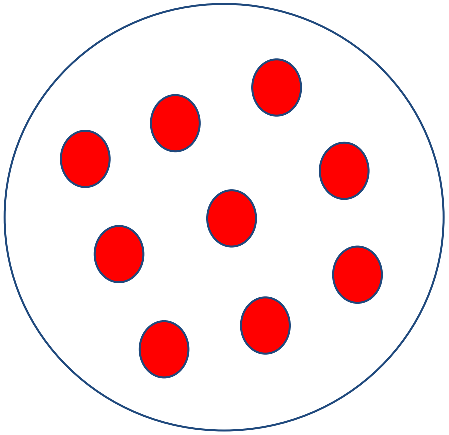
Measures for Selecting an Attribute Test Condition

- **Objective**
  - Prefer Attribute tests leading to **pure child nodes**.
- **Why ?**
  - Pure nodes helps to stop expanding nodes.
  - Impure nodes need more expansions, deepening the tree.
- **Concerns with Larger Trees**
  - Susceptible to overfitting.
  - Harder to interpret.
  - Longer training and testing times.

What is pure nodes ?

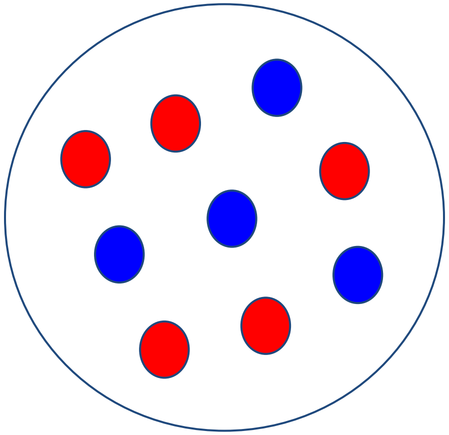
Pure set VS Impure set

- **Pure node**
  - Contains the same class label.
  - Only one class label **have probability 1**.



Pure set

- **Impure node**
  - Contains a mixture of different class labels.
  - Maximum impurity occurs when **class labels are equally probable.**



Impure set

Impurity Measure for a Single Node

Impurity of a node measures how dissimilar the class labels of instances in a node.

$$Entropy = -\sum_{i=0}^{c-1} p_i(t)\log_2 p_i(t)$$

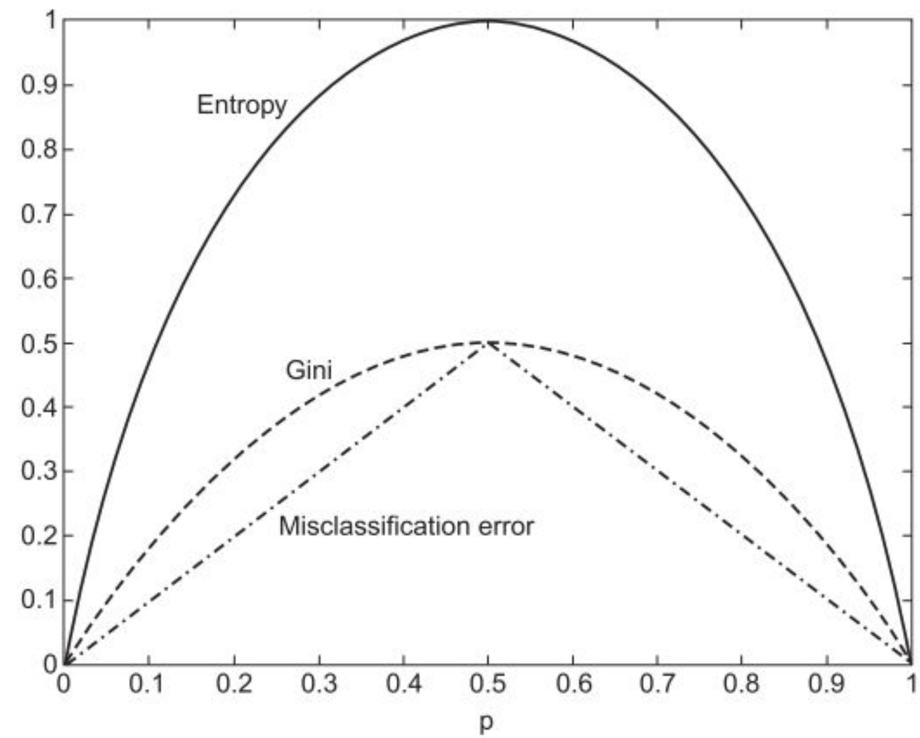
$$Gini\ Index = 1 - \sum_{i=0}^{c-1} p_i(t)^2$$

$$Classification\ Error = 1 - \max_i [p_i(t)]$$

$p_i(t)$: relative frequency of class **i** at node **t**, **c** is the number of classes.

Impurity Measure for a Single Node

- **Zero impurity for a single-class node.**
- **Maximum impurity for equally distributed classes.**
- **The three measures are consistent.**



Examples

Node N_1	Count
Class=0	0
Class=1	6

Node N_2	Count
Class=0	1
Class=1	5

Node N_3	Count
Class=0	3
Class=1	3

Examples

Node N_1	Count
Class=0	0
Class=1	6

$$\text{Gini} = 1 - (0/6)^2 - (6/6)^2 = 0$$
$$\text{Entropy} = -(0/6)\log_2(0/6) - (6/6)\log_2(6/6) = 0$$
$$\text{Error} = 1 - \max[0/6, 6/6] = 0$$

Node N_2	Count
Class=0	1
Class=1	5

$$\text{Gini} = 1 - (1/6)^2 - (5/6)^2 = 0.278$$
$$\text{Entropy} = -(1/6)\log_2(1/6) - (5/6)\log_2(5/6) = 0.650$$
$$\text{Error} = 1 - \max[1/6, 5/6] = 0.167$$

Node N_3	Count
Class=0	3
Class=1	3

$$\text{Gini} = 1 - (3/6)^2 - (3/6)^2 = 0.5$$
$$\text{Entropy} = -(3/6)\log_2(3/6) - (3/6)\log_2(3/6) = 1$$
$$\text{Error} = 1 - \max[3/6, 3/6] = 0.5$$

Collective Impurity of Child Nodes

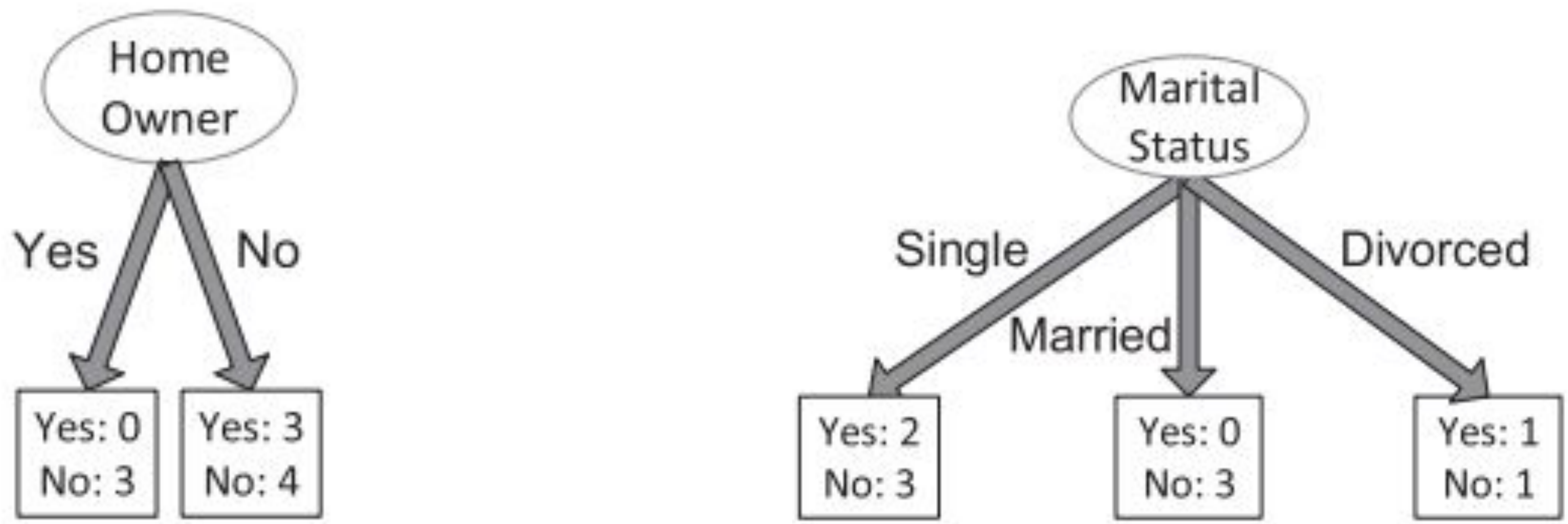
- Splits a node with N instances into k children **{v1, v2, ..., vk}**.

- $N(v_j)$: Number of instances in child node **vj**.
- $I(v_j)$: Impurity value of node **vj**.

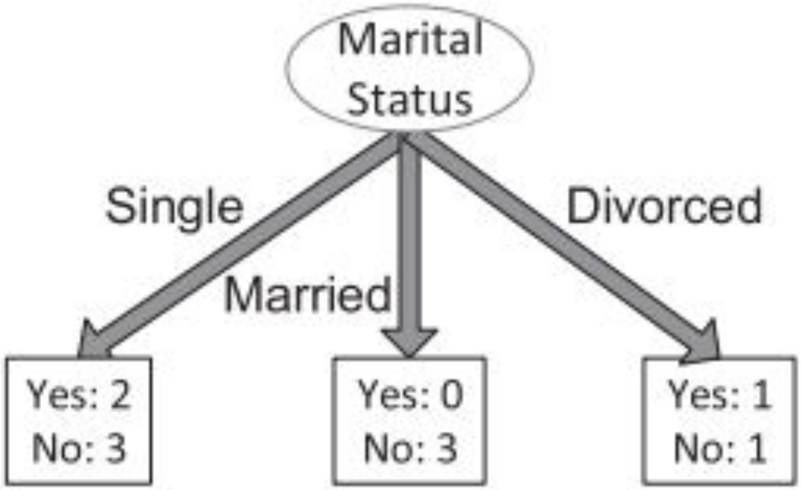
$$I(\text{children}) = \sum_{j=1}^{k} \frac{N(v_j)}{N} I(v_j)$$

- Collective impurity of child nodes:
  - Weighted sum of node children impurities.

Which split is better ?



Which split is better ?



$$I(\texttt{Home Owner = yes}) = -\frac{0}{3}\log_2\frac{0}{3} - \frac{3}{3}\log_2\frac{3}{3} = 0$$

$$I(\texttt{Home Owner = no}) = -\frac{3}{7}\log_2\frac{3}{7} - \frac{4}{7}\log_2\frac{4}{7} = 0.985$$

$$I(\texttt{Home Owner}) = \frac{3}{10} \times 0 + \frac{7}{10} \times 0.985 = 0.690$$

$$I(\texttt{Marital Status = Single}) = -\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5} = 0.971$$

$$I(\texttt{Marital Status = Married}) = -\frac{0}{3}\log_2\frac{0}{3} - \frac{3}{3}\log_2\frac{3}{3} = 0$$

$$I(\texttt{Marital Status = Divorced}) = -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2} = 1.000$$

$$I(\texttt{Marital Status}) = \frac{5}{10} \times 0.971 + \frac{3}{10} \times 0 + \frac{2}{10} \times 1 = 0.686$$

Identifying the best attribute test condition

Compare **parent node impurity (I(parent))** before splitting with after splitting.

$$\Delta = I(\text{parent}) - I(\text{children})$$

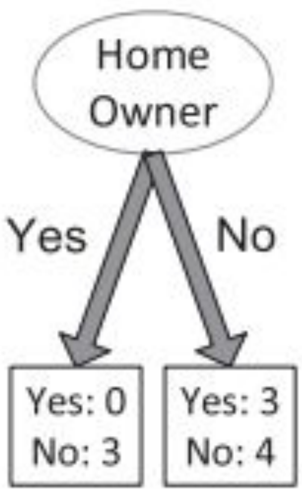
- **Large (Δ)** ⇒ better attribute test condition.
- **Δ_info:** information gain when entropy is used.
- **I(parent)≥I(children):** Gain is always non-negative.

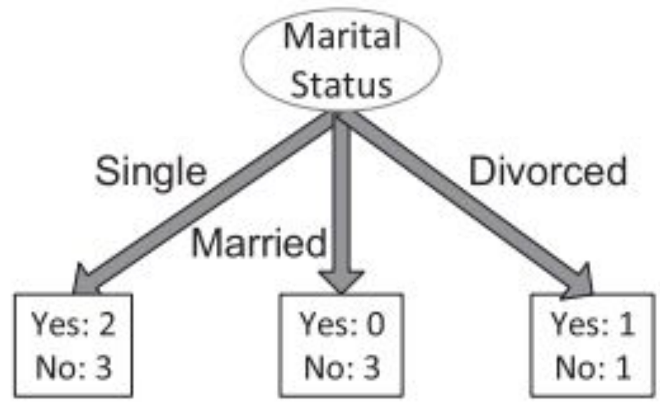
Decision trees select conditions with maximum gain for splitting.

Maximizing gain is equivalent to minimizing weighted child impurity.

Splitting of Qualitative Attributes

$$I(\text{parent}) = -\frac{3}{10}\log_2\frac{3}{10} - \frac{7}{10}\log_2\frac{7}{10} = 0.881$$





Δ_info = 0.881 - 0.690

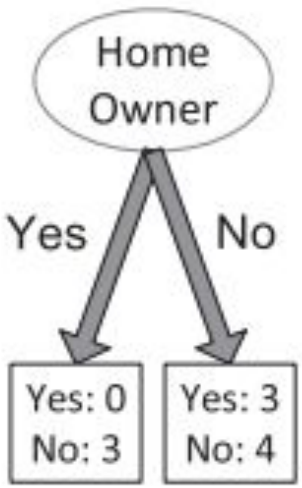
= 0.191

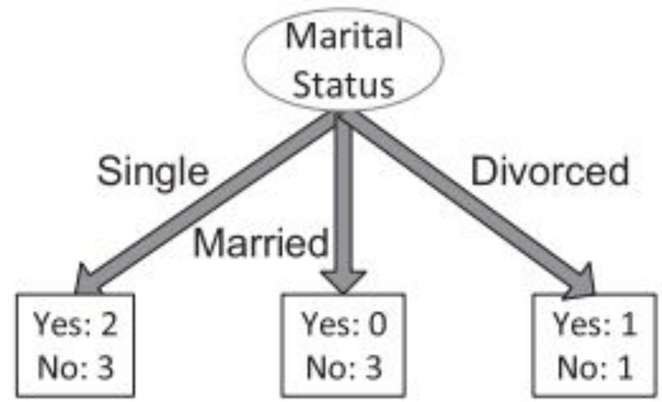
Δ_info = 0.881 - 0.686

= 0.195

Splitting of Qualitative Attributes

$$I(\text{parent}) = -\frac{3}{10}\log_2\frac{3}{10} - \frac{7}{10}\log_2\frac{7}{10} = 0.881$$





Δ_info = 0.881 - 0.690

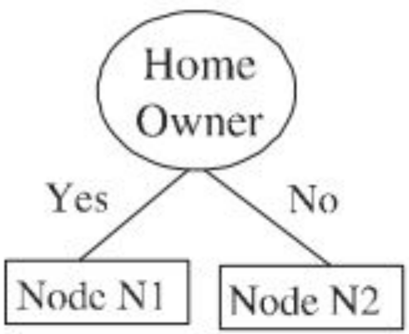
= 0.191

Δ_info = 0.881 - 0.686

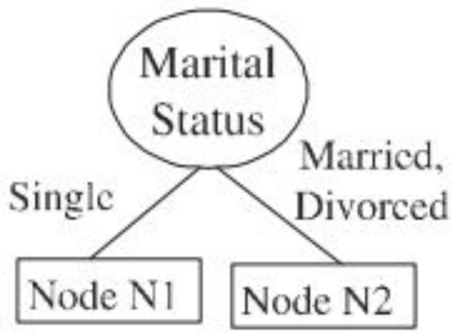
= 0.195

Binary Splitting of Qualitative Attributes

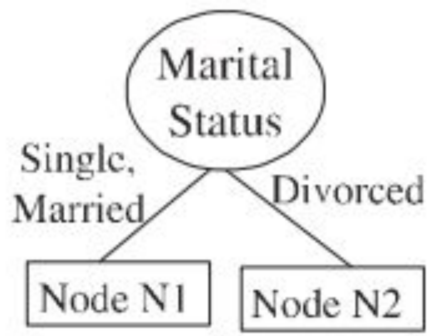
	Parent
No	7
Yes	3
Gini = 0.420	



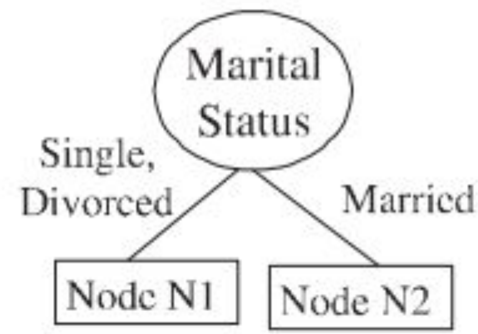
	N1	N2
No	3	4
Yes	0	3
Gini= 0.343		



	N1	N2
No	3	4
Yes	2	1
Gini= 0.400		



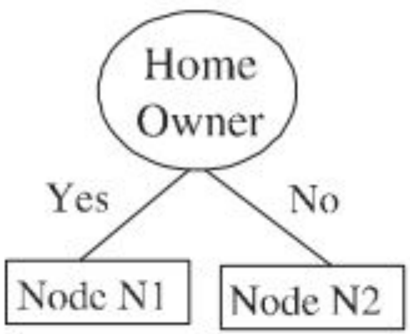
	N1	N2
No	6	1
Yes	2	1
Gini= 0.400		



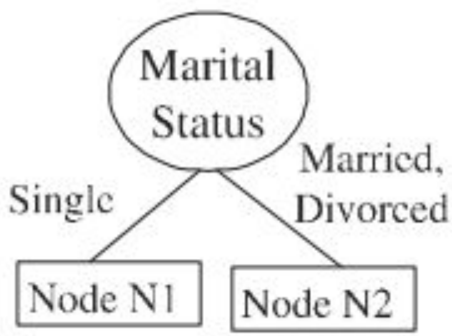
	N1	N2
No	4	3
Yes	3	0
Gini= 0.343		

Binary Splitting of Qualitative Attributes

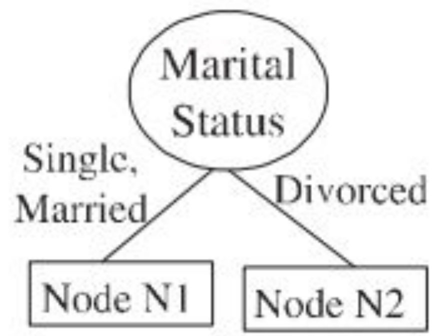
	Parent
No	7
Yes	3
Gini = 0.420	



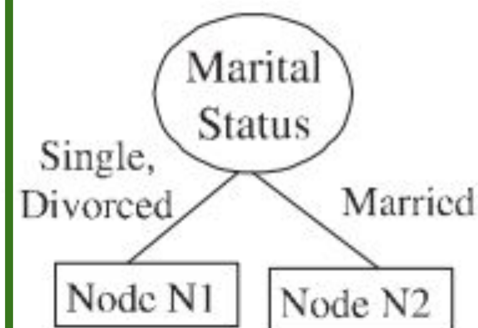
	N1	N2
No	3	4
Yes	0	3
Gini= 0.343		



	N1	N2
No	3	4
Yes	2	1
Gini= 0.400		



	N1	N2
No	6	1
Yes	2	1
Gini= 0.400		



	N1	N2
No	4	3
Yes	3	0
Gini= 0.343		

Binary Splitting of Quantitative Attributes

Class	No	No	No	Yes	Yes	Yes	No	No	No	No
	Annual Income (in '000s)									
Sorted Values →	60	70	75	85	90	95	100	120	125	220

Split Positions →	55		65		72.5		80		87.5		92.5		97.5		110		122.5		172.5		230	
	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>
Yes	0	3	0	3	0	3	0	3	1	2	2	1	3	0	3	0	3	0	3	0	3	0
No	0	7	1	6	2	5	3	4	3	4	3	4	3	4	4	3	5	2	6	1	7	0
Gini	0.420		0.400		0.375		0.343		0.417		0.400		*0.300*		0.343		0.375		0.400		0.420	

1. Order attribute values with **O(nlog(n))** complexity.
2. Choose split positions at midpoints between adjacent sorted values.
3. Compute Gini index in one pass, **O(n)**.
4. Identify optimal split using the Gini index.

The complexity of finding the best binary split is O(n).

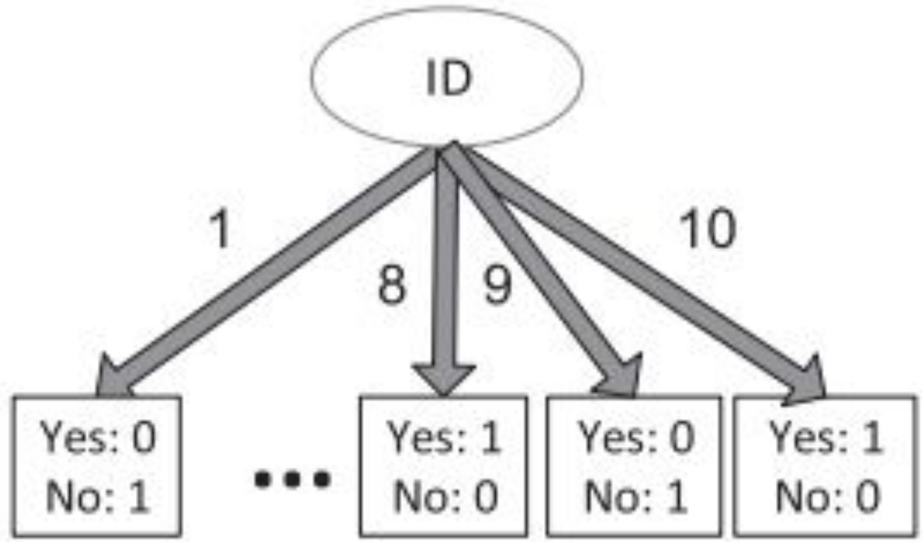
Synthesis of Attribute test selection

Selecting the best Attribute Test for Node Expansion :

1. Identify the best split (if more than one split) for each attribute using **Δ=I(parent)−I(children).**

2. Select the best attribute by comparing **Δ=I(parent)−I(children)** across all the attributes.

Limitation of the impurity measure Δ



What is the Δ(ID) ?