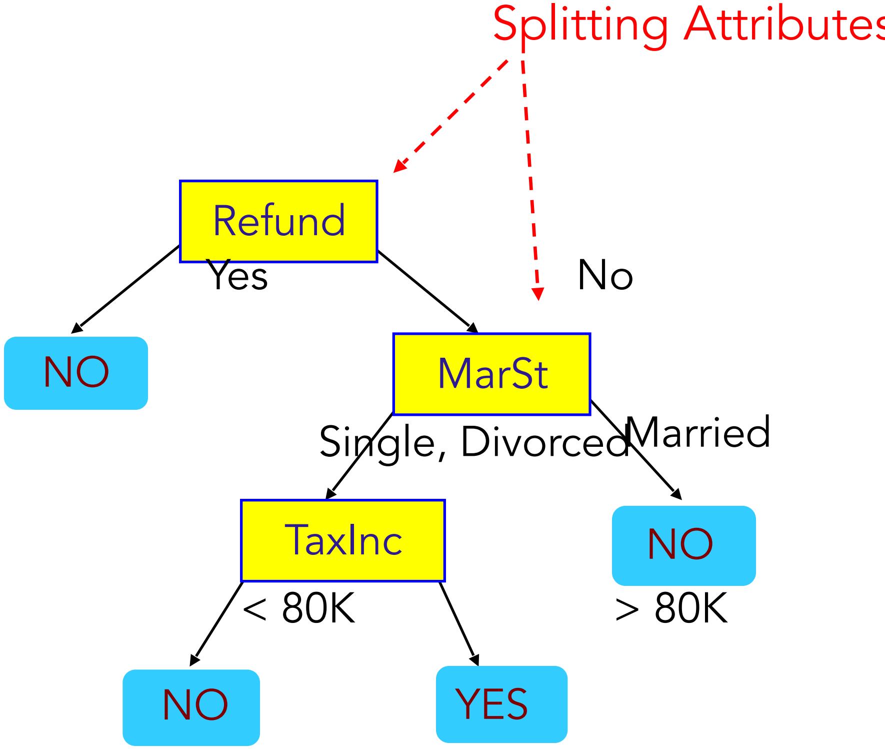
- Two basic machine learning algorithms
    - decision trees
    - nearest neighbor algorithm

-

Example of a Decision Tree

categorical categorical continuous class



Splitting Attributes

Refund

Yes

No

NO

MarSt

Single, Divorced

Married

TaxInc

NO

< 80K

> 80K

NO

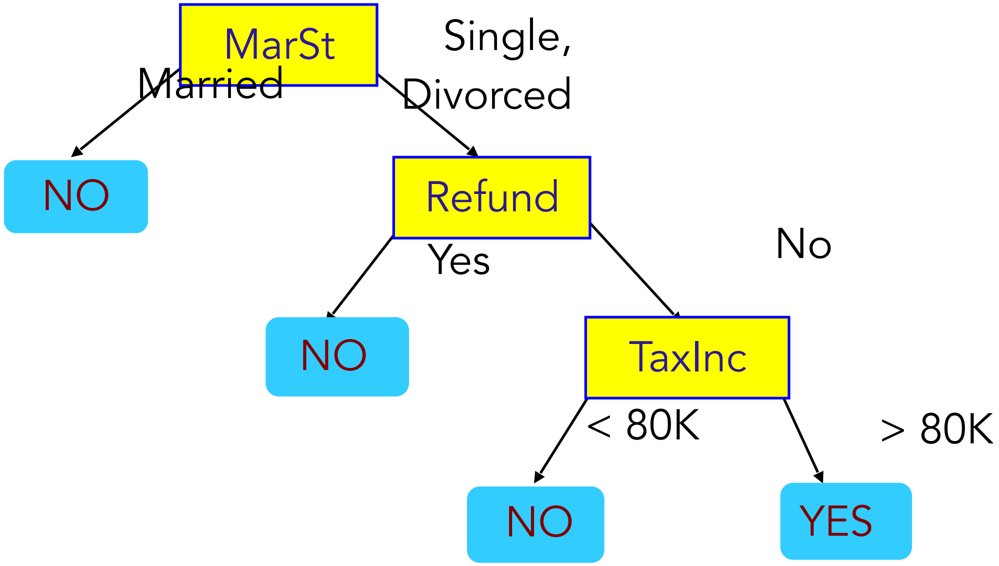
YES

Training Data

Model: Decision Tree

Another Example of Decision Tree

categorical categorical continuous class



There could be more than one tree that fits the same data!

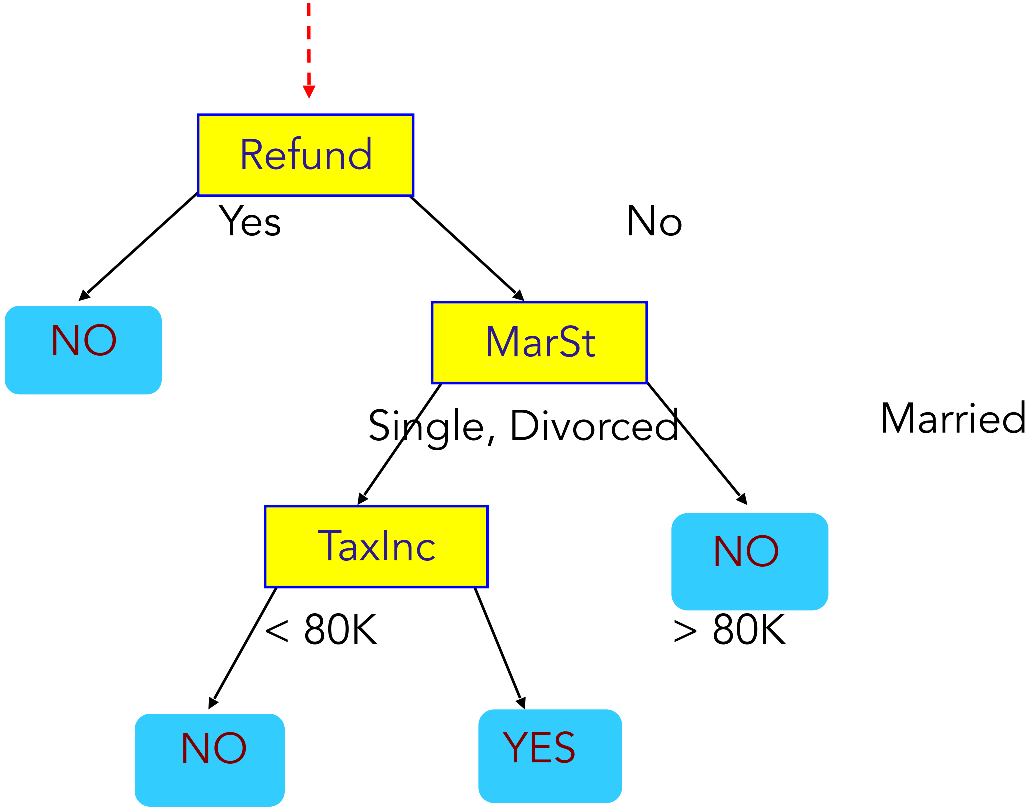
Decision Tree Classification Task

Decision
Tree

Apply Model to Test Data

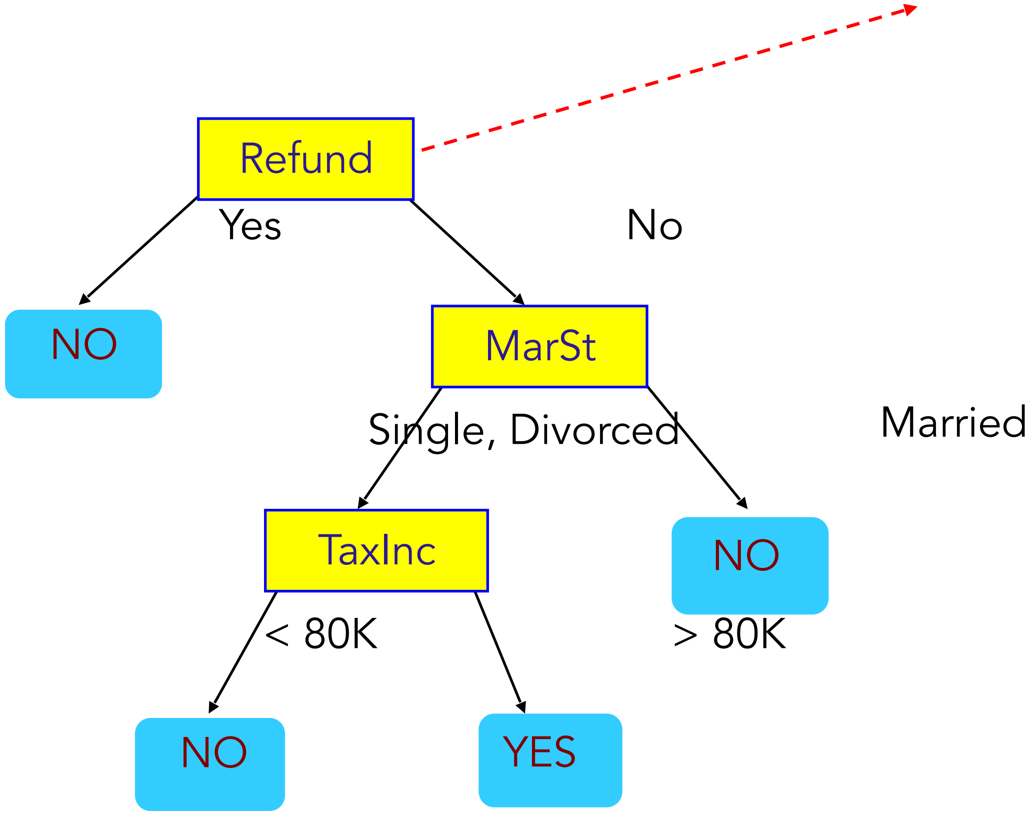
Test Data

Start from the root of tree.



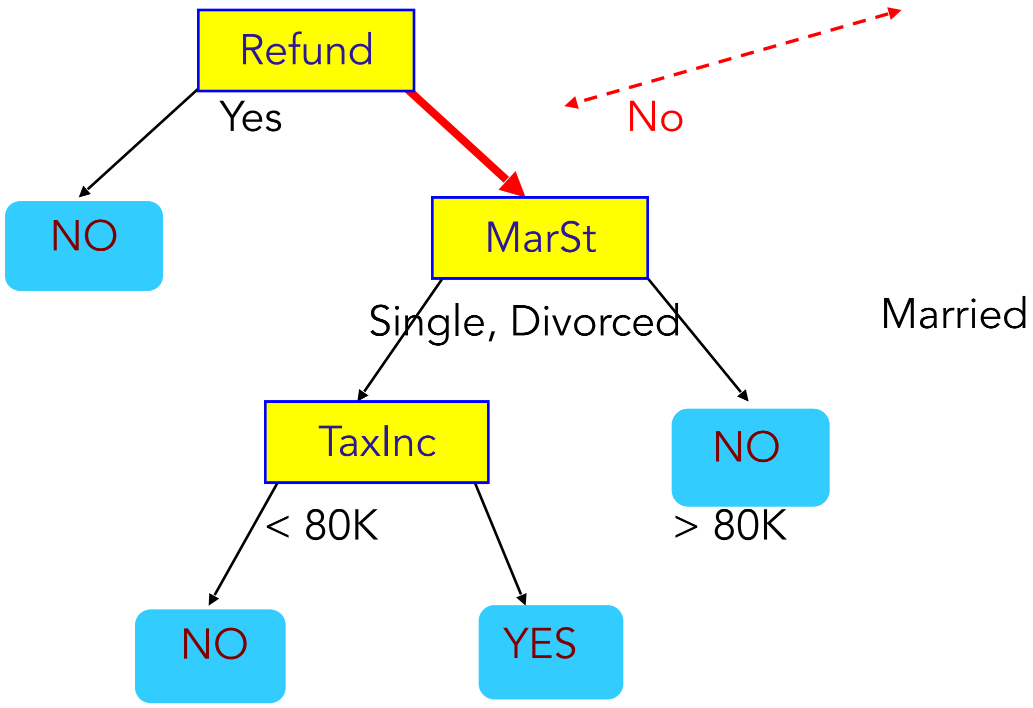
Apply Model to Test Data

Test Data



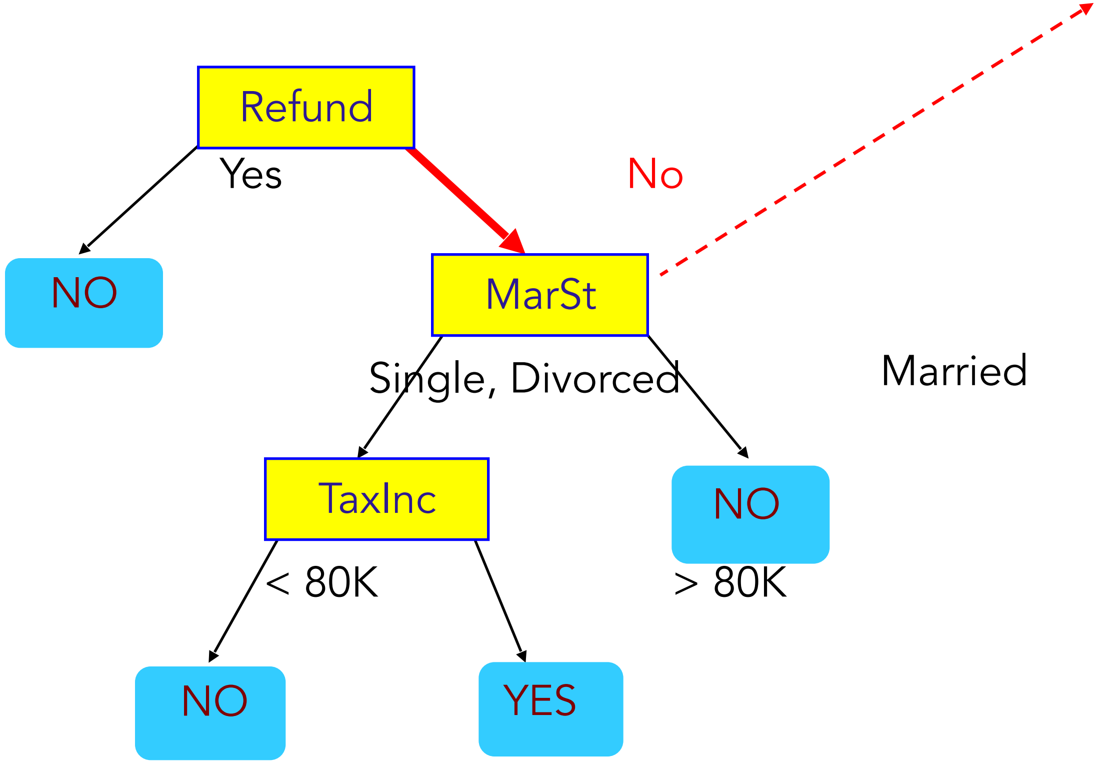
Apply Model to Test Data

Test Data



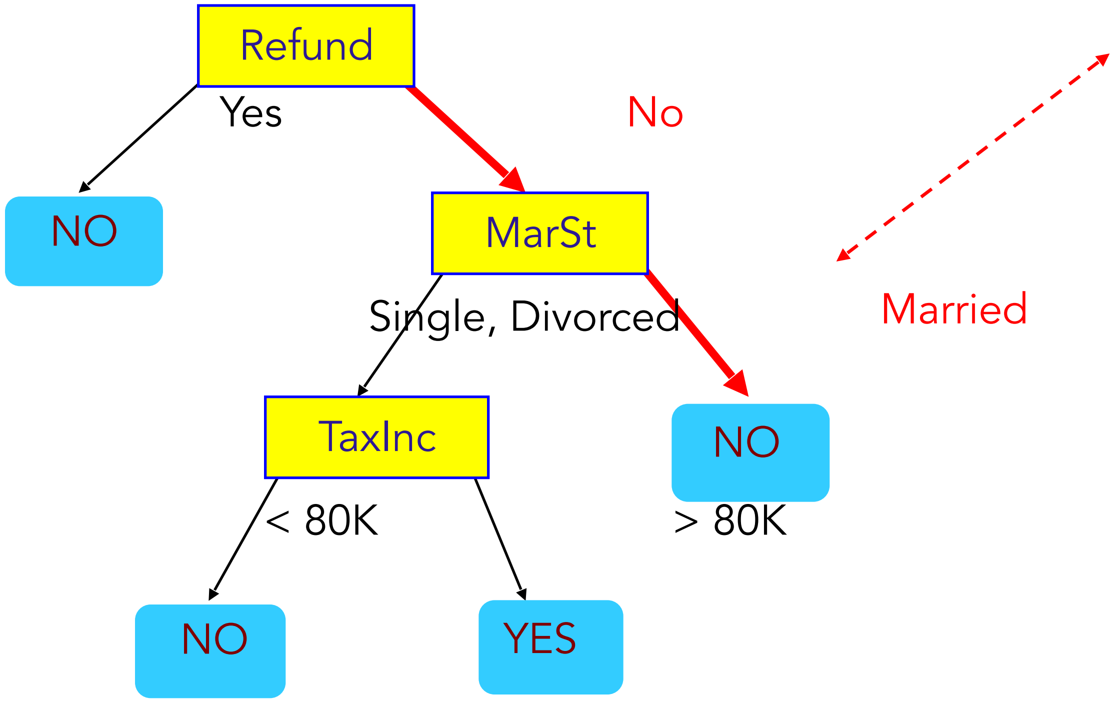
Apply Model to Test Data

Test Data



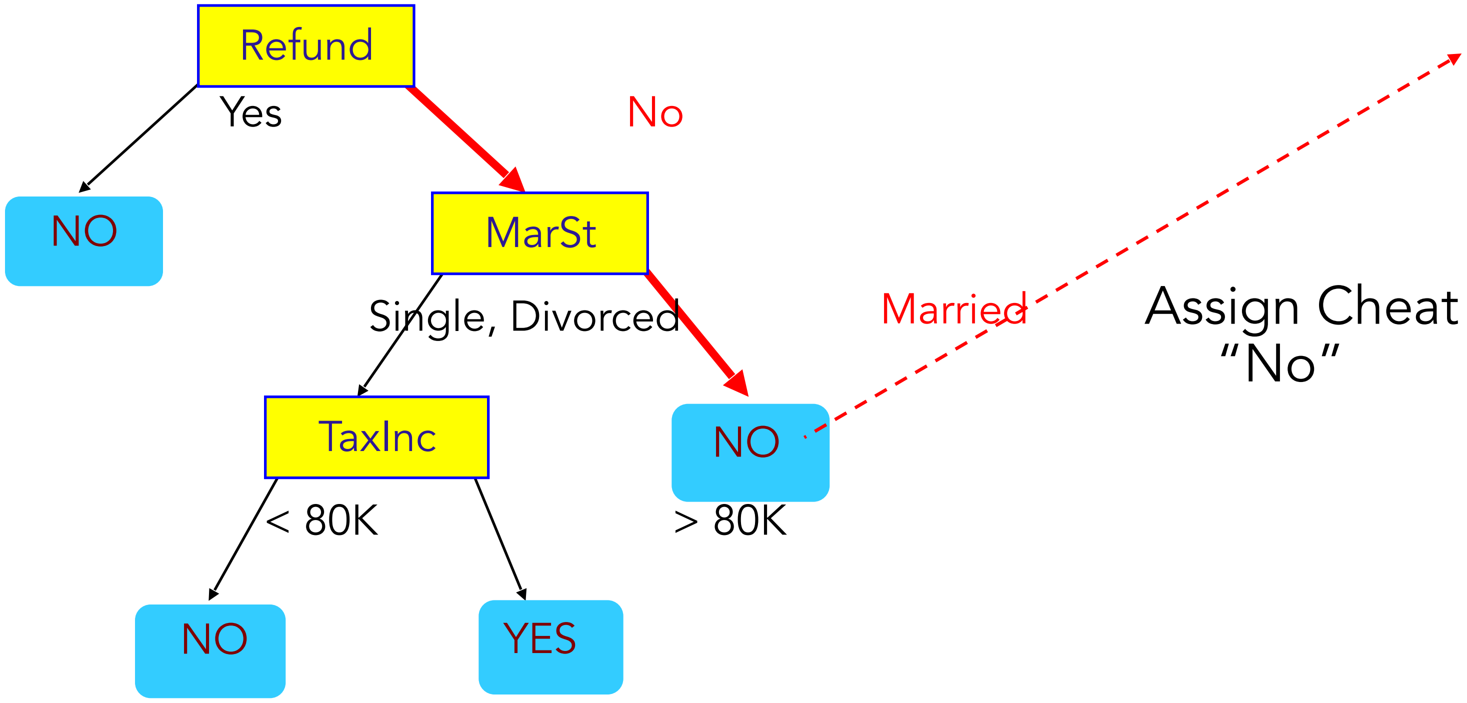
Apply Model to Test Data

Test Data



Apply Model to Test Data

Test Data



Assign Cheat to "No"

Decision Tree Classification Task

Decision Tree

Tree Induction

- Greedy strategy.
  - Split the records based on an attribute test that optimizes certain criterion.

- Issues
  - Determine how to split the records
    - How to specify the attribute test condition?
    - How to determine the best split?
  - Determine when to stop splitting

How to determine the Best Split

Before Splitting: 10 records of class 0,
10 records of class 1

Which test condition is the best?

How to determine the Best Split

- Greedy approach:
  - Nodes with homogeneous class distribution are preferred
- Need a measure of node impurity:

Non-homogeneous,

High degree of impurity

Homogeneous,

Low degree of impurity

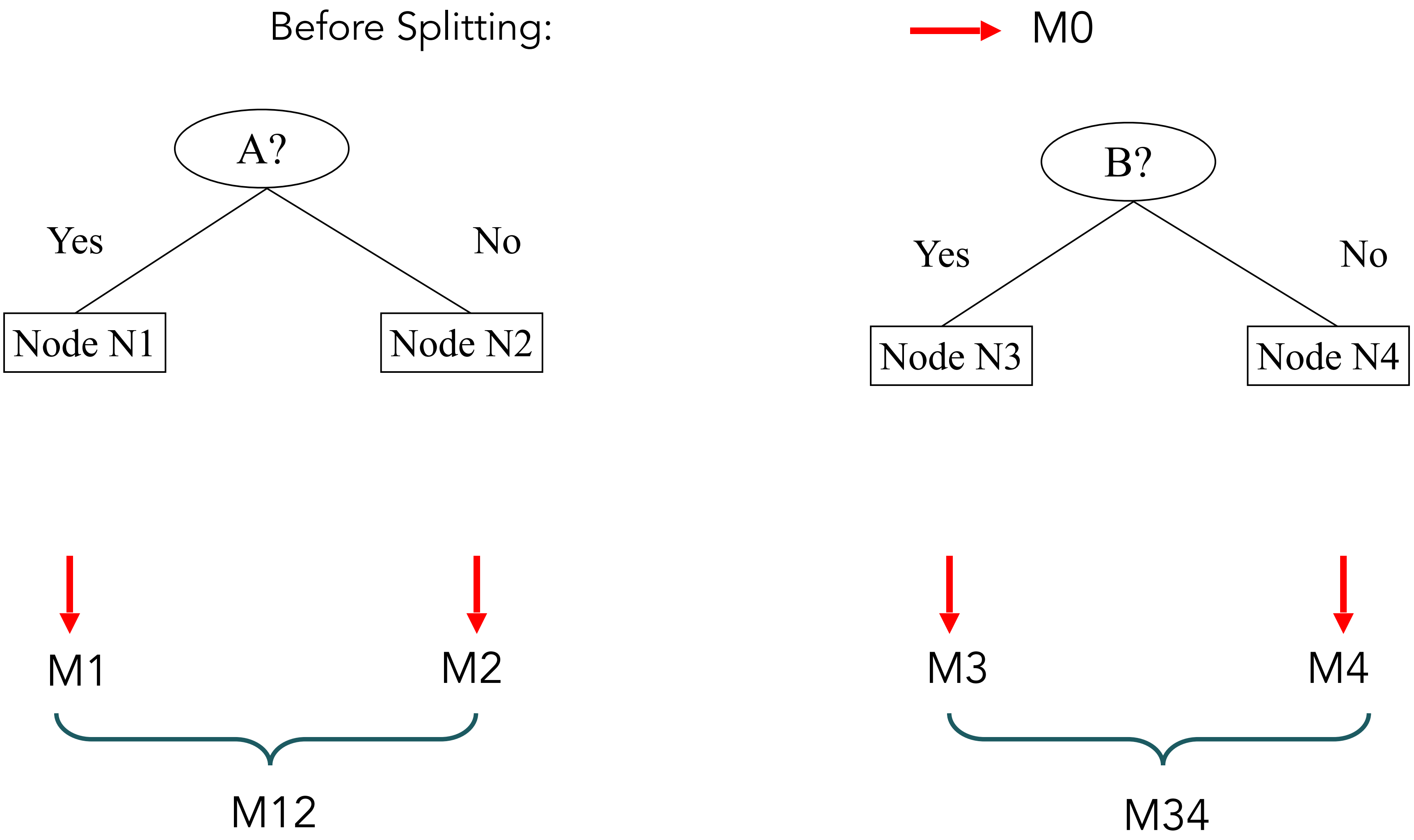
Measures of Node Impurity

- Gini Index

- Entropy

- Misclassification error

How to Find the Best Split


Before Splitting:

M0

A?

Yes

No

Node N1

Node N2

B?

Yes

No

Node N3

Node N4

M1

M2

M3

M4

M12

M34

Gain = M0 – M12 vs M0 – M34

Measure of Impurity: GINI

- Gini Index for a given node t :

(NOTE: $p(j \mid t)$ is the relative frequency of class j at node t).

- Maximum (1 - 1/n_c) when records are equally distributed among all classes, implying least interesting information
- Minimum (0.0) when all records belong to one class, implying most interesting information

Examples for computing GINI

$P(C1) = 0/6 = 0 \qquad P(C2) = 6/6 = 1$

$\text{Gini} = 1 – P(C1)^2 – P(C2)^2 = 1 – 0 – 1 = 0$

$P(C1) = 1/6 \qquad P(C2) = 5/6$

$\text{Gini} = 1 – (1/6)^2 – (5/6)^2 = 0.278$

$P(C1) = 2/6 \qquad P(C2) = 4/6$

$\text{Gini} = 1 – (2/6)^2 – (4/6)^2 = 0.444$

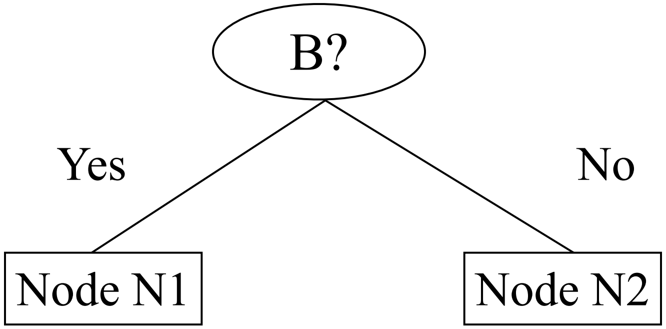
Splitting Based on GINI

- Used in CART, SLIQ, SPRINT.
- When a node p is split into k partitions (children), the quality of split is computed as,

where n_i = number of records at child i, n = number of records at node p.

Binary Attributes: Computing GINI Index

- Splits into two partitions
- Effect of Weighing partitions:
  - Larger and Purer Partitions are sought for.



Gini(N1)
$= 1 - (5/6)^2 - (2/6)^2$
$= 0.194$

Gini(N2)
$= 1 - (1/6)^2 - (4/6)^2$
$= 0.528$

Gini(Children)
$= 7/12 * 0.194 +$
$5/12 * 0.528$
$= 0.333$

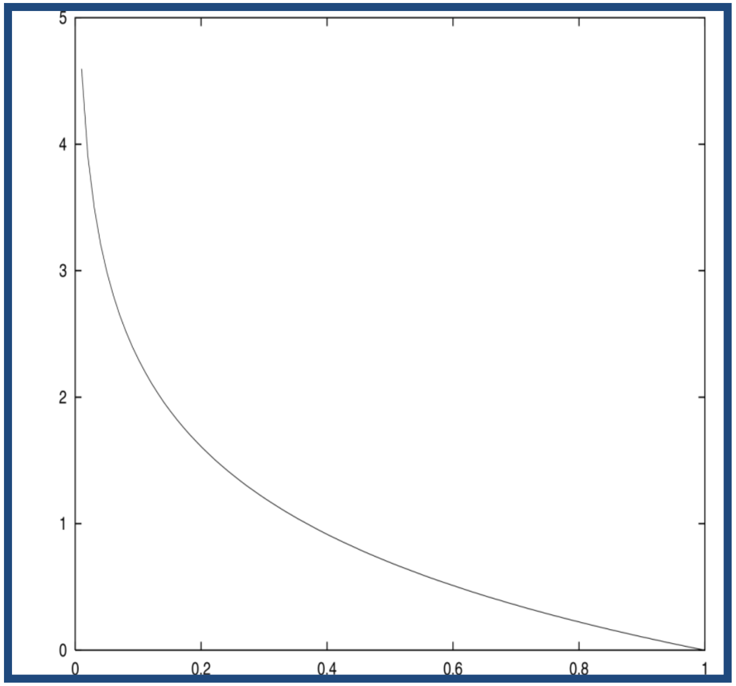
Information/Entropy

- Given probabilities p_1, p_2, .., p_s whose sum is 1, Entropy is defined as:

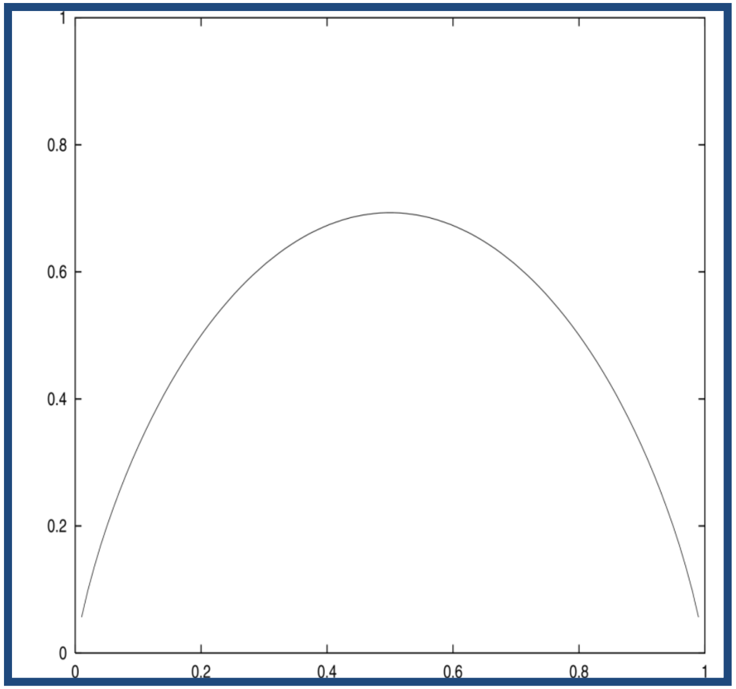
$$\boxed{H(p_1, p_2, ..., p_s) = \sum_{i=1}^{s} (p_i log(1/p_i))}$$

- Entropy measures the amount of randomness or surprise or uncertainty.
- Goal in classification
  - no surprise
  - entropy = 0

Entropy



log (1/p)



H(p,1-p)

Alternative Splitting Criteria based on INFO

- Entropy at a given node t:

  (NOTE: $p(j \mid t)$ is the relative frequency of class j at node t).

  - Measures homogeneity of a node.
    - Maximum (log n_c) when records are equally distributed among all classes implying least information
    - Minimum (0.0) when all records belong to one class, implying most information
  - Entropy based computations are similar to the GINI index computations

Examples for computing Entropy

$P(C1) = 0/6 = 0 \qquad P(C2) = 6/6 = 1$

$\text{Entropy} = -\,0 \log 0 - 1 \log 1 = -\,0 - 0 = 0$

$P(C1) = 1/6 \qquad P(C2) = 5/6$

$\text{Entropy} = -\,(1/6) \log_2 (1/6) - (5/6) \log_2 (1/6) = 0.65$

$P(C1) = 2/6 \qquad P(C2) = 4/6$

$\text{Entropy} = -\,(2/6) \log_2 (2/6) - (4/6) \log_2 (4/6) = 0.92$

Splitting Based on INFOrmation gain

- Information Gain:

Parent Node, p is split into k partitions;

n_i is number of records in partition i

- Measures Reduction in Entropy achieved because of the split. Choose the split that achieves most reduction (maximizes GAIN)
- Used in ID3 and C4.5
- Disadvantage: Tends to prefer splits that result in large number of partitions, each being small but pure.

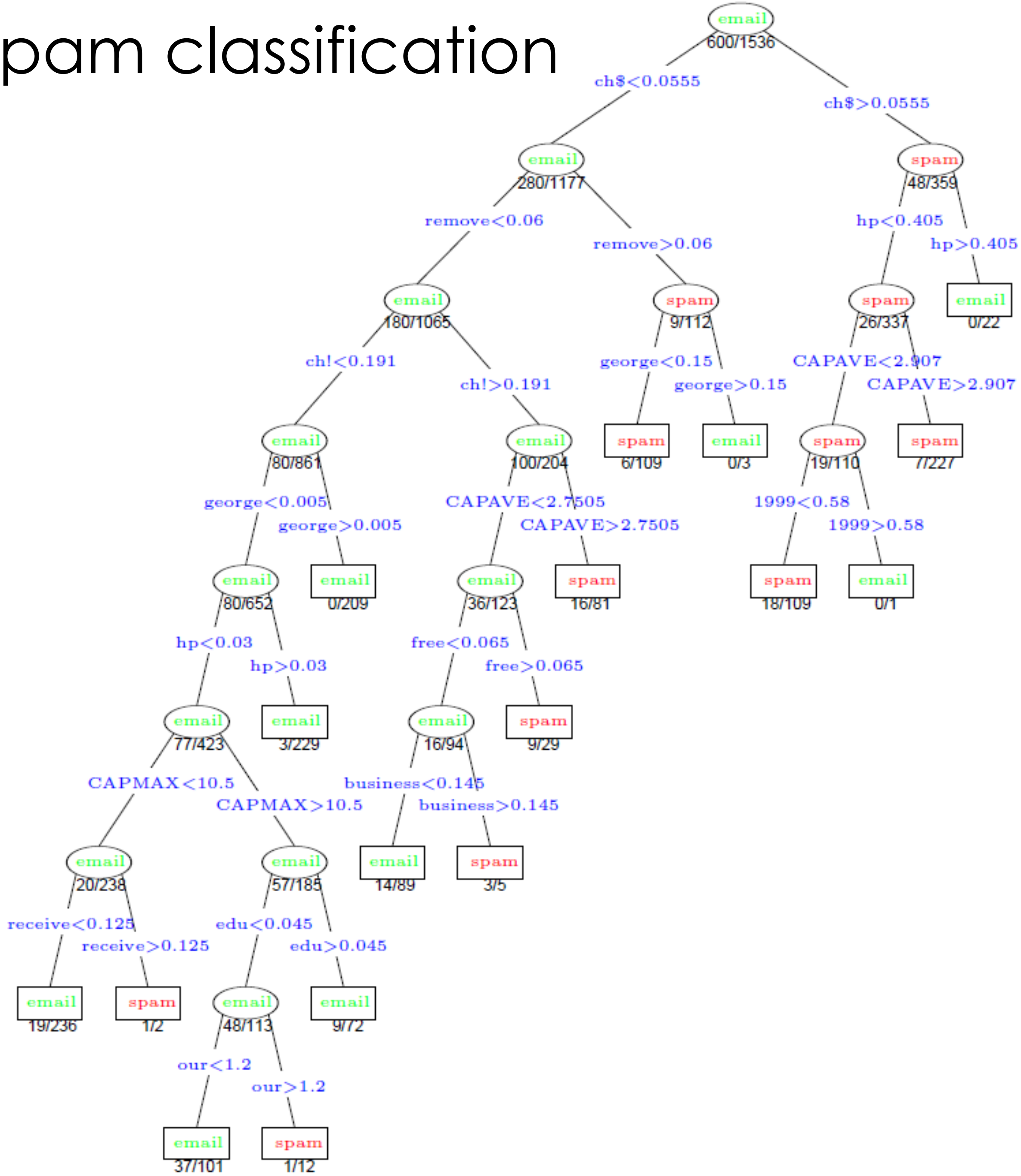
Splitting Based on Information Gain

Gain Ratio:

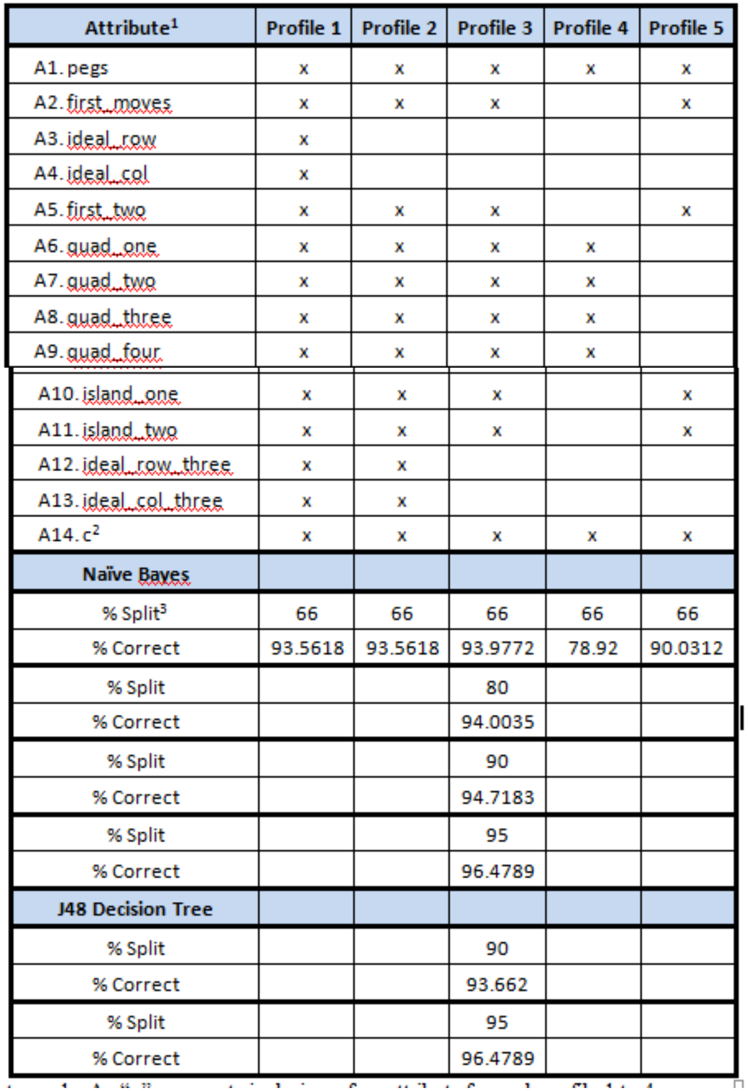
Parent Node, p is split into k partition, n_i is the number of records in partition i

- Adjusts Information Gain by the entropy of the partitioning (SplitINFO). Higher entropy partitioning (large number of small partitions) is penalized!
- Used in C4.5
- Designed to overcome the disadvantage of Information Gain

Spam classification



peg-solitaire classification using decision tree



Attribute[1]	Profile 1	Profile 2	Profile 3	Profile 4	Profile 5
A1. pegs	x	x	x	x	x
A2. first_moves	x	x	x		x
A3. ideal_row	x				
A4. ideal_col	x				
A5. first_two	x	x	x		x
A6. quad_one	x	x	x	x	
A7. quad_two	x	x	x	x	
A8. quad_three	x	x	x	x	
A9. quad_four	x	x	x	x	
A10. island_one	x	x	x		x
A11. island_two	x	x	x		x
A12. ideal_row_three	x	x			
A13. ideal_col_three	x	x			
A14. c[2]	x	x	x	x	x
Naïve Bayes					
% Split[3]	66	66	66	66	66
% Correct	93.5618	93.5618	93.9772	78.92	90.0312
% Split			80		
% Correct			94.0035		
% Split			90		
% Correct			94.7183		
% Split			95		
% Correct			96.4789		
J48 Decision Tree					
% Split			90		
% Correct			93.662		
% Split			95		
% Correct			96.4789		