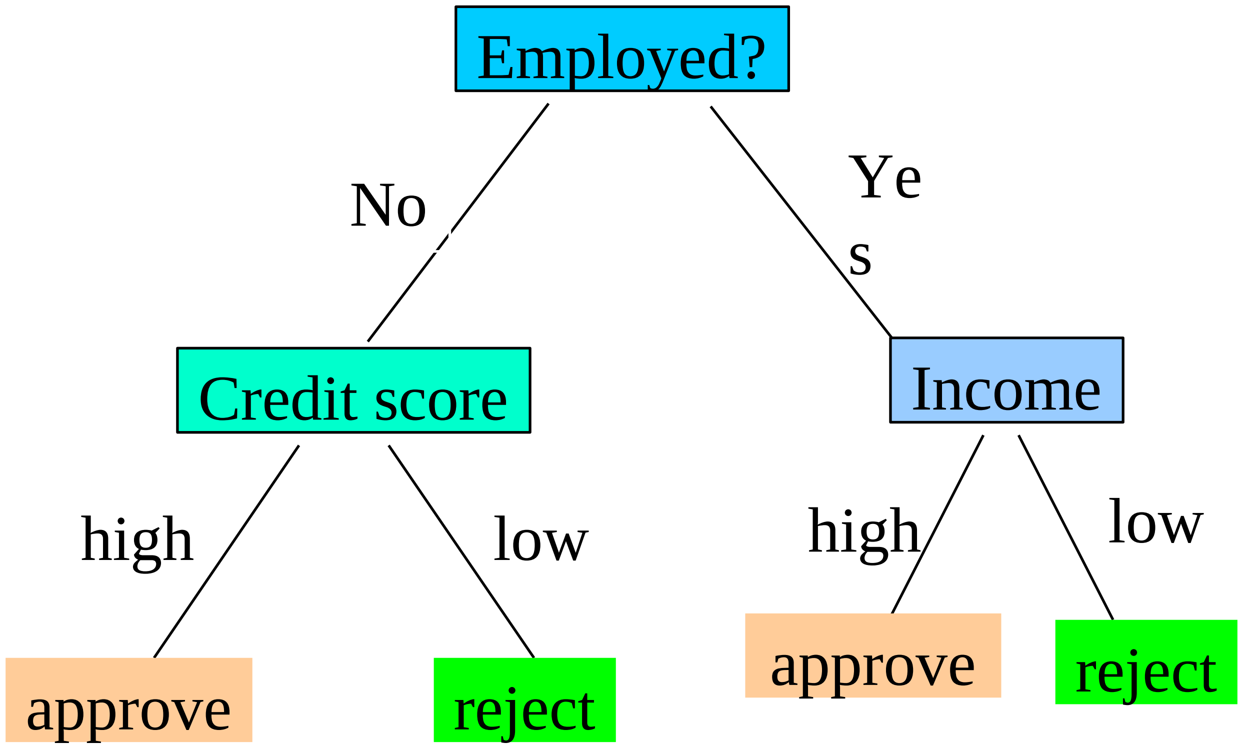
# Decision Tree Classifier

Dr.Manju Venugopalan
Department of Computer Science & Engineering
Amrita School of Engineering, Bengaluru

# What are Decision trees?

- A decision tree is a tree in which each branch node represents a choice between a number of alternatives, and each leaf node represents a decision.
- A type of supervised learning algorithm.

# Decision Tree An Example




Whether to approve/reject a loan application?

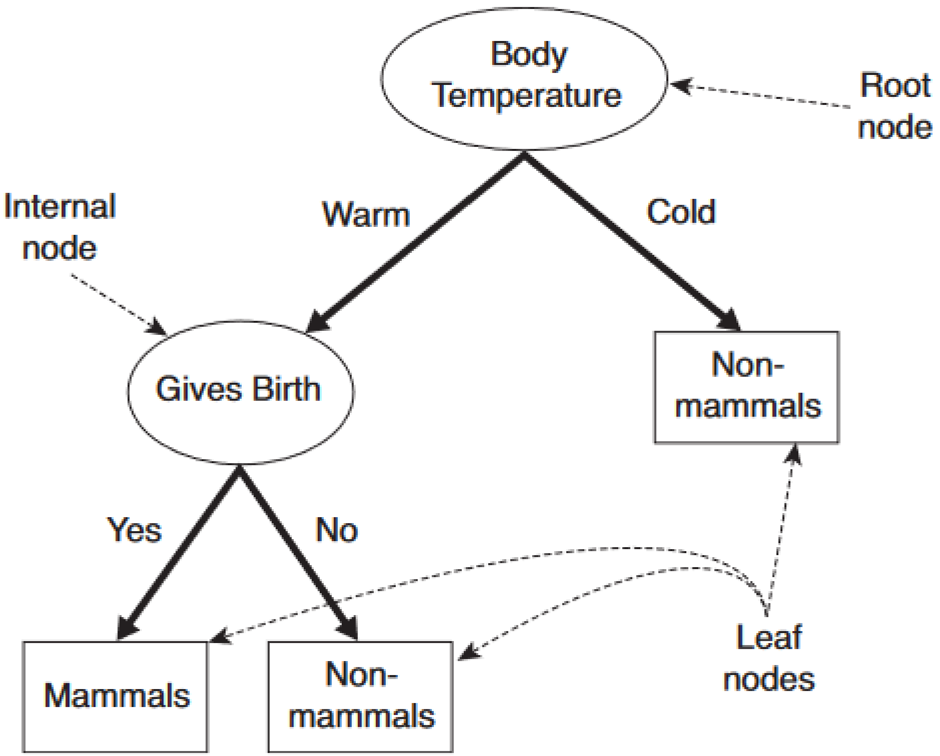


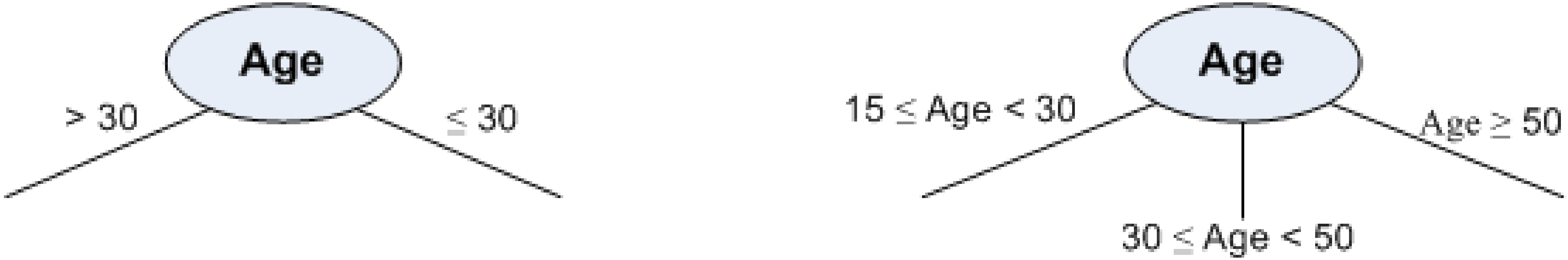

Figure 4.4. A decision tree for the mammal classification problem.

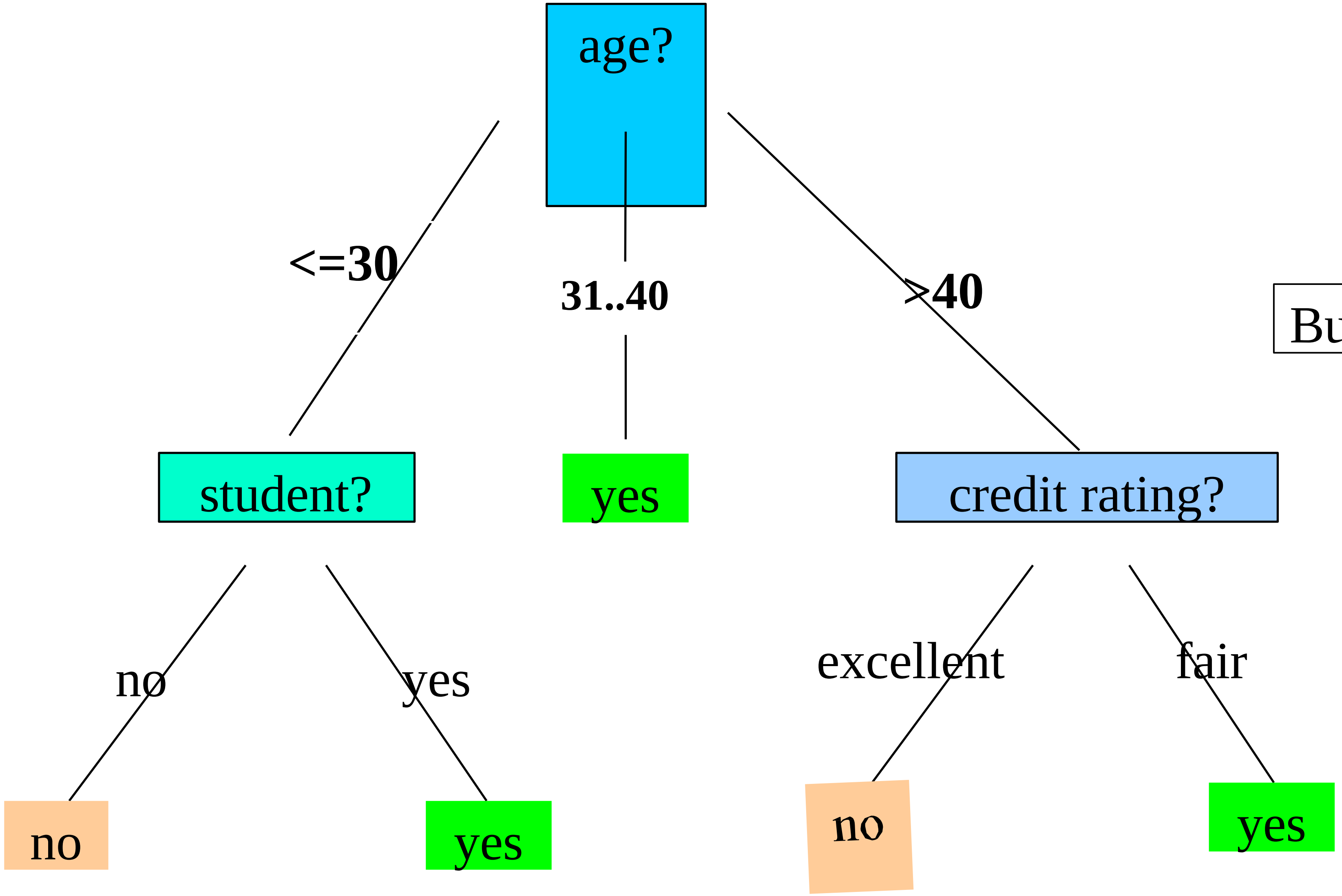
Unlabeled data

| Name | Body temperature | Gives Birth | ... | Class |
|---|---|---|---|---|
| Flamingo | Warm | No | ... | ? |

Class = Non-Mammals

# Numerical attribute






Buys computer/not?

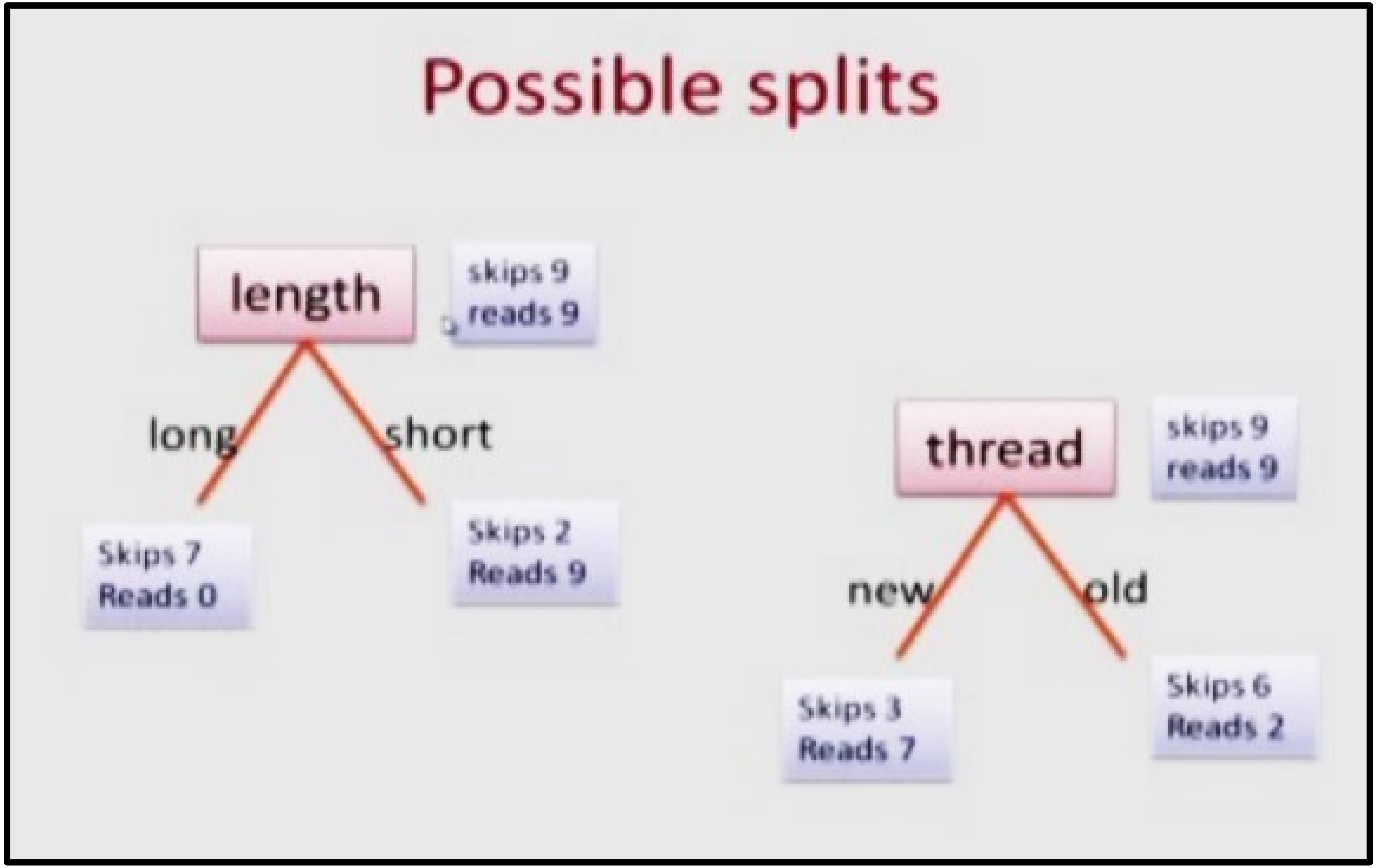
# Example data

Training Examples:

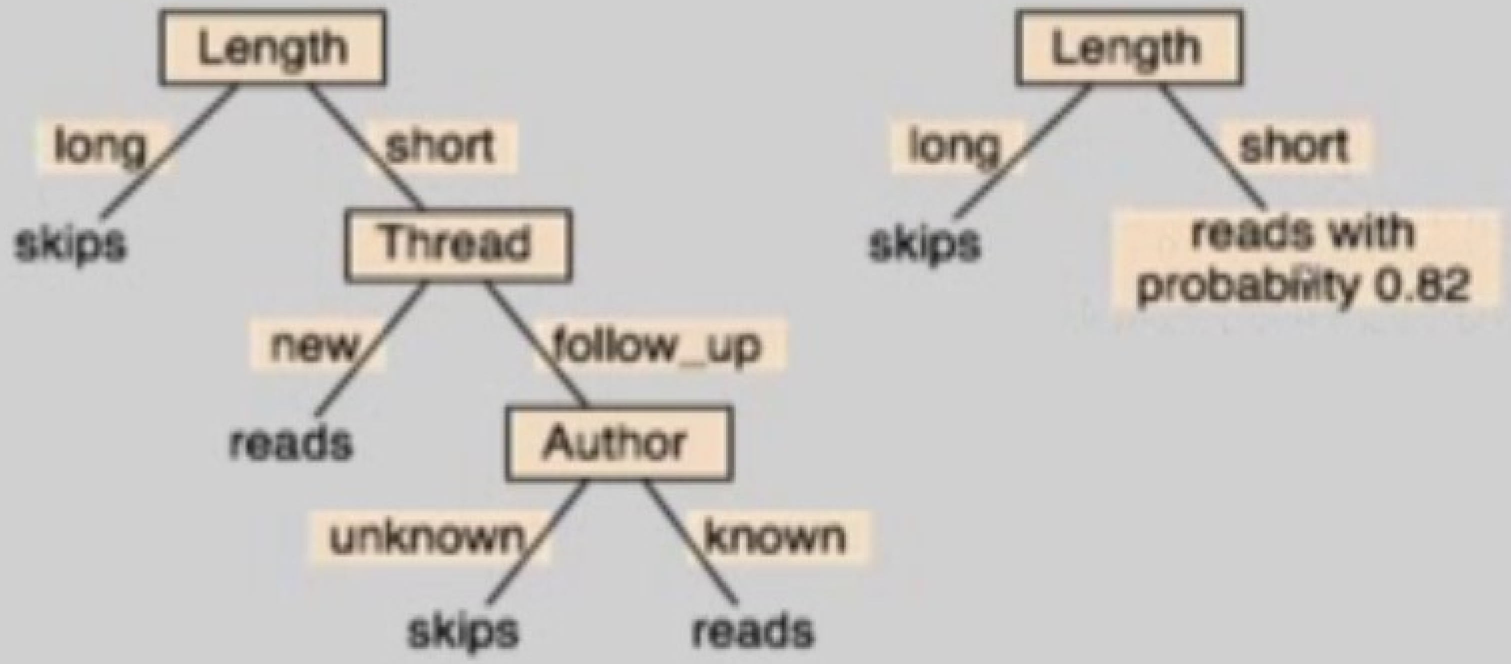
| | Action | Author | Thread | Length | Where |
|---|---|---|---|---|---|
| e1 | skips | known | new | long | Home |
| e2 | reads | unknown | new | short | Work |
| e3 | skips | unknown | old | long | Work |
| e4 | skips | known | old | long | home |
| e5 | reads | known | new | short | home |
| e6 | skips | known | old | long | work |

New Examples:

| | | | | | |
|---|---|---|---|---|---|
| e7 | ??? | known | new | short | work |
| e8 | ??? | unknown | new | short | work |


Possible splits
length
skips 9
reads 9
long
short
Skips 7
Reads 0
Skips 2
Reads 9
thread
skips 9
reads 9
new
old
Skips 3
Reads 7
Skips 6
Reads 2

# Two Example DTs



# Basic Algorithm for Top-Down Induction of Decision Trees

[ID3, C4.5 by Quinlan]

*node* = root of decision tree

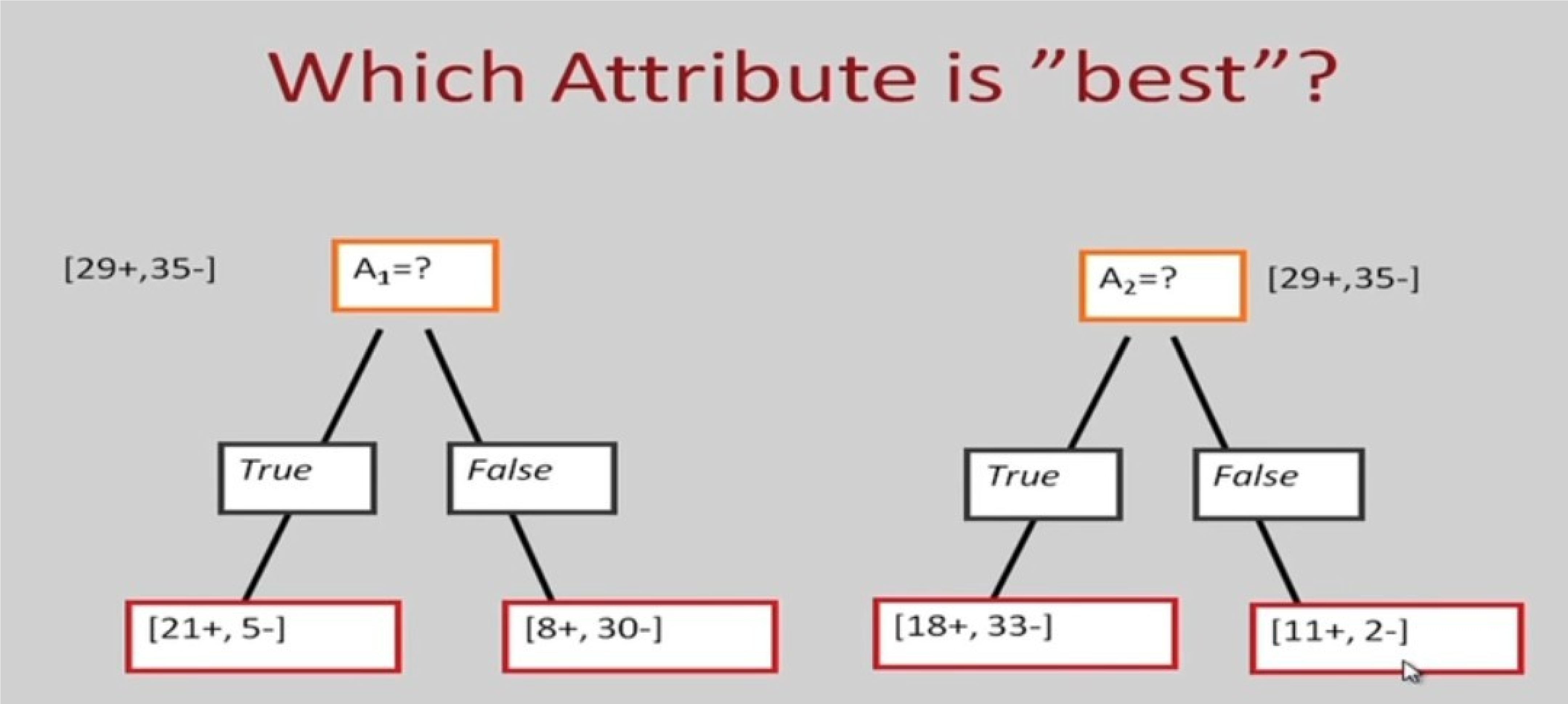
Main loop:

1. $A \leftarrow$ the "best" decision attribute for the next 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 leaf nodes.

How do we choose which attribute is best?

# Choices

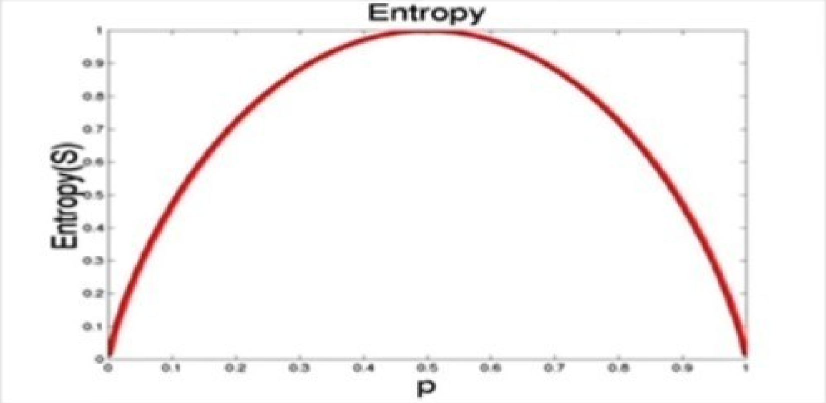
- When to stop
  - no more input features
  - all examples are classified the same
  - too few examples to make an informative split
- Which test to split on
  - split gives smallest error.
  - With multi-valued features
    - split on all values or
    - split values into half.


Which Attribute is "best"?
[29+,35-]
A1=?
True
False
[21+, 5-]
[8+, 30-]
A2=?
[29+,35-]
True
False
[18+, 33-]
[11+, 2-]

# Principled Criterion

- Selection of an attribute to test at each node - choosing the most useful attribute for classifying examples.
- information gain
  - measures how well a given attribute separates the training examples according to their target classification
  - This measure is used to select among the candidate attributes at each step while growing the tree
  - Gain is measure of how much we can reduce uncertainty (Value lies between 0,1)

# Entropy



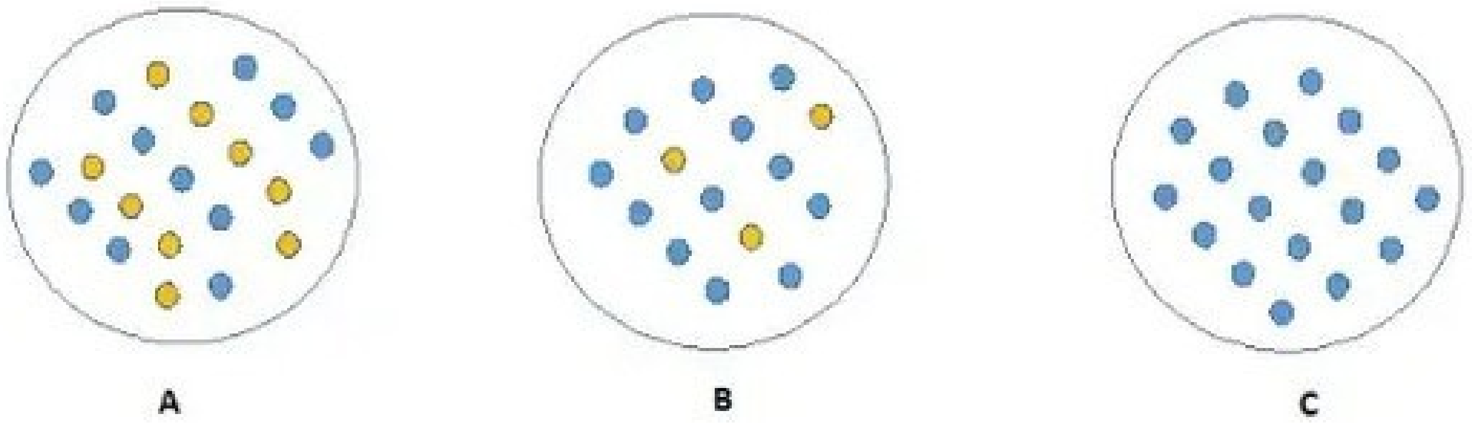

- The entropy is 0 if the outcome is ``certain".
- The entropy is maximum if we have no knowledge of the system (or any outcome is equally possible).

- S is a sample of training examples
- $p_+$ is the proportion of positive examples
- $p_-$ is the proportion of negative examples
- Entropy measures the impurity of S

$$\text{Entropy}(S) = -p_+\log_2 p_+ - p_-\log_2 p_-$$

# How to choose best decision node

Which node can be described easily?



- Less impure node requires less information to describe it.
- More impure node requires more information.

===> Information theory is a measure to define this degree of disorganization in a system known as **Entropy**.

- **Entropy** is **0** if all the members of S belong to the same class.
- **Entropy** is **1** when the collection contains an equal no. of +ve and -ve examples.
- **Entropy** is **between 0 and 1** if the collection contains unequal no. of +ve and -ve examples.

# Information Gain

Gain(S,A): expected reduction in entropy due to partitioning S on attribute A

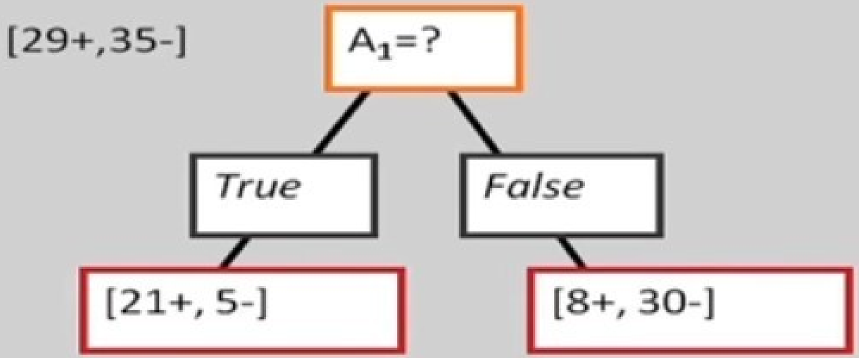
$$Gain(S,A)=Entropy(S) - \sum_{v \in values(A)} |S_v|/|S| \; Entropy(S_v)$$

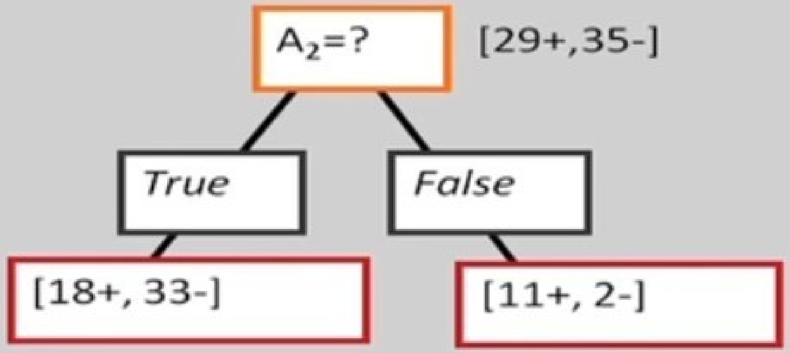
$$Entropy([29+,35-]) = -29/64 \log_2 29/64 - 35/64 \log_2 35/64$$
$$= 0.99$$

# Information Gain

Entropy([21+,5-]) = 0.71
Entropy([8+,30-]) = 0.74
Gain(S,$A_1$)=Entropy(S)
-26/64*Entropy([21+,5-])
-38/64*Entropy([8+,30-])
=0.27

Entropy([18+,33-]) = 0.94
Entropy([8+,30-]) = 0.62
Gain(S,$A_2$)=Entropy(S)
-51/64*Entropy([18+,33-])
-13/64*Entropy([11+,2-])
=0.12




ICS320 9



AMRITA VISHWA VIDYAPEETHAM

# Exampl

| age | income | student | credit_rating | buys_computer |
|---|---|---|---|---|
| <=30 | high | no | fair | no |
| <=30 | high | no | excellent | no |
| 31…40 | high | no | fair | yes |
| >40 | medium | no | fair | yes |
| >40 | low | yes | fair | yes |
| >40 | low | yes | excellent | no |
| 31…40 | low | yes | excellent | yes |
| <=30 | medium | no | fair | no |
| <=30 | low | yes | fair | yes |
| >40 | medium | yes | fair | yes |
| <=30 | medium | yes | excellent | yes |
| 31…40 | medium | no | excellent | yes |
| 31…40 | high | yes | fair | yes |
| >40 | medium | no | excellent | no |

$$\text{Entropy}(S) = -\frac{9}{14}\log_2\frac{9}{14} - \frac{5}{14}\log_2\frac{5}{14}$$

$$= 0.940$$

- Class P: buys_computer = “yes”
- Class N: buys_computer = “no”

Age

Gain$(S, Age)$ $\quad Gain(S, A) \equiv Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$

$$= Entropy(S) - \frac{5}{14} Entropy([2+, 3-])$$

$$- \frac{4}{14} Entropy[4+, 0-] \quad - \frac{5}{14} Entropy[(3^+, 2^-)]$$

$$= 0.94 - \frac{5}{14}\left[\frac{-2}{5} \log_2 \frac{2}{5} - \frac{3}{5} \log_2 \frac{3}{5}\right]$$

$$- \frac{4}{14}\left[\frac{-4}{4} \log_2 \frac{4}{4}\right] - \frac{5}{14}\left[\frac{-3}{5} \log_2 \frac{2}{5} - \frac{2}{5} \log_2 \frac{2}{5}\right]$$

$$= 0.94 - \frac{5}{14}[0.968] - \frac{4}{14}[0] - \frac{5}{14}[0.968$$

$$= 0.94 - 0.36 \cancel{0.568} - 0.36 = \underline{\underline{0.25}}$$

$$Gain(age) = 0.25$$
$$Gain(income) = 0.029$$
$$Gain(student) = 0.151$$
$$Gain(credit\_rating) = 0.048$$

Age has maximum information gain. So age is selected as the best node to split . So age is selected as root node

# Building The Tree: we choose "age" as a root

age

<=30

| income | student | credit | class |
|---|---|---|---|
| high | no | fair | no |
| high | no | excellent | no |
| medium | no | fair | no |
| low | yes | fair | yes |
| medium | yes | excellent | yes |

>40

| income | student | credit | class |
|---|---|---|---|
| medium | no | fair | yes |
| low | yes | fair | yes |
| low | yes | excellent | no |
| medium | yes | fair | yes |
| medium | no | excellent | no |

31…40

| income | student | credit | class |
|---|---|---|---|
| high | no | fair | yes |
| low | yes | excellent | yes |
| medium | no | excellent | yes |
| high | yes | fair | yes |

# Building The Tree: “age” as the root

age

<=30

| income | student | credit | class |
|---|---|---|---|
| high | no | fair | no |
| high | no | excellent | no |
| medium | no | fair | no |
| low | yes | fair | yes |
| medium | yes | excellent | yes |

>40

| income | student | credit | class |
|---|---|---|---|
| medium | no | fair | yes |
| low | yes | fair | yes |
| low | yes | excellent | no |
| medium | yes | fair | yes |
| medium | no | excellent | no |

31…40

class=yes

age

<=30

| income | student | credit | class |
|---|---|---|---|
| high | no | fair | no |
| high | no | excellent | no |
| medium | no | fair | no |
| low | yes | fair | yes |
| medium | yes | excellent | yes |

$S \rightarrow age <= 30 \quad [3^{+}, 2^{-}]$

$$E(S_{age}) = \frac{-3}{5} \log \frac{3}{5} - \frac{2}{5} \log \frac{2}{5}$$

$$= 0.968$$

$$Gain(S_{age}, income) = E(S_{age}) - \frac{2}{5} E([0^{+}, 2^{-}]) - \frac{2}{5} E([1^{+}, 1^{-}]) - \frac{1}{5} E([1^{+}, 0^{-}])$$

$$= 0.968 - 0 - \frac{2}{5}\left[-\frac{1}{2} \log \frac{1}{2} - \frac{1}{2} \log \frac{1}{2}\right] - 0$$

$$= 0.568$$

age

<=30

| income | student | credit | class |
|---|---|---|---|
| high | no | fair | no |
| high | no | excellent | no |
| medium | no | fair | no |
| low | yes | fair | yes |
| medium | yes | excellent | yes |

$Gain(S_{age}, student)$

$= 0.968 - \frac{3}{5}\left[E([0^+, 3^-])\right] - \frac{2}{5}\left[E([2^+, 0^-])\right]$

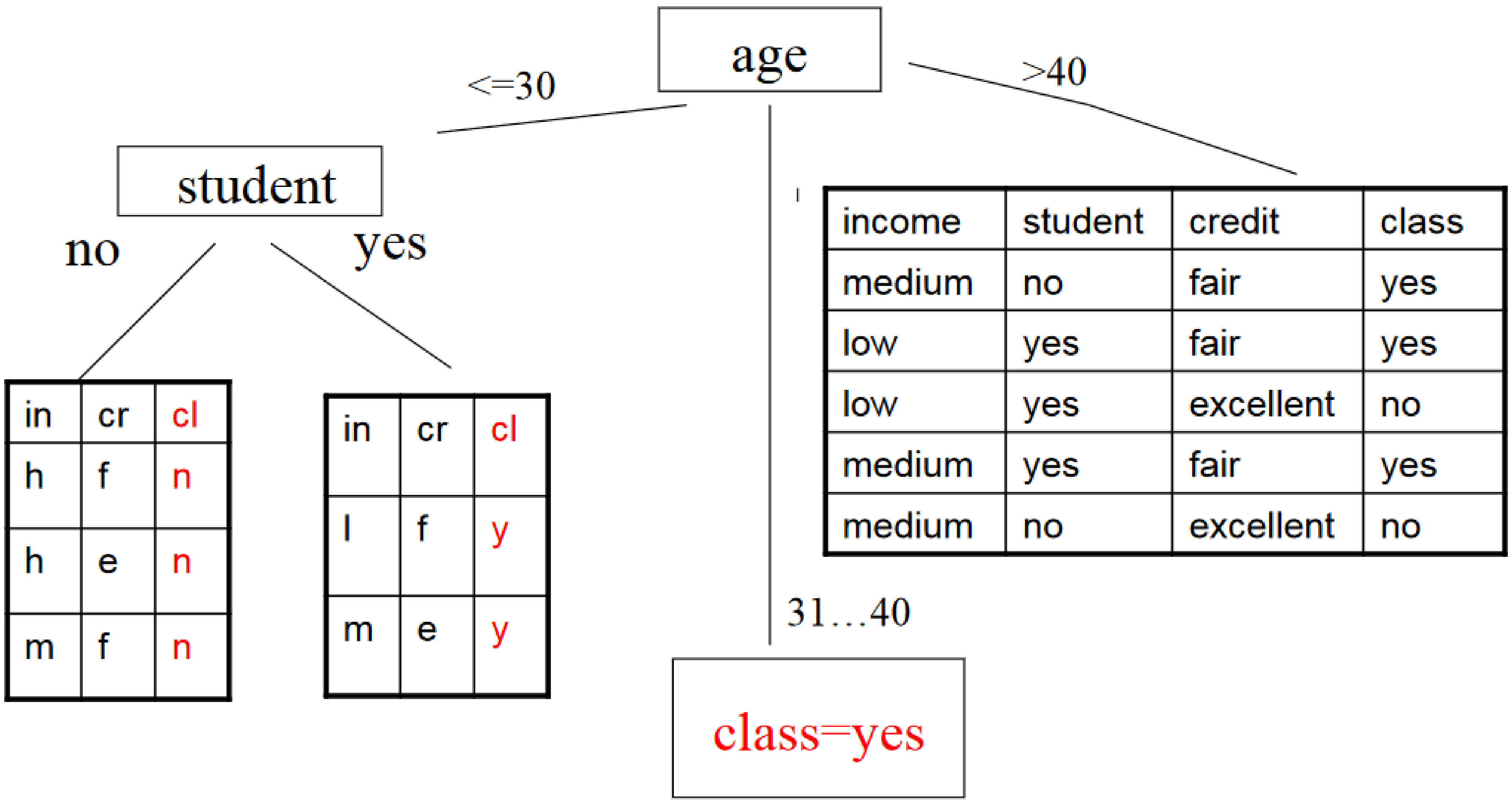
$= \underline{\underline{0.968}}$

$Gain(S_{age}, credit)$

$= 0.968 - \frac{3}{5} E([1^+, 2^-]) - \frac{2}{5} E([1^+, 1^-])$

max for student attribute

# Building The Tree: we chose “student” on <=30 branch

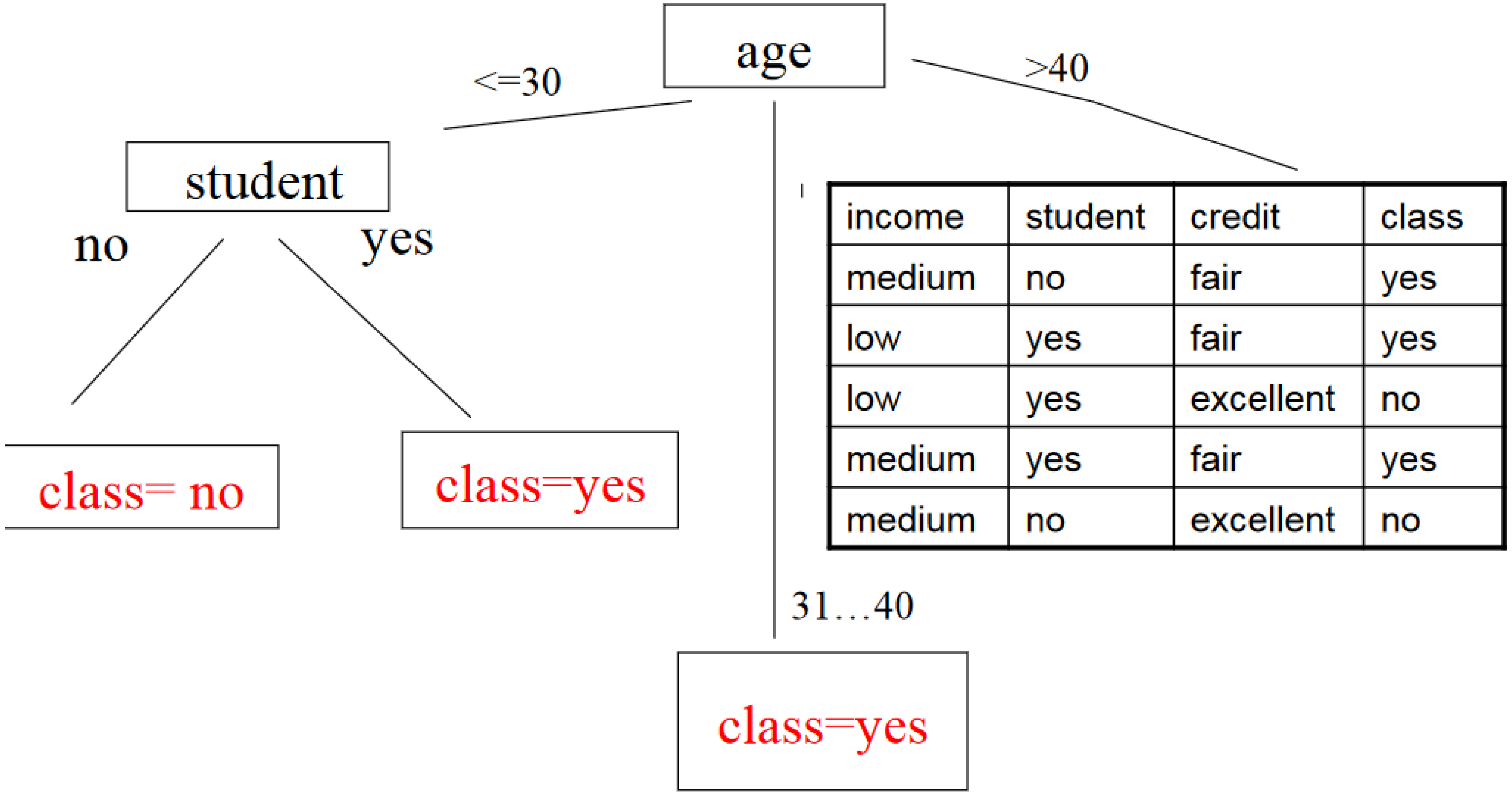



| in | cr | cl |
|---|---|---|
| h | f | n |
| h | e | n |
| m | f | n |

| in | cr | cl |
|---|---|---|
| l | f | y |
| m | e | y |

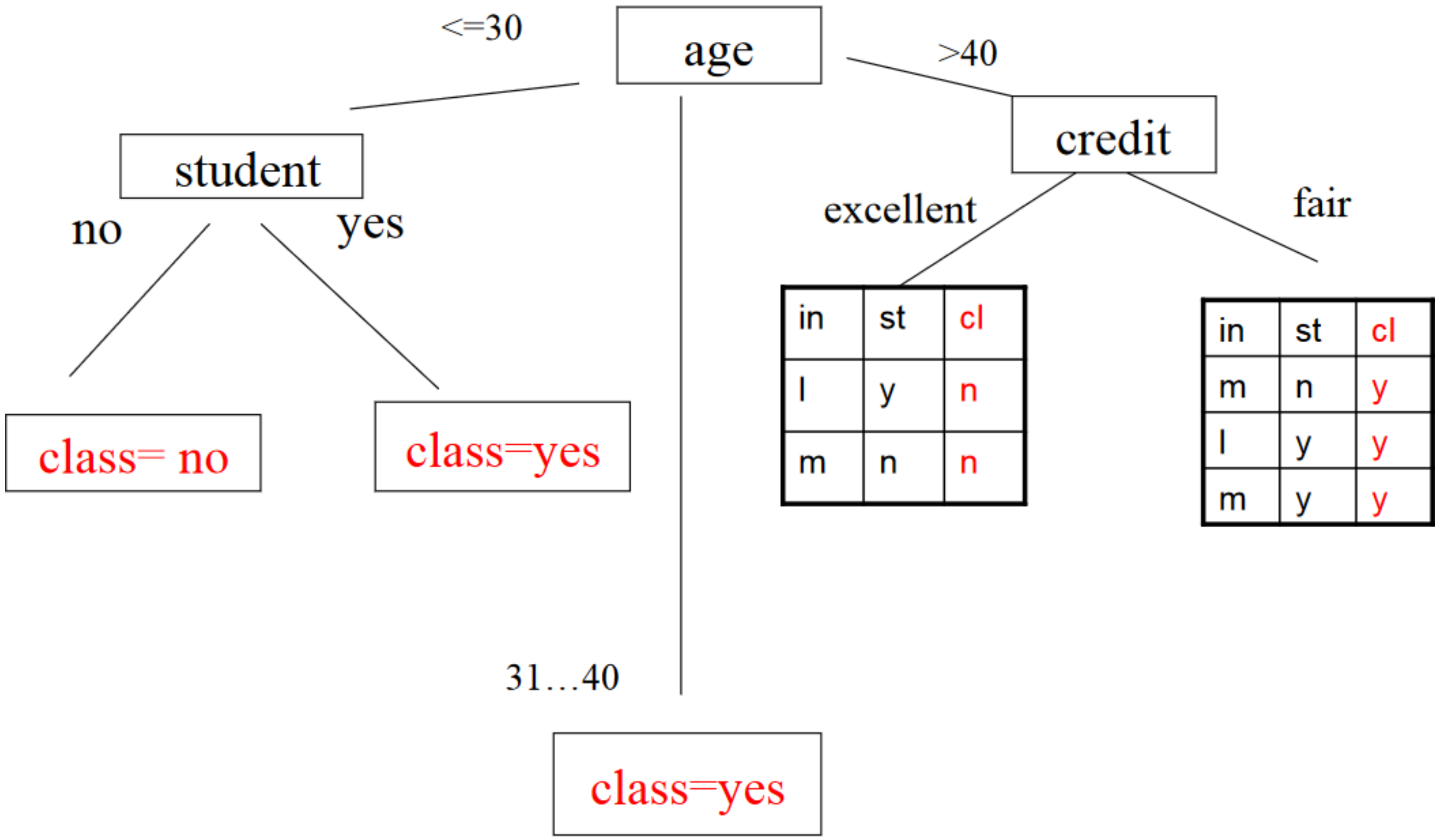
| income | student | credit | class |
|---|---|---|---|
| medium | no | fair | yes |
| low | yes | fair | yes |
| low | yes | excellent | no |
| medium | yes | fair | yes |
| medium | no | excellent | no |

# Building The Tree: we chose "student" on <=30 branch




| income | student | credit | class |
|---|---|---|---|
| medium | no | fair | yes |
| low | yes | fair | yes |
| low | yes | excellent | no |
| medium | yes | fair | yes |
| medium | no | excellent | no |

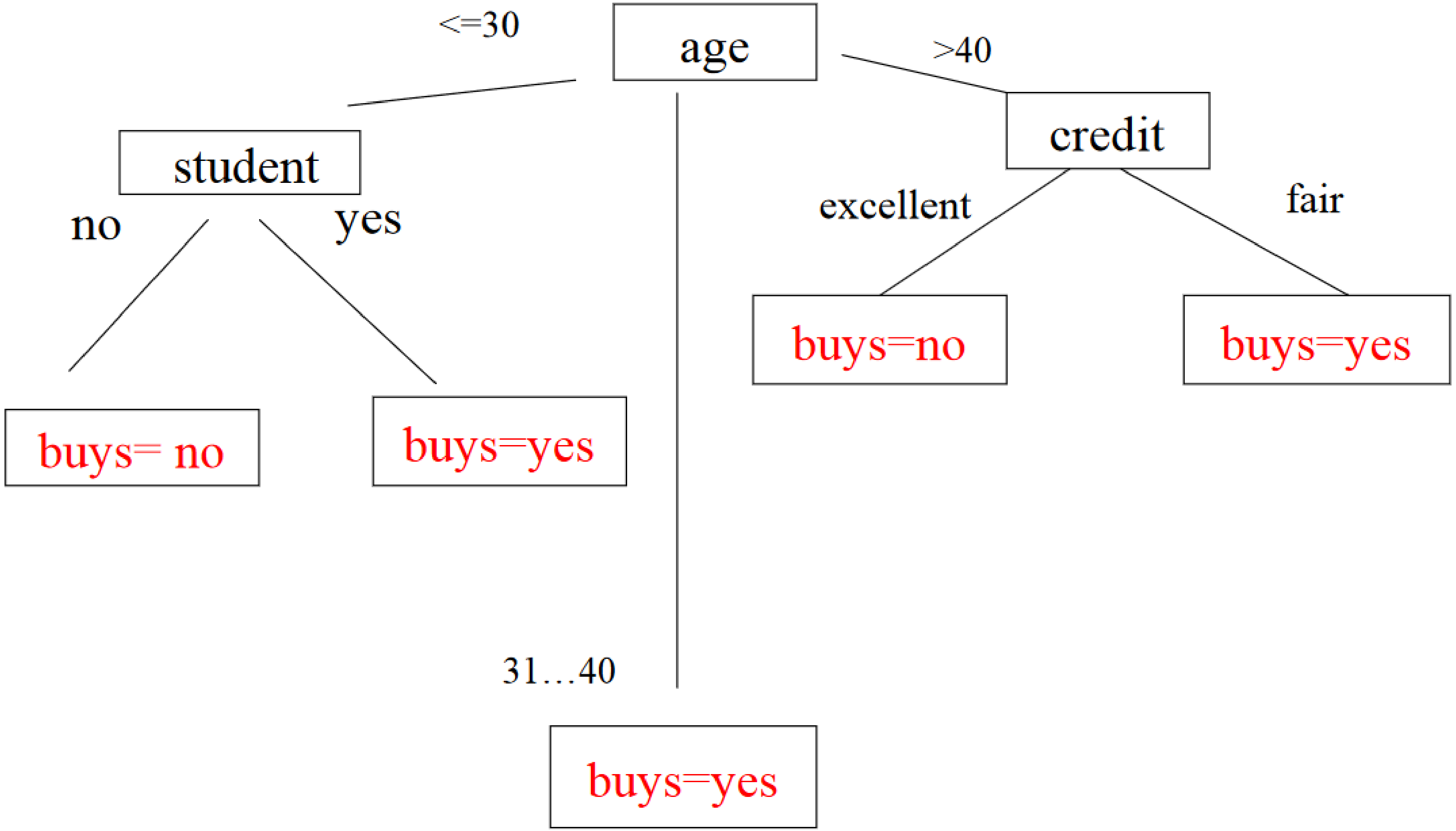
# Building The Tree: we chose “credit” on >40 branch




| in | st | cl |
|---|---|---|
| l | y | n |
| m | n | n |

| in | st | cl |
|---|---|---|
| m | n | y |
| l | y | y |
| m | y | y |

# Final tree


age
<=30
>40
student
no
yes
buys= no
buys=yes
31…40
buys=yes
credit
excellent
fair
buys=no
buys=yes

# Rules extracted from the tree

- The rules are:

IF *age* = "<=30" AND *student* = "*no*" THEN
*buys_computer* = "*no*"

IF *age* = "<=30" AND *student* = "*yes*" THEN
*buys_computer* = "*yes*"

IF *age* = "31…40" THEN
*buys_computer* = "*yes*"

IF *age* = ">40" AND *credit_rating* = "*excellent*" THEN
*buys_computer* = "no"

IF *age* = ">40" AND *credit_rating* = "*fair*" THEN
*buys_computer* = "yes"

# Inductive Bias

- Shorter trees are preferred over larger trees

# Overfitting and Tree Pruning

- Overfitting: An induced tree may overfit the training data
  - Too many branches, some may reflect anomalies due to noise or outliers
  - Poor accuracy for unseen samples
- Two approaches to avoid overfitting
  - Prepruning: *Halt tree construction early*-do not split a node if this would result in the goodness measure falling below a threshold
    - Difficult to choose an appropriate threshold
  - Postpruning: *Remove branches* from a "fully grown" tree—get a sequence of progressively pruned trees
    - Use a set of data different from the training data to decide which is the "best pruned tree"

# Attribute Selection Measures

- **Information gain**:
  - biased towards multivalued attributes
- **Gain ratio**:

$$GainRatio(S, A) \equiv \frac{Gain(S, A)}{SplitInformation(S, A)}$$

$$SplitInformation(S, A) \equiv -\sum_{i=1}^{c} \frac{|S_i|}{|S|} \log_2 \frac{|S_i|}{|S|}$$

  - tends to prefer unbalanced splits in which one partition is much smaller than the others

- **Gini index:**

$$gini(D) = 1 - \sum_{j=1}^{n} p_j^2$$

  - where pj is the relative frequency of class j in D

  - Choose attribute with low gini index
  - has difficulty when # of classes is large
  - tends to favor tests that result in equal-sized partitions and purity in both partitions

# Decision tree suited when -

- Instances are represented by attribute-value pairs
- The target function has discrete output values
- The training data may contain errors.
- The training data may contain missing attribute values

# Thank you