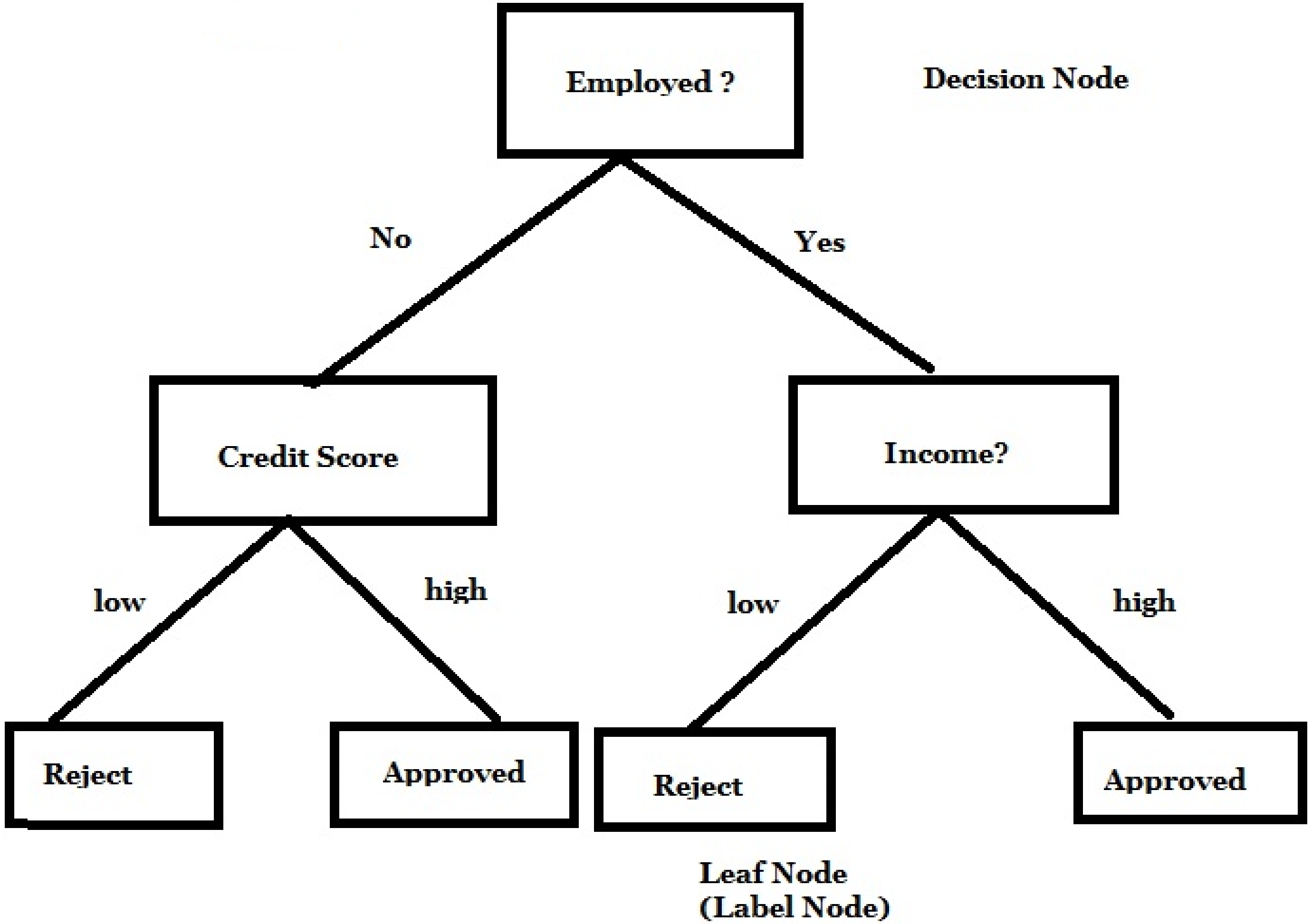
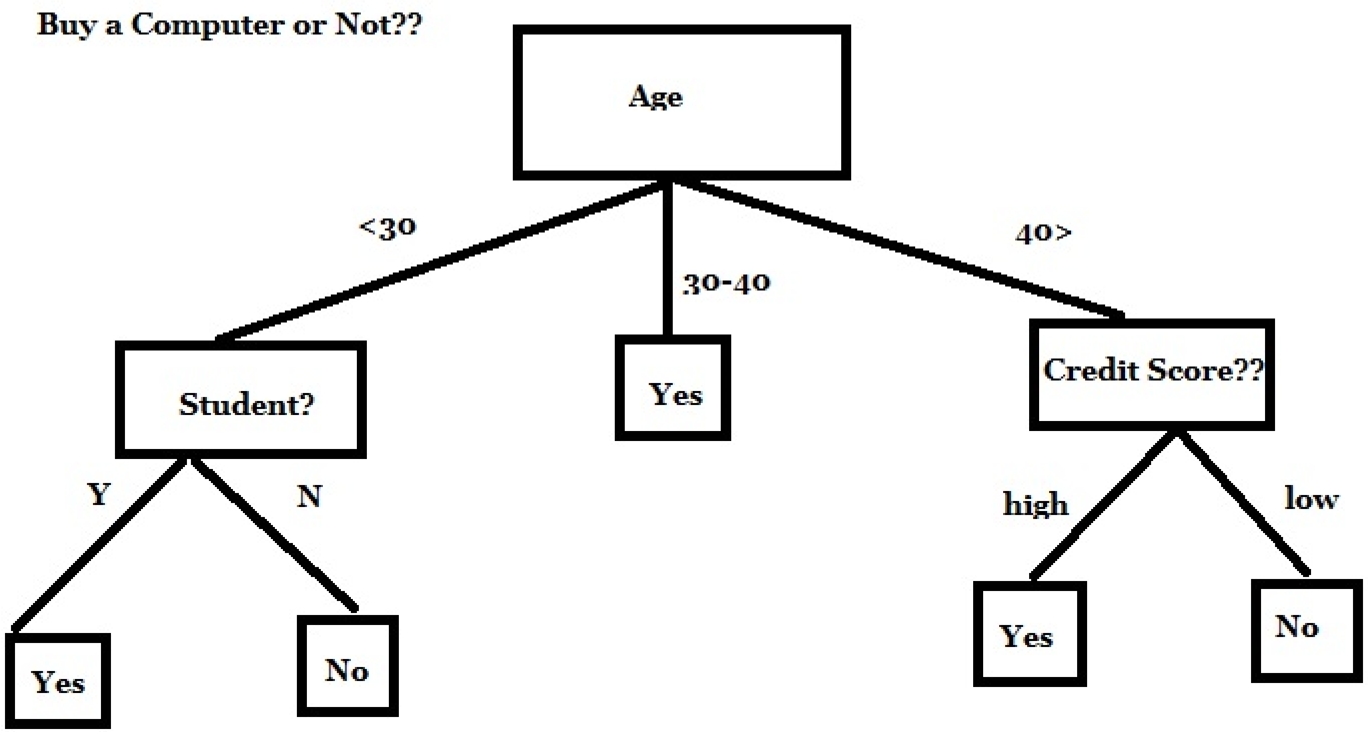
Decision Tree

- Graphical representation of all the possible solutions to a decision.
- Decision are based on some conditions.
- Decision made can be easily explained.

Loan Approved or Rejected ?



Buy a Computer or Not?



Training Examples:

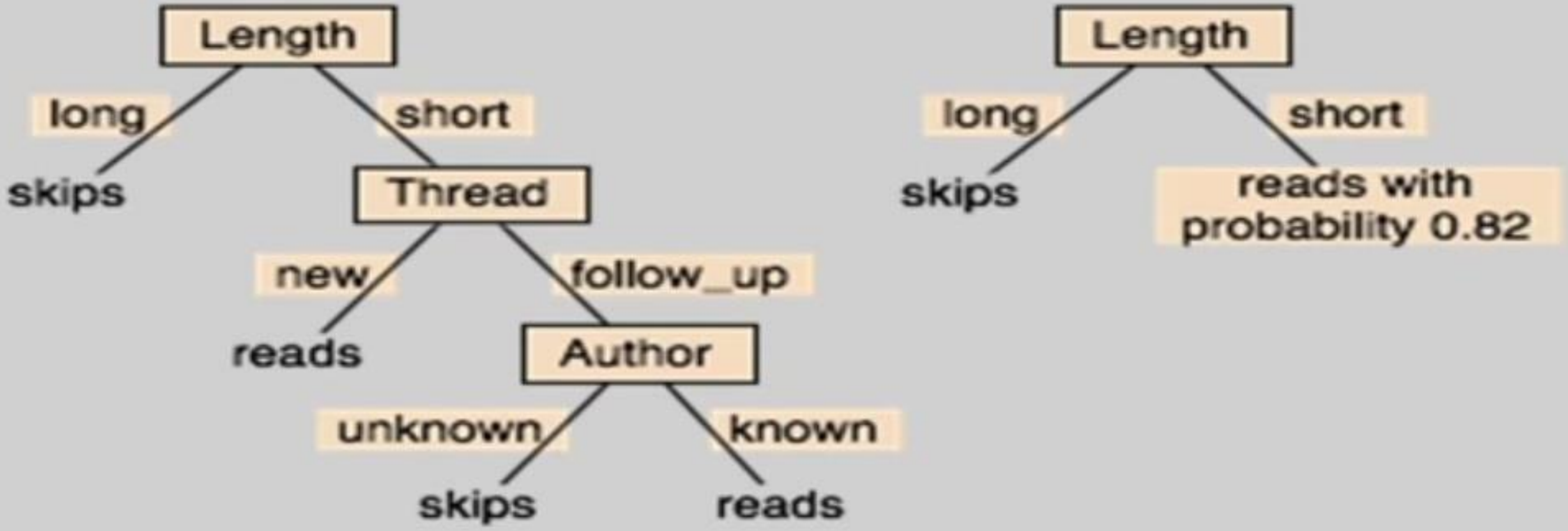
	Action	Author	Thread	Length	Where
e1	skips	known	new	long	Home
e2	reads	unknown	new	short	Work
e3	skips	unknown	old	long	Work
e4	skips	known	old	long	home
e5	reads	known	new	short	home
e6	skips	known	old	long	work

New Examples:

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

Possible splits

Two Example DTs



Decision Tree for PlayTennis

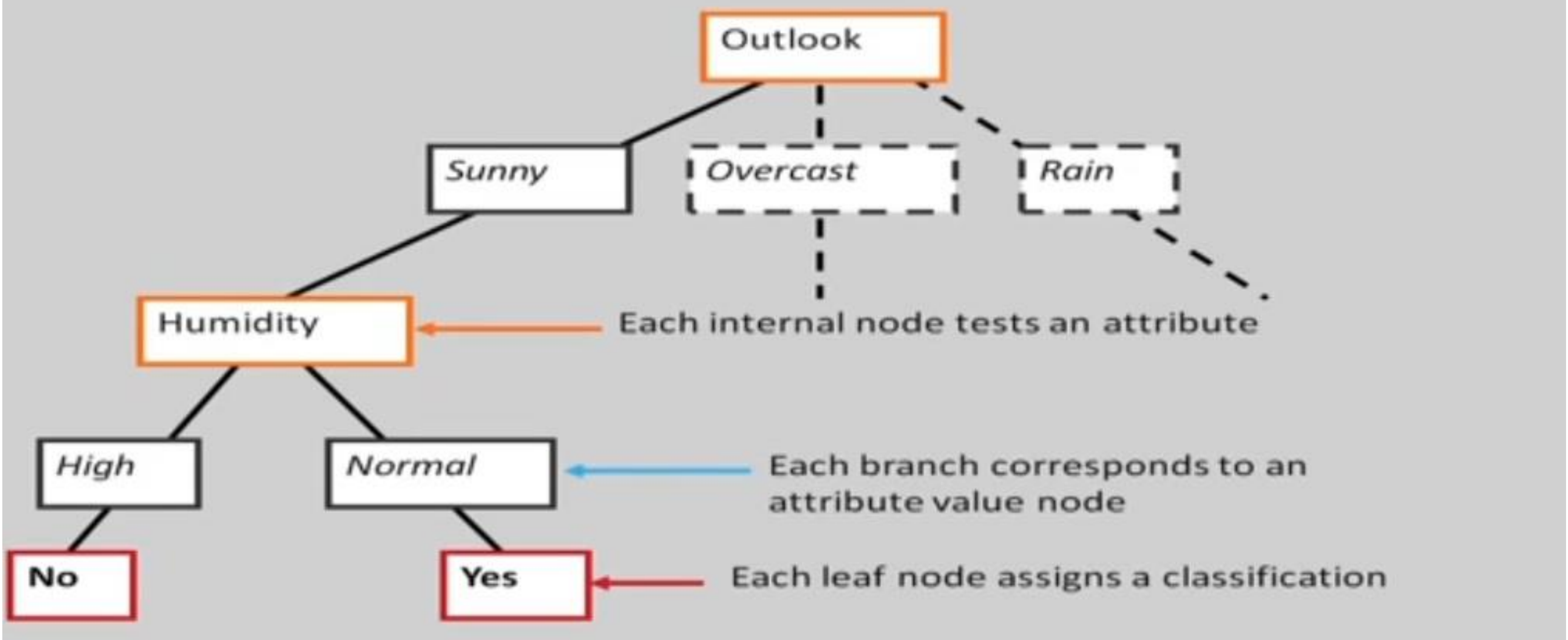
- Attributes and their values:
  - Outlook: *Sunny, Overcast, Rain*
  - Humidity: *High, Normal*
  - Wind: *Strong, Weak*
  - Temperature: *Hot, Mild, Cool*

- Target concept - Play Tennis: *Yes, No*

Training Examples

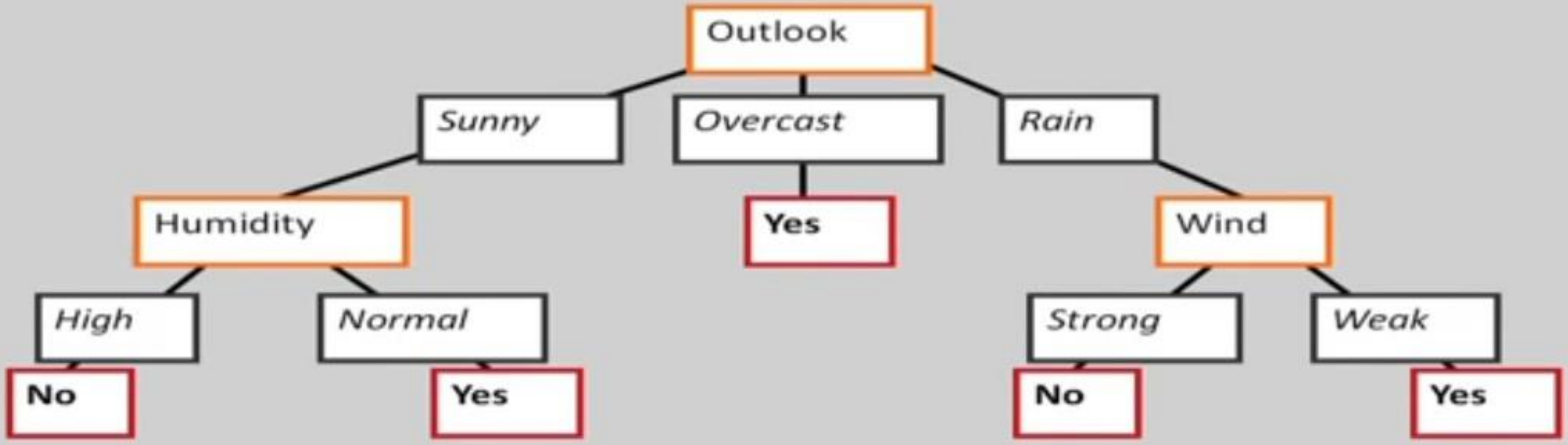
Day	Outlook	Temp	Humidity	Wind	Tennis?
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Decision Tree for PlayTennis



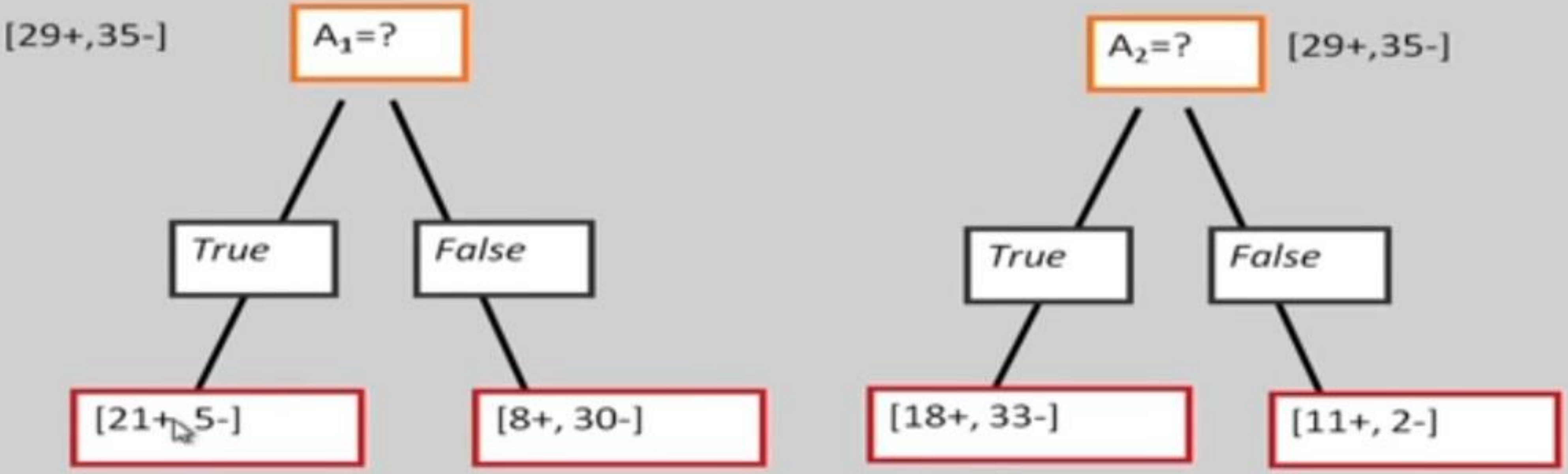
Decision Tree

decision trees represent disjunctions of conjunctions



(Outlook=Sunny $\wedge$ Humidity=Normal)
$\vee$ (Outlook=Overcast)
$\vee$ (Outlook=Rain $\wedge$ Wind=Weak)

Which Attribute is ”best”?



Principled Criterion

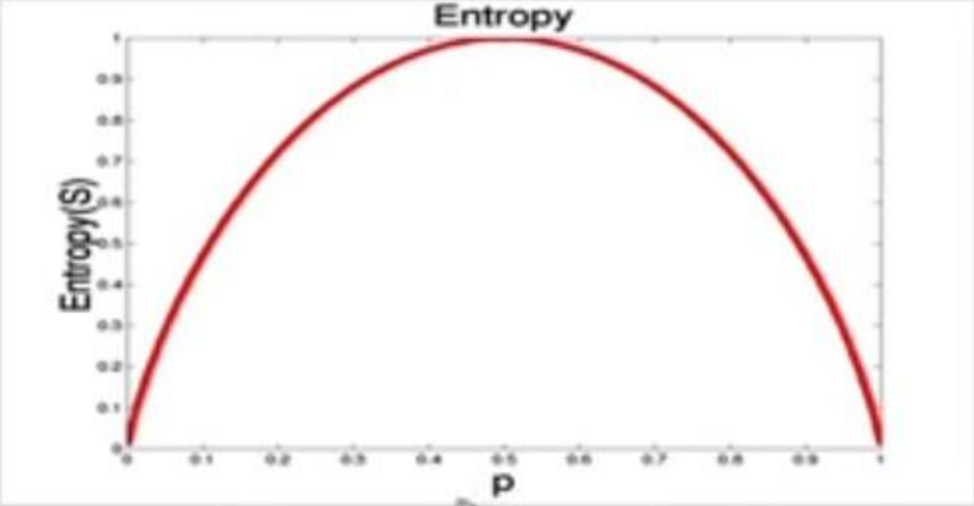
- Selection of an attribute to test at each node - choosing the most useful attribute for classifying examples.
- information gain
  - measures how well a given attribute separates the training examples according to their target classification
  - This measure is used to select among the candidate attributes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

Entropy

- A measure for
  - uncertainty
  - purity
  - information content
- Information theory: optimal length code assigns $(-\log_2 p)$ bits to message having probability p
- S is a sample of training examples
  - p_+ is the proportion of positive examples in S
  - p_- is the proportion of negative examples in S
- Entropy of S: average optimal number of bits to encode information about certainty/uncertainty about S

$$Entropy(S) = p_+(-\log_2 p_+) + p_-(-\log_2 p_-) = -p_+\log_2 p_+ - p_-\log_2 p_-$$

Entropy



- 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

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

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

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

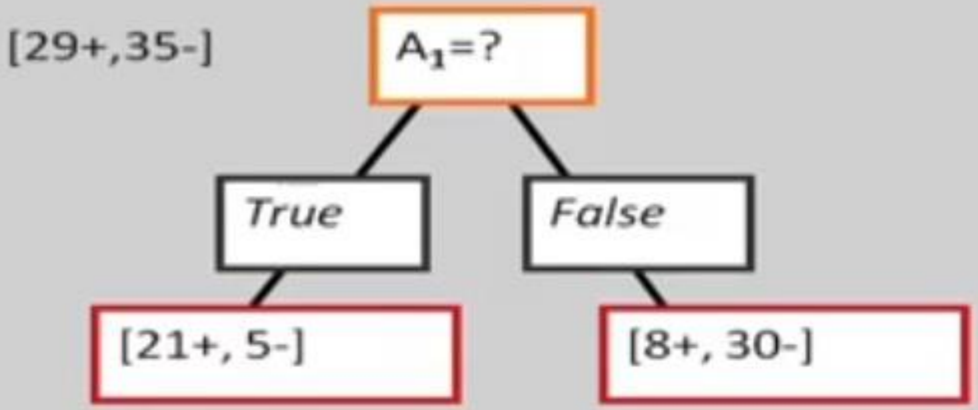
$$Gain(S,A)=Entropy(S) - \sum_{v \in values(A)} |S_v|/|S| \; Entropy(S_v)$$

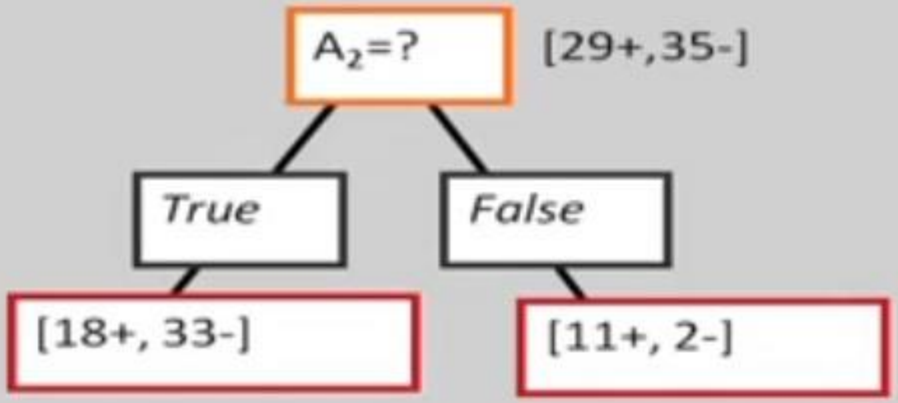
$$Entropy([29+,35-]) = -29/64 \log_2 29/64 - 35/64 \log_2 35/64 = 0.99$$

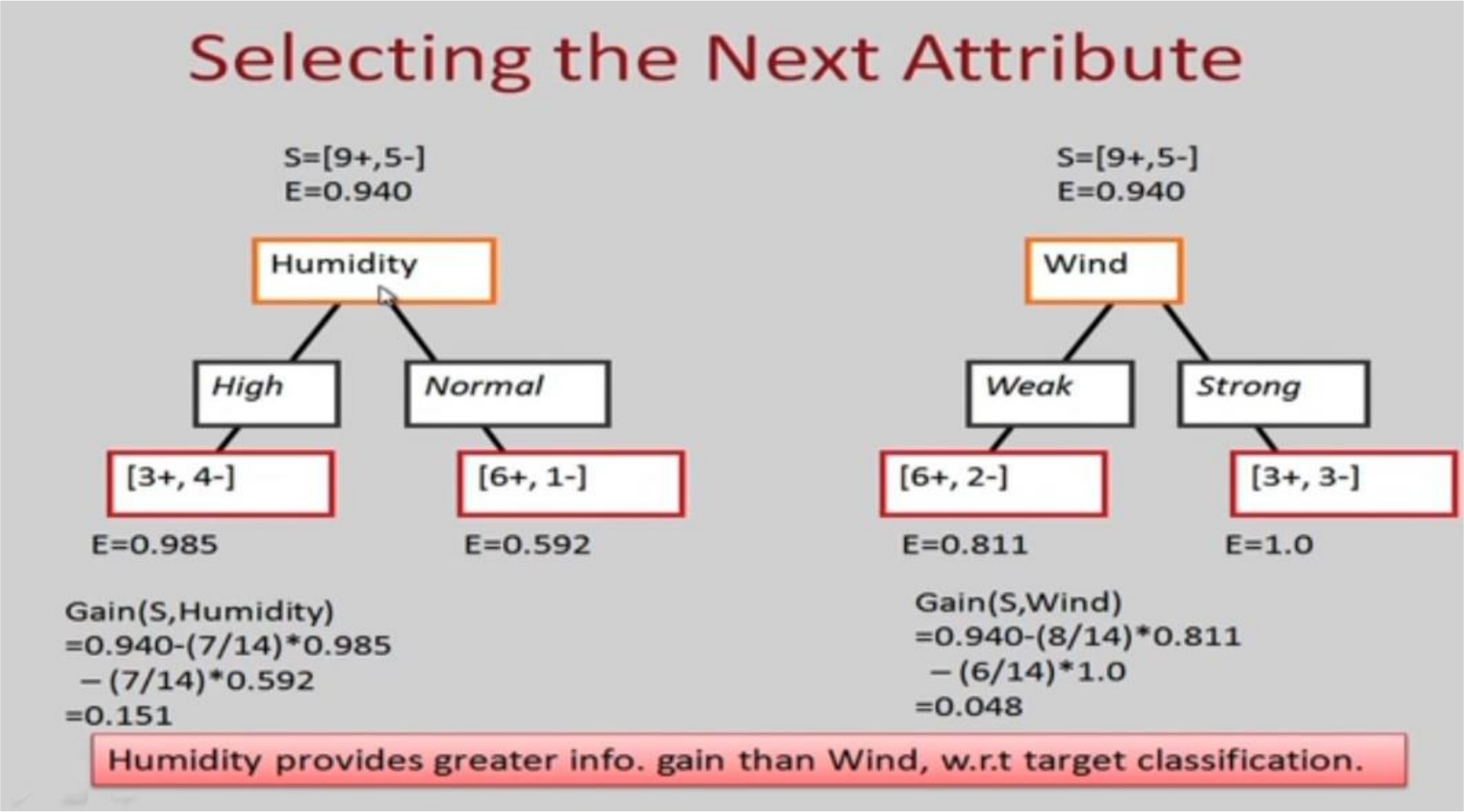
Information Gain

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

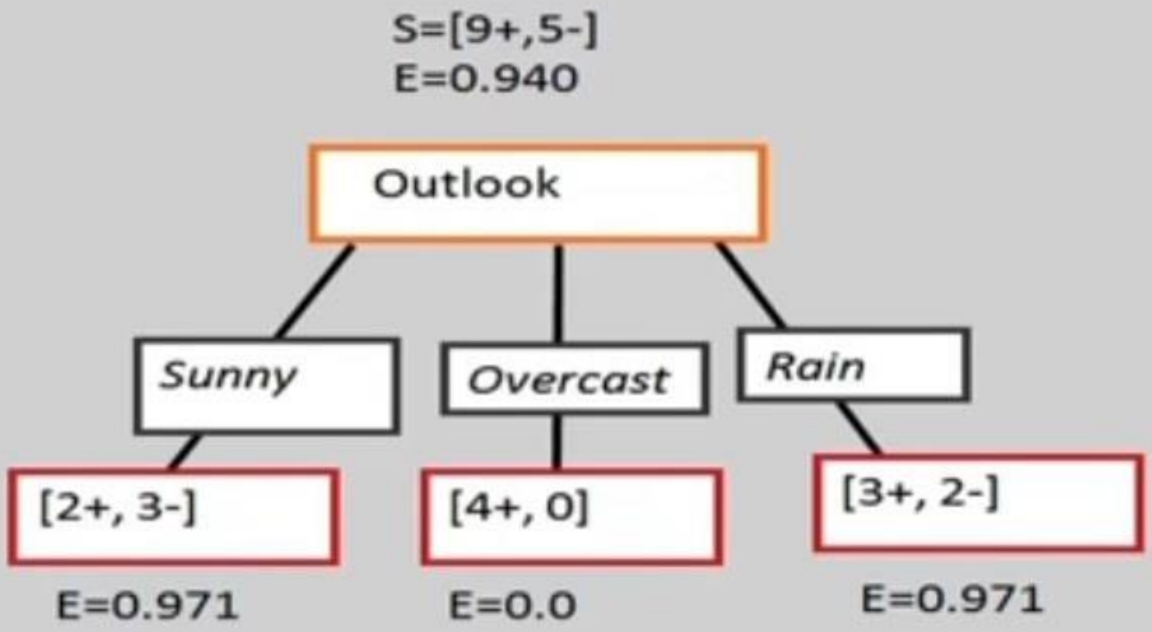
Entropy([18+,33-]) = 0.94
Entropy([8+,30-]) = 0.62
Gain(S,A_2)=Entropy(S)
-51/64*Entropy([18+,33-])
-13/64*Entropy([11+,2-])
=0.12






Selecting the Next Attribute
S=[9+,5-]
E=0.940
Humidity
High
Normal
[3+, 4-]
[6+, 1-]
E=0.985
E=0.592
Gain(S,Humidity)
=0.940-(7/14)*0.985
–(7/14)*0.592
=0.151
S=[9+,5-]
E=0.940
Wind
Weak
Strong
[6+, 2-]
[3+, 3-]
E=0.811
E=1.0
Gain(S,Wind)
=0.940-(8/14)*0.811
–(6/14)*1.0
=0.048
Humidity provides greater info. gain than Wind, w.r.t target classification.

Selecting the Next Attribute



Gain(S,Outlook)
=0.940-(5/14)*0.971
-(4/14)*0.0 – (5/14)*0.0971
=0.247

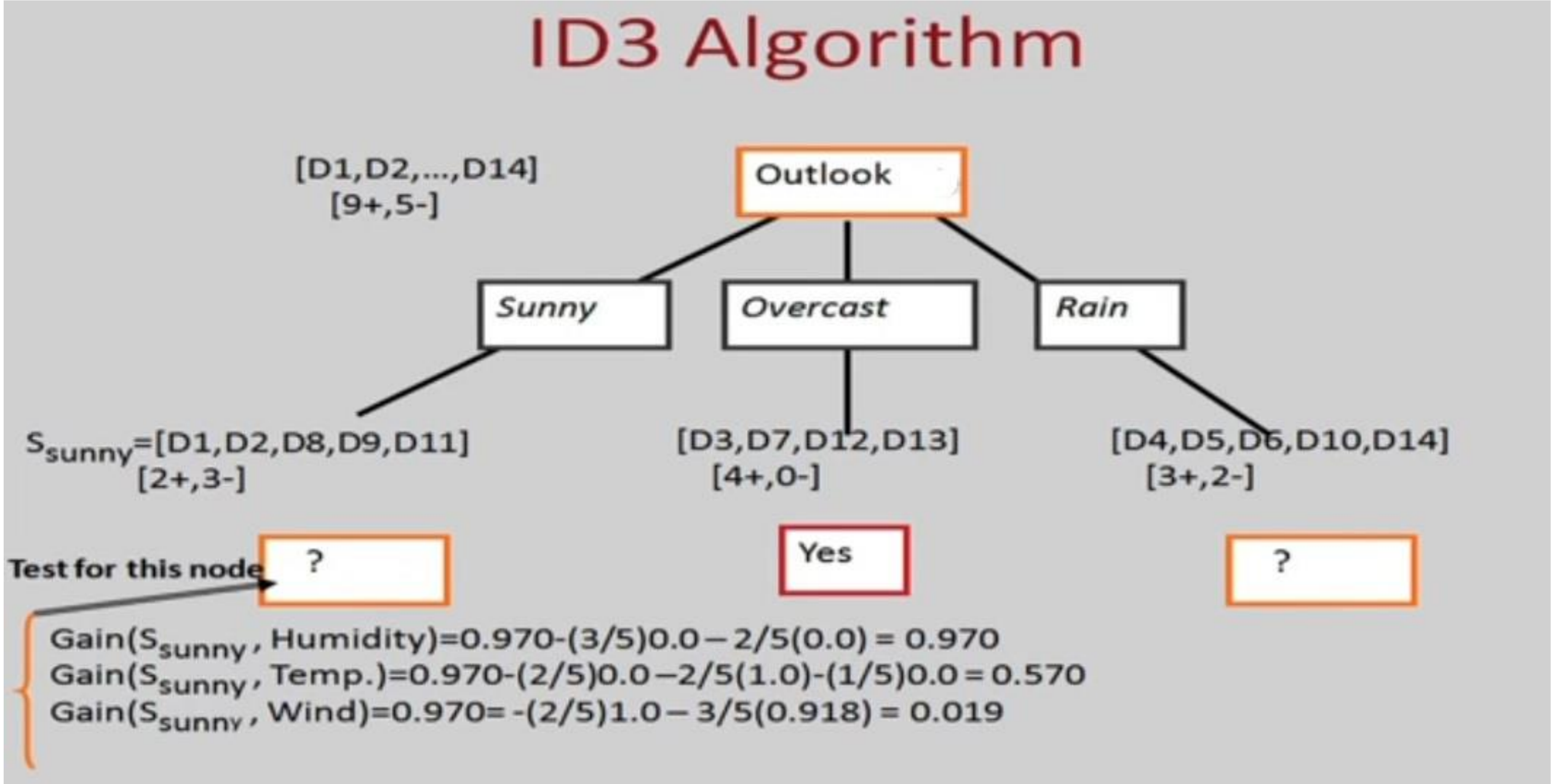
Selecting the Next Attribute

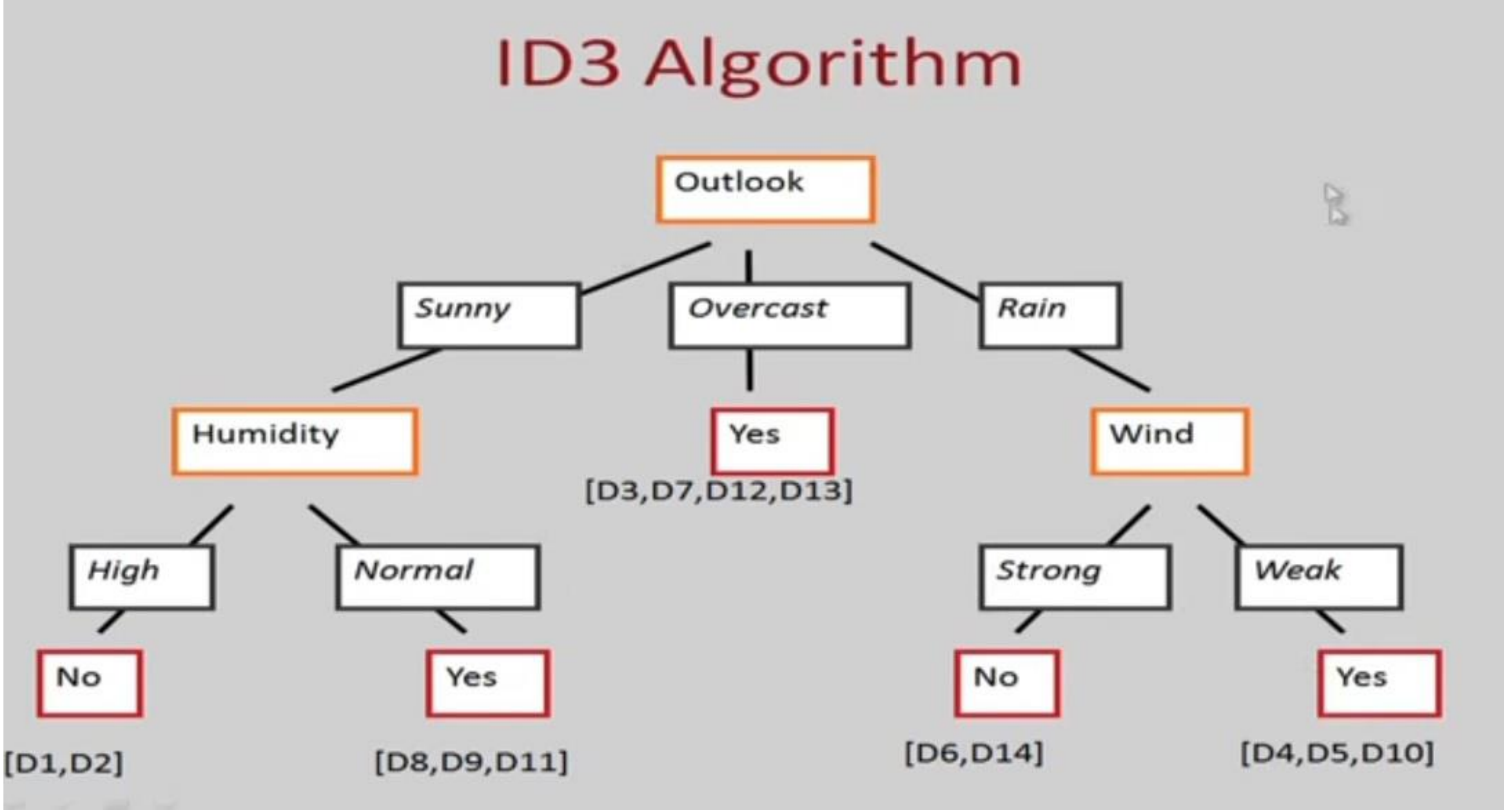
The information gain values for the 4 attributes are:

- Gain(S,Outlook) =0.247
- Gain(S,Humidity) =0.151
- Gain(S,Wind) =0.048
- Gain(S,Temperature) =0.029

where S denotes the collection of training examples

Note: $0 Log_2 0 = 0$


ID3 Algorithm
[D1,D2,...,D14]
[9+,5-]
Outlook
Sunny
Overcast
Rain
Ssunny=[D1,D2,D8,D9,D11]
[2+,3-]
[D3,D7,D12,D13]
[4+,0-]
[D4,D5,D6,D10,D14]
[3+,2-]
Test for this node
?
Yes
?
Gain(Ssunny, Humidity)=0.970-(3/5)0.0 – 2/5(0.0) = 0.970
Gain(Ssunny, Temp.)=0.970-(2/5)0.0 –2/5(1.0)-(1/5)0.0 = 0.570
Gain(Ssunny, Wind)=0.970= -(2/5)1.0 – 3/5(0.918) = 0.019


ID3 Algorithm
Outlook
Sunny
Overcast
Rain
Humidity
Yes
[D3,D7,D12,D13]
Wind
High
Normal
Strong
Weak
No
Yes
No
Yes
[D1,D2]
[D8,D9,D11]
[D6,D14]
[D4,D5,D10]

Splitting Rule: GINI Index

- GINI Index
  - Measure of node impurity

$$GINI_{node}(Node) = 1 - \sum_{c \in classes} [p(c)]^2$$

$$GINI_{split}(A) = \sum_{v \in Values(A)} \frac{|S_v|}{|S|} GINI(N_v)$$

Decision Tree using Gini Index – Solved Example

Weekend	Weather	Parents	Money	Decision
W1	Sunny	Yes	Rich	Cinema
W2	Sunny	No	Rich	Tennis
W3	Windy	Yes	Rich	Cinema
W4	Rainy	Yes	Poor	Cinema
W5	Rainy	No	Rich	Stay In
W6	Rainy	Yes	Poor	Cinema
W7	Windy	No	Poor	Cinema
W8	Windy	No	Rich	Shopping
W9	Windy	Yes	Rich	Cinema
W10	Sunny	No	Rich	Tennis

- Compute the Gini Index for the overall collection of training examples.
- There are four possible output variables Cinema, Tennis, Stay In and Shopping.
- The data has 6 instances of Cinema, 2 instances of Tennis, 1 instance of Stay In and 1 of shopping.
- $Gini(S) = 1 - [\left(\frac{6}{10}\right)^2 + \left(\frac{2}{10}\right)^2 + \left(\frac{1}{10}\right)^2 + \left(\frac{1}{10}\right)^2] = 0.58$

Computation of **Gini Index for Money** Attribute

It has **two possible values of Rich (7 examples)** and **Poor (3 examples).**

For **Money = Poor,** there are **3 examples with "Cinema".**

$$Gini(S) = 1 - [\left(\frac{3}{3}\right)^2] = 0$$

For **Money = Rich,** there are **2 examples with "Tennis", 3 examples with "Cinema"** and **1 example with "Stay in", "Shopping" each**

$$Gini(S) = 1 - [\left(\frac{2}{7}\right)^2 + \left(\frac{3}{7}\right)^2 + \left(\frac{1}{7}\right)^2 + \left(\frac{1}{7}\right)^2] = 0.694$$

$$\boldsymbol{Weighted\ Average(Money)}$$

$$= 0 * \left(\frac{3}{10}\right) + 0.694 * \left(\frac{7}{10}\right) = 0.486$$

Computation of **Gini Index for Parents** Attribute

It has two possible values of **Yes (5 examples)** and **No (5 examples).**

For **Parents = Yes,** there are **5 examples**, all with "Cinema".

$$Gini(S) = 1 - [\left(\frac{5}{5}\right)^2] = 0$$

For **Parents = No,** there are **2 examples with "Tennis", 1 example with "Stay in", "Shopping" and "Cinema" each**

$$Gini(S) = 1 - [\left(\frac{2}{5}\right)^2 + \left(\frac{1}{5}\right)^2 + \left(\frac{1}{5}\right)^2 + \left(\frac{1}{5}\right)^2] = 0.72$$

$$\boldsymbol{Weighted\ Average(Parents)}$$

$$= \mathbf{0} * \left(\frac{\mathbf{5}}{\mathbf{10}}\right) + [\mathbf{0.72} * \left(\frac{\mathbf{5}}{\mathbf{10}}\right) = \mathbf{0.36}$$

Computation of **Gini Index for Weather** Attribute

It has three possible values of **Sunny (3 examples), Rainy (3 examples)** and **Windy (4 examples).**

For **Weather = Sunny**, there are **2 examples with "Cinema"** and **1 with "Tennis".**

$$Gini(Sunny) = 1 - [\left(\frac{2}{3}\right)^2 + \left(\frac{1}{3}\right)^2] = 0.444$$

For **Weather = Rainy**, there are **2 examples with "Cinema"** and **1 example with "Stay in"**

$$Gini(Rainy) = 1 - [\left(\frac{2}{3}\right)^2 + \left(\frac{1}{3}\right)^2] = 0.444$$

For **Weather = Windy**, there are **3 examples with "Cinema"** and **1 example with "Shopping"**

$$Gini(Windy) = 1 - [\left(\frac{3}{4}\right)^2 + \left(\frac{1}{4}\right)^2] = 0.375$$

$$Weighted\ Average(Weather)$$

$$= 0.444 * \left(\frac{3}{10}\right) + 0.444 * \left(\frac{3}{10}\right) + 0.375 * \left(\frac{4}{10}\right)$$

$$= 0.416$$

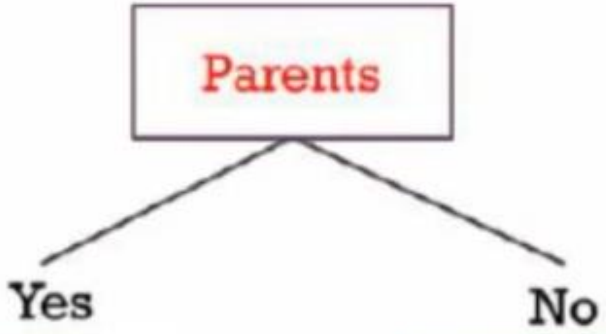
For Weather - Gini Index: 0.416

For Parents - Gini Index: 0.36

For Money - Gini Index: 0.486

Parents is selected as it has smallest Gini index.

Decision Tree using Gini Index – Solved Example



Weekend	Weather	Parents	Money	Decision
W1	Sunny	Yes	Rich	Cinema
W3	Windy	Yes	Rich	Cinema
W4	Rainy	Yes	Poor	Cinema
W6	Rainy	Yes	Poor	Cinema
W9	Windy	Yes	Rich	Cinema

Weekend	Weather	Parents	Money	Decision
W2	Sunny	No	Rich	Tennis
W5	Rainy	No	Rich	Stay In
W7	Windy	No	Poor	Cinema
W8	Windy	No	Rich	Shopping
W10	Sunny	No	Rich	Tennis

Decision Tree using Gini Index – Solved Example

Weekend	Weather	Parents	Money	Decision
W2	Sunny	No	Rich	Tennis
W5	Rainy	No	Rich	Stay In
W7	Windy	No	Poor	Cinema
W8	Windy	No	Rich	Shopping
W10	Sunny	No	Rich	Tennis

Computation of Gini Index for Parents = No |Weather Attribute

- **Sunny (2 examples)**
- For Parent= No |Weather = Sunny, there are 2 example with "Tennis.
- $Gini(S) = 1 - [(\frac{2}{2})^2] = 0$

Decision Tree using Gini Index – Solved Example

Weekend	Weather	Parents	Money	Decision
W2	Sunny	No	Rich	Tennis
W5	Rainy	No	Rich	Stay In
W7	Windy	No	Poor	Cinema
W8	Windy	No	Rich	Shopping
W10	Sunny	No	Rich	Tennis

Computation of Gini Index for Parents = No |Weather Attribute

- **Rainy (1 example).**
- For Parents = No | Weather = Rainy, there is 1 example with "Stay In".
- $Gini(S) = 1 - [\left(\frac{1}{1}\right)^2] = 0$

Decision Tree using Gini Index – Solved Example

Weekend	Weather	Parents	Money	Decision
W2	Sunny	No	Rich	Tennis
W5	Rainy	No	Rich	Stay In
W7	Windy	No	Poor	Cinema
W8	Windy	No	Rich	Shopping
W10	Sunny	No	Rich	Tennis

Computation of Gini Index for Parents = No |Weather Attribute

- **Windy (2 example)**
- For Parents = No | Weather = Windy, there is 1 example with "Cinema" and 1 example with "Shopping".
- $Gini(S) = 1 - \left[\left(\frac{1}{2}\right)^2 + \left(\frac{1}{2}\right)^2\right] = 0.5$

$$Weighted\ Average(Parents = No\,|\,Weather) = 0 * \left(\frac{2}{5}\right) + 0 * \left(\frac{1}{5}\right) + 0.5 * \left(\frac{2}{5}\right) = 0.2$$

Decision Tree using Gini Index – Solved Example

Weekend	Weather	Parents	Money	Decision
W2	Sunny	No	Rich	Tennis
W5	Rainy	No	Rich	Stay In
W7	Windy	No	Poor	Cinema
W8	Windy	No	Rich	Shopping
W10	Sunny	No	Rich	Tennis

Computation of Gini Index for Parents = No|Money Attribute

- Rich (4 examples)
- For Parents = No| Money = Rich, there is 1 example with "stay in" and "Shopping" each and 2 examples of "Tennis".
- $Gini(S) = 1 - [\left(\frac{1}{4}\right)^2 + \left(\frac{1}{4}\right)^2 + \left(\frac{2}{4}\right)^2] = 0.625$

Decision Tree using Gini Index – Solved Example

Weekend	Weather	Parents	Money	Decision
W2	Sunny	No	Rich	Tennis
W5	Rainy	No	Rich	Stay In
W7	Windy	No	Poor	Cinema
W8	Windy	No	Rich	Shopping
W10	Sunny	No	Rich	Tennis

Computation of Gini Index for Parents = No|Money Attribute

- Poor (1 example)
- For Parents = No| Money = Poor, there is 1 example with "Cinema".
- $Gini(S) = 1 - [\left(\frac{1}{1}\right)^2] = 0$
- $Weighted\ Average\ (Parents = No|\ Money) = 0.625 * (4/5) + 0 * (1/5) = 0.5$

Decision Tree using Gini Index – Solved Example

Weekend	Weather	Parents	Money	Decision
W2	Sunny	No	Rich	Tennis
W5	Rainy	No	Rich	Stay In
W7	Windy	No	Poor	Cinema
W8	Windy	No	Rich	Shopping
W10	Sunny	No	Rich	Tennis

For Parents = No | Weather - Gini Index: 0.2

For Parents = No | Money - Gini Index: 0.5

Decision Tree using Gini Index – Solved Example

Weekend	Weather	Parents	Money	Decision
W2	Sunny	No	Rich	Tennis
W5	Rainy	No	Rich	Stay In
W7	Windy	No	Poor	Cinema
W8	Windy	No	Rich	Shopping
W10	Sunny	No	Rich	Tennis

Now, for Parent=No & Weather=Sunny, we have all instances as Tennis.

Weekend	Weather	Parents	Money	Decision
W2	Sunny	No	Rich	Tennis
W10	Sunny	No	Rich	Tennis

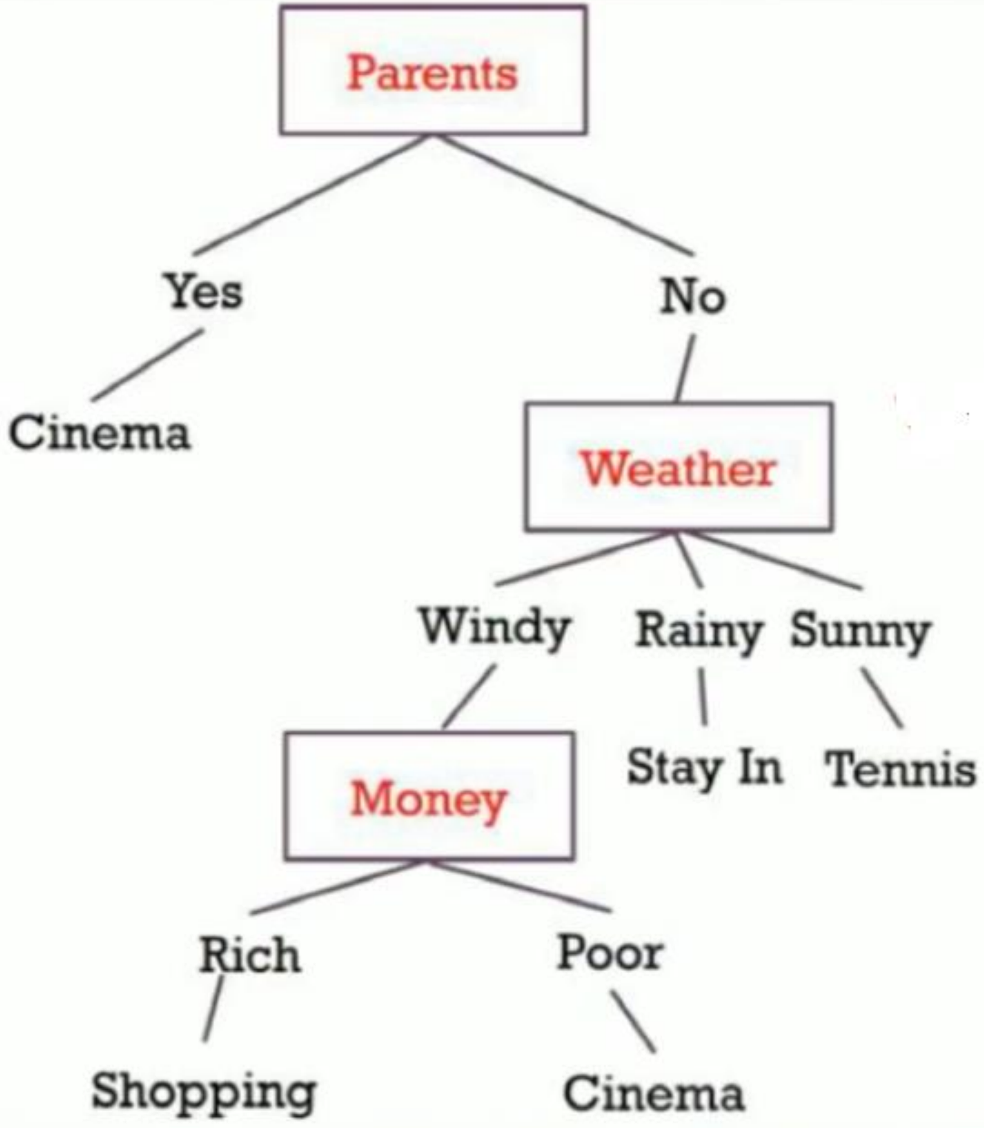
Now, for Parents=No & Weather=Rainy, we have all instances as Stay In.

Weekend	Weather	Parents	Money	Decision
W5	Rainy	No	Rich	Stay In

Now, for Parent=No & Weather=Windy, we need to split.

Weekend	Weather	Parents	Money	Decision
W7	Windy	No	Poor	Cinema
W8	Windy	No	Rich	Shopping

Decision Tree using Gini Index – Solved Example



Avoid overfitting in Decision Trees

1. Stop growing the tree earlier, before it reaches the point where it perfectly classifies the training data
2. Allow the tree to *overfit* the data, and then *post-prune* the tree
   - Split the training in two parts (training and validation) and use validation to assess the utility of *post-pruning*
     - Reduced error pruning
     - Rule Post pruning

Avoid overfitting in Decision Trees

Reduced-error pruning

1. Each node is a candidate for pruning
2. *Pruning* consists in removing a subtree rooted in a node: the node becomes a leaf and is assigned the most common classification
3. Nodes are removed only if the resulting tree performs better on the **validation set.**
4. Nodes are pruned iteratively: at each iteration the node whose removal most increases accuracy on the validation set is pruned.

Avoid overfitting in Decision Trees

Reduced-error pruning

Avoid overfitting in Decision Trees

Rule post-pruning

1. Create the decision tree from the training set
2. Convert the tree into an equivalent set of rules
   - Each path corresponds to a rule
   - Each node along a path corresponds to a pre-condition
   - Each leaf classification to the post-condition
3. Prune (generalize) each rule by removing those preconditions whose removal improves accuracy over validation set
4. Sort the rules in estimated order of accuracy, and consider them in sequence when classifying new instances

Avoid overfitting in Decision Trees

Rule post-pruning

1. Outlook=sunny ^ humidity=high -> No
2. Outlook=sunny ^ humidity=normal -> Yes
3. Outlook=overcast -> Yes
4. Outlook=rain ^ wind=strong -> No
5. Outlook=rain ^ wind=weak -> Yes

Compare first rule to:

Outlook=sunny->No

Humidity=high->No

Calculate accuracy of 3 rules based on validation set and pick best version.